SolidFire and Element Software 12.5 Documentation

Element Software

NetApp
November 19, 2025

Table of Contents

SolidFire and Element Software 12.5 Documentation

Current and previous release information

You can find links to the latest and earlier release notes for various components of the Element storage environment.

You will be prompted to log in using your NetApp Support credentials.

NetApp Element software

- NetApp Element Software 12.7 Release Notes
- NetApp Element Software 12.5 Release Notes
- NetApp Element Software 12.3.2 Release Notes
- NetApp Element Software 12.3.1 Release Notes
- NetApp Element Software 12.3 Release Notes
- NetApp Element Software 12.2.1 Release Notes
- NetApp Element Software 12.2 Release Notes
- NetApp Element Software 12.0.1 Release Notes
- NetApp Element Software 12.0 Release Notes
- NetApp Element Software 11.8.2 Release Notes
- NetApp Element Software 11.8.1 Release Notes
- NetApp Element Software 11.8 Release Notes
- NetApp Element Software 11.7 Release Notes
- NetApp Element Software 11.5.1 Release Notes
- NetApp Element Software 11.3P1 Release Notes

Management services

- Management Services Release Notes

NetApp Element Plug-in for vCenter Server

- vCenter Plug-in 5.3 Release Notes *NEW*
- vCenter Plug-in 5.2 Release Notes
- vCenter Plug-in 5.1 Release Notes
- vCenter Plug-in 5.0 Release Notes
- vCenter Plug-in 4.10 Release Notes
- vCenter Plug-in 4.9 Release Notes
- vCenter Plug-in 4.8 Release Notes
- vCenter Plug-in 4.7 Release Notes
- vCenter Plug-in 4.6 Release Notes

- vCenter Plug-in 4.5 Release Notes
- vCenter Plug-in 4.4 Release Notes
- vCenter Plug-in 4.3 Release Notes

Storage firmware

- Storage Firmware Bundle 2.175.0 Release Notes *NEW*
- Storage Firmware Bundle 2.164.0 Release Notes
- Storage Firmware Bundle 2.150 Release Notes
- Storage Firmware Bundle 2.146 Release Notes
- Storage Firmware Bundle 2.99.2 Release Notes
- Storage Firmware Bundle 2.76 Release Notes
- Storage Firmware Bundle 2.27 Release Notes
- H610S BMC 3.84.07 Release Notes
- Supported firmware and ESXi driver versions *NEW*

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server
- Documentation for earlier versions of NetApp SolidFire and Element products
- SolidFire all-flash storage overview

What's new in Element software 12.5 and later

NetApp periodically updates SolidFire and Element software to bring you new features, enhancements, and bug fixes. Element 12.7 is the latest release and includes security and system component updates, operational improvements, and resolved issues.

The cumulative software and firmware updates are installed as part of an Element 12.7 upgrade based on the current Element version running on a storage cluster. For example, if a cluster is currently running Element 12.3.x, you can upgrade directly to Element 12.7 to get the cumulative updates from both Element 12.5 and 12.7. For supported upgrade paths, see this KB article

Element 12.7

Learn more about what is new in Element 12.7.

Secure CHAP algorithms

Element 12.7 includes support for secure FIPS compliant Challenge-Handshake Authentication Protocol (CHAP) algorithms SHA1, SHA-256, and SHA3-256. Learn more.

Dynamic block (bin) sync-in rate

Cluster operations like additions, upgrades, or maintenance of nodes, or addition of drives, and so on, trigger block (bin) sync to distribute block data to the new or updated nodes in a cluster layout. Using a single slow speed as the default sync-in rate causes these operations to take a long time and does not take advantage of the higher processing power of larger nodes. Beginning with Element 12.7, the sync-in rate is dynamically tuned based on the number of cores on the storage node, enabling these operations to go significantly faster.

For example, when you add large 28-core storage nodes (H610S, SF19210, and SF38410) running Element 12.7 to an existing cluster, the sync-in rate for data automatically tunes to 110Mbps instead of 60Mbps. Additionally, when you bring these large storage nodes out of node maintenance mode, for example, during an upgrade from Element 12.3.x or later to Element 12.7 using NetApp Hybrid Cloud Control, the sync-in rate for changed block data rate automatically tunes to 110Mbps instead of 20Mbps.

When you add the medium 16-core storage nodes (H410S) and small 12-core storage nodes (SF4805) to an Element 12.7 cluster, the sync-in rate for data remains at 60Mbps; however, for syncing changed blocks when you bring them out of node maintenance mode during upgrades from Element 12.3.x to Element 12.7, the sync-in rate automatically tunes from 20Mbps to 60Mbps for medium storage nodes and 40Mbps for smaller storage nodes.

When you remove storage nodes, there is no impact on the block sync-out rate which avoids performance impacts on client I/O.

Garbage Collection improvement

For clusters with larger storage nodes, for example, an H610S-4, that have 1PB of used space, are running very high workloads with overwrites, and have high deduplication and compression, the Garbage Collection operation can now keep up as the default bloom filter size has been increased for the larger nodes from 700GB or greater of memory to 1048576 bits. This change automatically takes effect after you upgrade your storage nodes to Element 12.7 and has no impact on smaller nodes.

Scale improvement

With Element 12.7, you no longer need to follow specific sequencing when adding multiple storage nodes worth of block and metadata drives to an existing cluster. Using the Element UI or API, you can simply select all available drives and bulk add them simultaneously. Element 12.7 automatically manages the data synchronization such that all block services are synced in simultaneously. As the block services for each node complete syncing, the metadata drive on that node becomes assignable to host volumes. This scale improvement significantly reduces read response latency and prevents performance degradation while data is syncing across newly added storage nodes.

Storage node firmware updates

Element 12.7 includes storage firmware bundle version 2.164.0, which includes support for new system components. Learn more.

> There are no new firmware updates in the Element 12.7 release. However, based on the current firmware bundle running on the storage nodes, the cumulative updates are installed when you upgrade to Element 12.7.

SolidFire Active IQ documentation

In the SolidFire Active IQ UI, you can now navigate to the QoS Management page to view recommendations and node throttling information for your cluster. In addition, the cluster dashboard now displays the total

snapshot count. Other recent enhancements include the addition of primary and secondary node information for active volumes and average throughput, IOPS, and average latency for the last 30 minutes on the primary volumes on a node.

You can now access the SolidFire Active IQ documentation from the Element software documentation. Learn more.

NetApp Bugs Online contains resolved and known issues

Resolved and known issues are listed in the NetApp Bugs Online tool. You can browse these issues for Element software and other products at NetApp Bugs Online.

Element 12.5

Element 12.5 introduces improved storage node access, enhanced custom Protection Domains manageability, new and improved cluster faults and events, enhanced Create Cluster UI functionality, and enhanced security.

Improved storage node access

Element 12.5 brings improved remote access to individual nodes using signed SSH certificates. To provide secure remote access to storage nodes, a new, limited-privilege local user account called `sfreadonly` is now created during RTFI of a storage node. The `sfreadonly` account enables access to the storage node backend for basic maintenance or troubleshooting purposes. You can now configure the `supportAdmin` access type for a cluster administrator user to allow NetApp support access to the cluster on an as-needed basis.

Enhanced custom Protection Domains manageability

Element 12.5 features a new user interface that enables you to quickly and easily view existing custom Protection Domains and configure new custom Protection Domains.

New and improved cluster faults, events, and alerts

Element 12.5 enhances your system troubleshooting with the introduction of the new cluster fault codes `BmcSelfTestFailed` and `CpuThermalEventThreshold`. Element 12.5 also contains robustness improvements for existing cluster events and alerts, such as `nodeOffline`, `volumeOffline`, `driveHealthFault`, `networkEvent`, and `cSumEvent`.

Enable Software Encryption at Rest from the Create Cluster UI

With the addition of a new checkbox in the Create Cluster UI, Element 12.5 gives you the option to enable cluster-wide Software Encryption at Rest for SolidFire all-flash storage clusters during cluster creation.

Storage node firmware updates

Element 12.5 includes firmware updates for storage nodes. Learn more.

Enhanced security

Element 12.5 contains the mitigation that closes the Element software exposure to the Apache Log4j vulnerability. NetApp SolidFire storage clusters with the Virtual Volumes (VVols) feature enabled are exposed to the Apache Log4j vulnerability. For information on the workaround for the Apache Log4j vulnerability in NetApp Element software, see the KB article.

If you're running Element 11.x, 12.0, or 12.2 or your storage cluster is already at Element 12.3 or 12.3.1 with the VVols feature enabled, you should upgrade to 12.5.

Element 12.5 also includes more than 120 CVE security vulnerability remediations.

Find more information

- NetApp Hybrid Cloud Control and Management Services Release Notes
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation
- SolidFire and Element Software Documentation
- SolidFire and Element Software Documentation Center for previous versions
- NetApp HCI Resources page
- Supported storage firmware versions for SolidFire storage nodes

Concepts

Learn basic concepts related to Element software.

- Product overview
- SolidFire architecture overview
- Nodes
- Clusters
- Security
- Accounts and permissions
- Volumes
- Data protection
- Performance and quality of service

Find more information

- SolidFire all-flash storage overview
- SolidFire and Element Software Documentation

Product overview

A SolidFire all-flash storage system is comprised of discrete hardware components (drive and nodes) that are combined into a single pool of storage resources. This unified cluster presents as a single storage system for use by external clients and is managed with NetApp Element software.

Using the Element interface, API, or other management tools, you can monitor SolidFire cluster storage capacity and performance, and manage storage activity across a multi-tenant infrastructure.

SolidFire features

A Solidfire system provides the following features:

- Offers high performance storage for your large scale, private cloud infrastructure
- Provides a flexible scale that lets you meet changing storage needs
- Uses an API-driven storage management Element software interface
- Guarantees performance using Quality of Service policies
- Includes automatic load balancing across all nodes in the cluster
- Rebalances clusters automatically when nodes are added or subtracted

SolidFire deployment

Use storage nodes provided by NetApp and integrated with NetApp Element software.

SolidFire all-flash storage architecture overview

Find more information

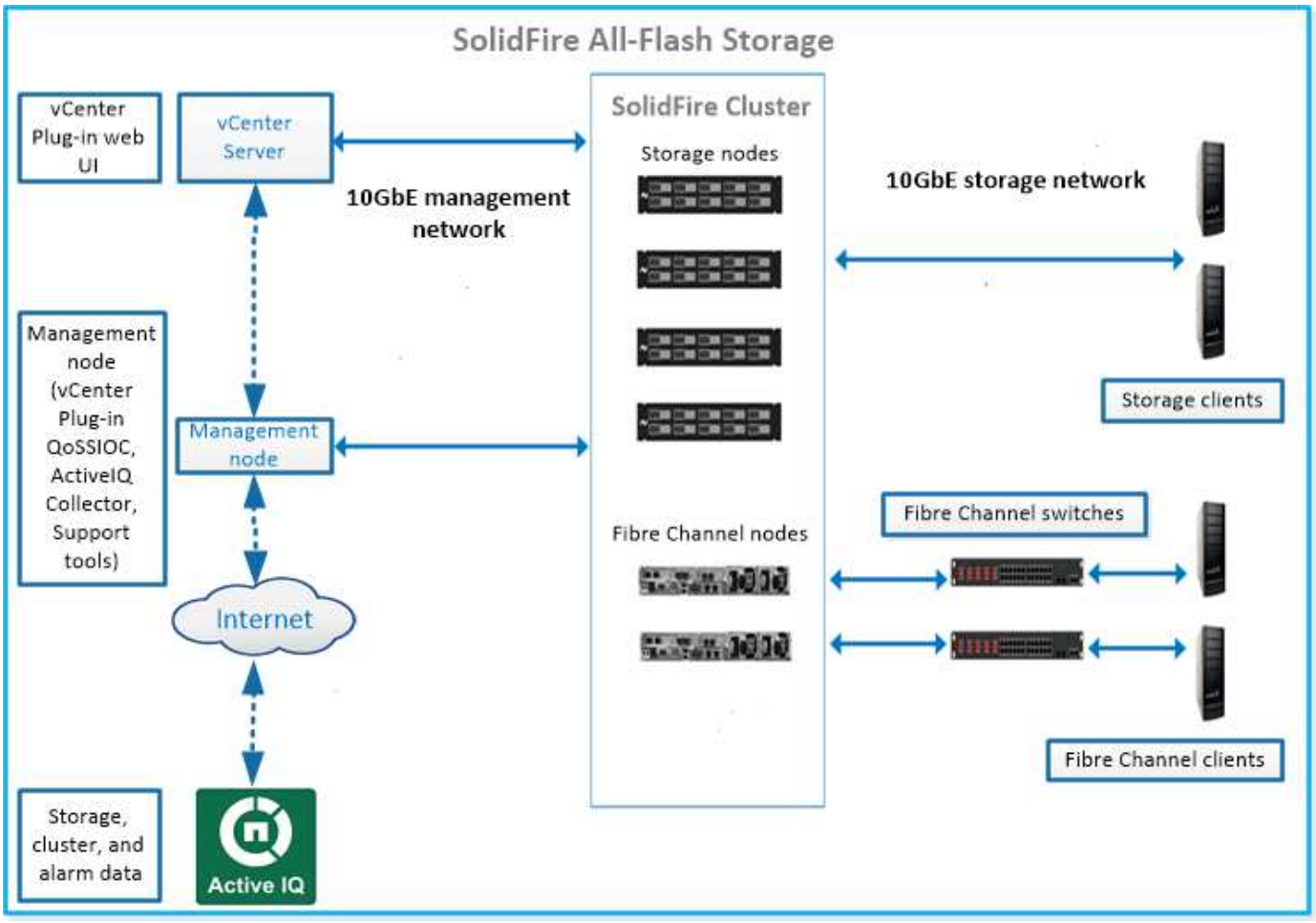
- SolidFire all-flash storage overview
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

SolidFire architecture overview

A SolidFire all-flash storage system is comprised of discrete hardware components (drive and nodes) that are combined into a pool of storage resources with NetApp Element software running independently on each node. This single storage system is managed as a single entity by using the Element software UI, API and other management tools.

A SolidFire storage system includes the following hardware components:

- **Cluster**: The hub of the SolidFire storage system that is a collection of nodes.
- **Nodes**: The hardware components grouped into a cluster. There are two types of nodes:
  - Storage nodes, which are servers containing a collection of drives
  - Fibre Channel (FC) nodes, which you use to connect to FC clients
- **Drives**: Used in storage nodes to store data for the cluster. A storage node contains two types of drives:
  - Volume metadata drives store information that defines the volumes and other objects within a cluster.
  - Block drives store data blocks for volumes.



You can manage, monitor, and update the system using the Element web UI and other compatible tools:

- SolidFire software interfaces
- SolidFire Active IQ
- Management node for Element software
- Management services

Common URLs

These are the common URLs you use with a SolidFire all-flash storage system:

URL	Description
`https://[storage cluster MVIP address]`	Access the NetApp Element software UI.
`https://activeiq.solidfire.com`	Monitor data and receive alerts to any performance bottlenecks or potential system issues.
`https://[management node IP address]`	Access NetApp Hybrid Cloud Control to upgrade your storage installation and update management services.
`https://[IP address]:442`	From the per-node UI, access network and cluster settings and utilize system tests and utilities. Learn more.

URL	Description
`https://[management node IP address]/mnode`	Use management services REST API and other functionality from the management node. Learn more.
`https://[management node IP address]:9443`	Register the vCenter Plug-in package in the vSphere Web Client. Learn more.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

SolidFire software interfaces

You can manage a SolidFire storage system using different NetApp Element software interfaces and integration utilities.

Options

- NetApp Element software user interface
- NetApp Element software API
- NetApp Element Plug-in for vCenter Server
- NetApp Hybrid Cloud Control
- Management node UIs
- Additional integration utilities and tools

NetApp Element software user interface

Enables you to set up Element storage, monitor cluster capacity and performance, and manage storage activity across a multi-tenant infrastructure. Element is the storage operating system at the heart of a SolidFire cluster. Element software runs independently on all nodes in the cluster and enables the nodes of the cluster to combine resources that are presented as a single storage system to external clients. Element software is responsible for all cluster coordination, scale and management of the system as a whole. The software interface is built upon the Element API.

Manage storage with Element software

NetApp Element software API

Enables you to use a set of objects, methods, and routines to manage Element storage. The Element API is based on the JSON-RPC protocol over HTTPS. You can monitor API operations in the Element UI by enabling the API Log; this enables you to see the methods that are being issued to the system. You can enable both requests and responses to see how the system replies to the methods that are issued.

Manage storage with the Element API

NetApp Element Plug-in for vCenter Server

Enables you to configure and manage storage clusters running Element software using an alternative interface for the Element UI within VMware vSphere.

NetApp Element Plug-in for vCenter Server

NetApp Hybrid Cloud Control

Enables you to upgrade Element storage and management services and manage storage assets using the NetApp Hybrid Cloud Control interface.

Manage and monitor storage with NetApp Hybrid Cloud Control overview

Management node UIs

The management node contains two UIs: a UI for managing REST-based services and a per-node UI for managing network and cluster settings and operating system tests and utilities. From the REST API UI, you can access a menu of service-related APIs that control service-based system functionality from the management node.

Additional integration utilities and tools

Although you typically manage your storage with NetApp Element, NetApp Element API, and NetApp Element Plug-in for vCenter Server, you can use additional integration utilities and tools to access storage.

Element CLI

Element CLI enables you to control a SolidFire storage system using a command-line interface without having to use the Element API.

Element PowerShell Tools

Element PowerShell Tools enable you to use a collection of Microsoft Windows PowerShell functions that use the Element API to manage a SolidFire storage system.

Element SDKs

Element SDKs enable you to manage your SolidFire cluster using these tools:

- Element Java SDK: Enables programmers to integrate the Element API with the Java programming language.
- Element .NET SDK: Enables programmers to integrate the Element API with the .NET programming platform.
- Element Python SDK: Enables programmers to integrate the Element API with the Python programming language.

SolidFire Postman API testing suite

Enables programmers to use a collection of Postman functions that test Element API calls.

SolidFire Storage Replication Adapter

SolidFire Storage Replication Adapter integrates with the VMware Site Recovery Manager (SRM) to enable

communication with replicated SolidFire storage clusters and execute supported workflows.

SolidFire vRO

SolidFire vRO provides a convenient way to use the Element API to administer your SolidFire storage system with VMware vRealize Orchestrator.

SolidFire VSS Provider

SolidFire VSS Provider integrates VSS shadow copies with Element snapshots and clones.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

SolidFire Active IQ

SolidFire Active IQ is a web-based tool that provides continually updated historical views of cluster-wide data. You can set up alerts for specific events, thresholds, or metrics. SolidFire Active IQ enables you to monitor system performance and capacity, as well as stay informed about cluster health.

You can find the following information about your system in SolidFire Active IQ:

- Number of nodes and status of the nodes: healthy, offline, or fault
- Graphical representation of CPU, memory usage, and node throttling
- Details about the node, such as serial number, slot location in the chassis, model, and version of NetApp Element software running on the storage node
- CPU and storage-related information about the virtual machines

To learn about SolidFire Active IQ, see the SolidFire Active IQ documentation.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server
- NetApp Support Site > Tools for Active IQ

Management node for Element software

The management node (mNode) is a virtual machine that runs in parallel with one or more Element software-based storage clusters. It is used to upgrade and provide system services including monitoring and telemetry, manage cluster assets and settings, run system tests and utilities, and enable NetApp Support access for troubleshooting.

The management node interacts with a storage cluster to perform management actions, but is not a member of the storage cluster. Management nodes periodically collect information about the cluster through API calls and report this information to Active IQ for remote monitoring (if enabled). Management nodes are also responsible for coordinating software upgrades of the cluster nodes.

As of the Element 11.3 release, the management node functions as a microservice host, allowing for quicker updates of select software services outside of major releases. These microservices or management services are updated frequently as service bundles.

Management services for SolidFire all-flash storage

As of the Element 11.3 release, **management services** are hosted on the management node, allowing for quicker updates of select software services outside of major releases.

Management services provide central and extended management functionality for SolidFire all-flash storage. These services include NetApp Hybrid Cloud Control, Active IQ system telemetry, logging, and service updates, as well as the QoSSIOC service for the Element Plug-in for vCenter.

Learn more about management services releases.

Nodes

Nodes are hardware or virtual resources that are grouped into a cluster to provide block storage and compute capabilities.

NetApp Element software defines various node roles for a cluster. The types of node roles are the following:

- Management node
- Storage node
- Fibre Channel node

Nodes states vary depending on cluster association.

Management node

A management node is a virtual machine used to upgrade and provide system services including monitoring and telemetry, manage cluster assets and settings, run system tests and utilities, and enable NetApp Support access for troubleshooting.
Learn more

Storage node

A SolidFire storage node is a server containing a collection of drives that communicate with each other through the Bond10G network interface. Drives in the node contain block and metadata space for data storage and data management. Each node contains a factory image of NetApp Element software.

Storage nodes have the following characteristics:

- Each node has a unique name. If a node name is not specified by an administrator, it defaults to SF-XXXX, where XXXX is four random characters generated by the system.
- Each node has its own high-performance non-volatile random access memory (NVRAM) write cache to improve overall system performance and reduce write latency.
- Each node is connected to two networks, storage and management, each with two independent links for redundancy and performance. Each node requires an IP address on each network.

- You can create a cluster with new storage nodes, or add storage nodes to an existing cluster to increase storage capacity and performance.
- You can add or remove nodes from the cluster at any time without interrupting service.

Fibre Channel node

SolidFire Fibre Channel nodes provide connectivity to a Fibre Channel switch, which you can connect to Fibre Channel clients. Fibre Channel nodes act as a protocol converter between the Fibre Channel and iSCSI protocols; this enables you to add Fibre Channel connectivity to any new or existing SolidFire cluster.

Fibre Channel nodes have the following characteristics:

- Fibre Channel switches manage the state of the fabric, providing optimized interconnections.
- The traffic between two ports flows through the switches only; it is not transmitted to any other port.
- Failure of a port is isolated and does not affect operation of other ports.
- Multiple pairs of ports can communicate simultaneously in a fabric.

Node states of operation

A node can be in one of several states depending on the level of configuration.

- **Available**

  The node has no associated cluster name and is not yet part of a cluster.

- **Pending**

  The node is configured and can be added to a designated cluster.

  Authentication is not required to access the node.

- **Pending Active**

  The system is in the process of installing compatible Element software on the node. When complete, the node will move to the Active state.

- **Active**

  The node is participating in a cluster.

  Authentication is required to modify the node.

In each of these states, some fields are read only.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Clusters

A cluster is the hub of a SolidFire storage system and is made up of a collection of nodes. You must have at least four nodes in a cluster for SolidFire storage efficiencies to be realized. A cluster appears on the network as a single logical group and can then be accessed as block storage.

Creating a new cluster initializes a node as communications owner for a cluster and establishes network communications for each node in the cluster. This process is performed only once for each new cluster. You can create a cluster using the Element UI or the API.

You can scale out a cluster by adding additional nodes. When you add a new node, there is no interruption of service and the cluster automatically uses the performance and capacity of the new node.

Administrators and hosts can access the cluster using virtual IP addresses. Any node in the cluster can host the virtual IP addresses. The management virtual IP (MVIP) enables cluster management through a 1GbE connection, while the storage virtual IP (SVIP) enables host access to storage through a 10GbE connection. These virtual IP addresses enable consistent connections regardless of the size or makeup of a SolidFire cluster. If a node hosting a virtual IP address fails, another node in the cluster begins hosting the virtual IP address.

Beginning in Element version 11.0, nodes can be configured with IPv4, IPv6, or both addresses for their management network. This applies to both storage nodes and management nodes, except for management node 11.3 and later which does not support IPv6. When creating a cluster, only a single IPv4 or IPv6 address can be used for the MVIP and the corresponding address type must be configured on all nodes.

More on clusters

- Authoritative storage clusters
- Rule of thirds
- Stranded capacity
- Storage efficiency
- Storage cluster quorum

Authoritative storage clusters

The authoritative storage cluster is the storage cluster that NetApp Hybrid Cloud Control uses to authenticate users.

If your management node only has one storage cluster, then it is the authoritative cluster. If your management node has two or more storage clusters, one of those clusters is assigned as the authoritative cluster and only users from that cluster can log into NetApp Hybrid Cloud Control. To find out which cluster is the authoritative cluster, you can use the `GET /mnode/about` API. In the response, the IP address in the `token_url` field is the management virtual IP address (MVIP) of the authoritative storage cluster. If you attempt to log into NetApp Hybrid Cloud Control as a user that is not on the authoritative cluster, the login attempt will fail.

Many NetApp Hybrid Cloud Control features are designed to work with multiple storage clusters, but authentication and authorization have limitations. The limitation around authentication and authorization is that the user from the authoritative cluster can execute actions on other clusters tied to NetApp Hybrid Cloud Control even if they are not a user on the other storage clusters.

Before proceeding with managing multiple storage clusters, you should ensure that users defined on the authoritative clusters are defined on all other storage clusters with the same permissions. You can manage users from the Element software user interface.

See create and manage storage cluster assets for more information on working with management node storage cluster assets.

Rule of thirds

When you mix storage node types in a NetApp SolidFire storage cluster, no single storage node can contain more than 33% of the total storage cluster capacity.

Stranded capacity

If a newly added node accounts for more than 50 percent of the total cluster capacity, some of the capacity of this node is made unusable ("stranded"), so that it complies with the capacity rule. This remains the case until more storage capacity is added. If a very large node is added that also disobeys the capacity rule, the previously stranded node will no longer be stranded, while the newly added node becomes stranded. Capacity should always be added in pairs to avoid this from happening. When a node becomes stranded, an appropriate cluster fault is thrown.

Storage efficiency

Netapp SolidFire storage clusters make use of deduplication, compression, and thin provisioning to reduce the amount of physical storage needed for storing a volume.

- **Compression**

  Compression reduces the amount of physical storage required for a volume by combining data blocks in compression groups, each of which is stored as a single block.

- **Deduplication**

  Deduplication reduces the amount of physical storage required for a volume by discarding duplicate data blocks.

- **Thin provisioning**

  A thin-provisioned volume or LUN is one for which storage is not reserved in advance. Instead, storage is allocated dynamically, as it is needed. Free space is released back to the storage system when data in the volume or LUN is deleted

Storage cluster quorum

Element software creates a storage cluster from selected nodes, which maintains a replicated database of the cluster configuration. A minimum of three nodes are required to participate in the cluster ensemble to maintain quorum for cluster resiliency.

Security

When you use your SolidFire all-flash storage system, your data is protected by industry-standard security protocols.

Encryption at Rest (hardware)

All drives in storage nodes are capable of encryption leverage AES 256-bit encryption at the drive level. Each drive has its own encryption key, which is created when the drive is first initialized. When you enable the encryption feature, a cluster-wide password is created, and chunks of the password are then distributed to all nodes in the cluster. No single node stores the entire password. The password is then used to password-protect all access to the drives. The password is needed to unlock the drive and then not needed unless power is removed from the drive or the drive is locked.

Enabling the hardware encryption at rest feature does not affect performance or efficiency on the cluster. If an encryption-enabled drive or node is removed from cluster configuration with the Element API or Element UI, encryption at rest will be disabled on the drives. After the drive is removed, the drive can be secure erased by using the `SecureEraseDrives` API method. If a physical drive or node is forcibly removed, the data remains protected by the cluster-wide password and the drive's individual encryption keys.

Encryption at Rest (software)

Another type of encryption-at-rest, software encryption-at-rest enables all data written to SSDs in a storage cluster to be encrypted. When enabled, it encrypts all data written and decrypts all data read automatically in software. Software encryption at rest mirrors the Self-Encrypting Drive (SED) implementation in hardware to provide data security in the absence of SED.

For SolidFire all-flash storage clusters, software encryption at rest must be enabled during cluster creation and cannot be disabled after the cluster has been created.

Both software and hardware-based encryption-at-rest can be used independently or in combination with one another.

External key management

You can configure Element software to use a third-party KMIP-compliant key management service (KMS) to manage storage cluster encryption keys. When you enable this feature, the storage cluster's cluster-wide drive access password encryption key is managed by a KMS that you specify.

Element can use the following key management services:

- Gemalto SafeNet KeySecure
- SafeNet AT KeySecure
- HyTrust KeyControl
- Vormetric Data Security Manager
- IBM Security Key Lifecycle Manager

For more information on configuring external key management, see the get started with external key management documentation.

Multi-factor authentication

Multi-factor authentication (MFA) enables you to require users to present multiple types of evidence to authenticate with the NetApp Element web UI or storage node UI upon login. You can configure Element to accept only multi-factor authentication for logins integrating with your existing user management system and identity provider.
You can configure Element to integrate with an existing SAML 2.0 identity provider which can enforce multiple

authentication schemes, such as password and text message, password and email message, or other methods.

You can pair multi-factor authentication with common SAML 2.0 compatible identity providers (IdPs), such as Microsoft Active Directory Federation Services (ADFS) and Shibboleth.

To configure MFA, see the enable multi-factor authentication documentation.

FIPS 140-2 for HTTPS and data at rest encryption

NetApp SolidFire storage clusters support encryption that complies with the Federal Information Processing Standard (FIPS) 140-2 requirements for cryptographic modules. You can enable FIPS 140-2 compliance on your SolidFire cluster for both HTTPS communications and drive encryption.

When you enable FIPS 140-2 operating mode on your cluster, the cluster activates the NetApp Cryptographic Security Module (NCSM) and leverages FIPS 140-2 Level 1 certified encryption for all communication via HTTPS to the NetApp Element UI and API. You use the `EnableFeature` Element API with the `fips` parameter to enable FIPS 140-2 HTTPS encryption. On storage clusters with FIPS-compatible hardware, you can also enable FIPS drive encryption for data at rest using the `EnableFeature` Element API with the `FipsDrives` parameter.

For more information about preparing a new storage cluster for FIPS 140-2 encryption, see Create a cluster supporting FIPS drives.

For more information about enabling FIPS 140-2 on an existing, prepared cluster, see the EnableFeature Element API.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Accounts and permissions

To administer and provide access to storage resources on your system, you'll need to set up accounts for system resources.

Using Element storage, you can create and manage the following types of accounts:

- Administrator user accounts for the storage cluster
- User accounts for storage volume access
- Authoritative cluster user accounts for NetApp Hybrid Cloud Control

Storage cluster administrator accounts

There are two types of administrator accounts that can exist in a storage cluster running NetApp Element software:

- **Primary cluster administrator account**: This administrator account is created when the cluster is created. This account is the primary administrative account with the highest level of access to the cluster. This account is analogous to a root user in a Linux system. You can change the password for this administrator account.

- **Cluster administrator account**: You can give a cluster administrator account a limited range of administrative access to perform specific tasks within a cluster. The credentials assigned to each cluster administrator account are used to authenticate API and Element UI requests within the storage system.

A local (non-LDAP) cluster administrator account is required to access active nodes in a cluster via the per-node UI. Account credentials are not required to access a node that is not yet part of a cluster.

You can manage cluster administrator accounts by creating, deleting, and editing cluster administrator accounts, changing the cluster administrator password, and configuring LDAP settings to manage system access for users.

User accounts

User accounts are used to control access to the storage resources on a NetApp Element software-based network. At least one user account is required before a volume can be created.

When you create a volume, it is assigned to an account. If you have created a virtual volume, the account is the storage container.

Here are some additional considerations:

- The account contains the CHAP authentication required to access the volumes assigned to it.
- An account can have up to 2000 volumes assigned to it, but a volume can belong to only one account.
- User accounts can be managed from the NetApp Element Management extension point.

Authoritative cluster user accounts

Authoritative cluster user accounts can authenticate against any storage asset associated with the NetApp Hybrid Cloud Control instance of nodes and clusters. With this account, you can manage volumes, accounts, access groups, and more across all clusters.

Authoritative user accounts are managed from the top right menu User Management option in NetApp Hybrid Cloud Control.

The authoritative storage cluster is the storage cluster that NetApp Hybrid Cloud Control uses to authenticate users.

All users created on the authoritative storage cluster can log into the NetApp Hybrid Cloud Control. Users created on other storage clusters *cannot* log into Hybrid Cloud Control.

- If your management node only has one storage cluster, then it is the authoritative cluster.
- If your management node has two or more storage clusters, one of those clusters is assigned as the authoritative cluster and only users from that cluster can log into NetApp Hybrid Cloud Control.

While many NetApp Hybrid Cloud Control features work with multiple storage clusters, authentication and authorization have necessary limitations. The limitation around authentication and authorization is that users from the authoritative cluster can execute actions on other clusters tied to NetApp Hybrid Cloud Control even if they are not a user on the other storage clusters. Before proceeding with managing multiple storage clusters, you should ensure that users defined on the authoritative clusters are defined on all other storage clusters with the same permissions. You can manage users from NetApp Hybrid Cloud Control.

Volume accounts

Volume-specific accounts are specific only to the storage cluster on which they were created. These accounts enable you to set permissions on specific volumes across the network, but have no effect outside of those volumes.

Volume accounts are managed within the NetApp Hybrid Cloud Control Volumes table.

Storage

Volumes

The NetApp Element storage system provisions storage using volumes. Volumes are block devices accessed over the network by iSCSI or Fibre Channel clients.

Element storage enables you to create, view, edit, delete, clone, backup or restore volumes for user accounts. You can also manage each volume on a cluster, and add or remove volumes in volume access groups.

Persistent volumes

Persistent volumes allow management node configuration data to be stored on a specified storage cluster, rather than locally with a VM, so that data can be preserved in the event of management node loss or removal. Persistent volumes are an optional yet recommended management node configuration.

An option to enable persistent volumes is included in the installation and upgrade scripts when deploying a new management node. Persistent volumes are volumes on an Element software-based storage cluster that contain management node configuration information for the host management node VM that persists beyond the life of the VM. If the management node is lost, a replacement management node VM can reconnect to and recover configuration data for the lost VM.

Persistent volumes functionality, if enabled during installation or upgrade, automatically creates multiple volumes. These volumes, like any Element software-based volume, can be viewed using the Element software web UI, NetApp Element Plug-in for vCenter Server, or API, depending on your preference and installation. Persistent volumes must be up and running with an iSCSI connection to the management node to maintain current configuration data that can be used for recovery.

> Persistent volumes that are associated with management services are created and assigned to a new account during installation or upgrade. If you are using persistent volumes, do not modify or delete the volumes or their associated account

Virtual volumes (vVols)

vSphere Virtual Volumes is a storage paradigm for VMware that moves much of the storage management for vSphere from the storage system to VMware vCenter. With Virtual Volumes (vVols), you can allocate storage according to the requirements of individual virtual machines.

Bindings

The NetApp Element cluster chooses an optimal protocol endpoint, creates a binding that associates the ESXi host and virtual volume with the protocol endpoint, and returns the binding to the ESXi host. After it is bound, the ESXi host can perform I/O operations with the bound virtual volume.

Protocol endpoints

VMware ESXi hosts use logical I/O proxies known as protocol endpoints to communicate with virtual volumes. ESXi hosts bind virtual volumes to protocol endpoints to perform I/O operations. When a virtual machine on the host performs an I/O operation, the associated protocol endpoint directs I/O to the virtual volume with which it is paired.

Protocol endpoints in a NetApp Element cluster function as SCSI administrative logical units. Each protocol endpoint is created automatically by the cluster. For every node in a cluster, a corresponding protocol endpoint is created. For example, a four-node cluster will have four protocol endpoints.

iSCSI is the only supported protocol for NetApp Element software. Fibre Channel protocol is not supported. Protocol endpoints cannot be deleted or modified by a user, are not associated with an account, and cannot be added to a volume access group.

Storage containers

Storage containers are logical constructs that map to NetApp Element accounts and are used for reporting and resource allocation. They pool raw storage capacity or aggregate storage capabilities that the storage system can provide to virtual volumes. A VVol datastore that is created in vSphere is mapped to an individual storage container. A single storage container has all available resources from the NetApp Element cluster by default. If more granular governance for multi-tenancy is required, multiple storage containers can be created.

Storage containers function like traditional accounts and can contain both virtual volumes and traditional volumes. A maximum of four storage containers per cluster is supported. A minimum of one storage container is required to use VVols functionality. You can discover storage containers in vCenter during VVols creation.

VASA provider

To make vSphere aware of the vVol feature on the NetApp Element cluster, the vSphere admin must register the NetApp Element VASA Provider with vCenter. The VASA provider is the out-of-band control path between vSphere and the Element cluster. It is responsible for executing requests on the Element cluster on behalf of vSphere, such as creating VMs, making VMs available to vSphere, and advertising storage capabilities to vSphere.

The VASA provider runs as part of the cluster master in Element software. The cluster master is a highly available service that fails over to any node in the cluster as needed. If the cluster master fails over, the VASA provider moves with it, ensuring high availability for the VASA provider. All provisioning and storage management tasks use the VASA provider, which handles any changes needed on the Element cluster.

For Element 12.5 and earlier, do not register more than one NetApp Element VASA provider to a single vCenter instance. Where a second NetApp Element VASA provider is added, this renders all VVOL datastores inaccessible.

VASA support for up to 10 vCenters is available as an upgrade patch if you have already registered a VASA provider with your vCenter. To install, follow the directions in the VASA39 manifest and download the .tar.gz file from the NetApp Software Downloads site. The NetApp Element VASA provider uses a NetApp certificate. With this patch, the certificate is used unmodified by vCenter to support multiple vCenters for VASA and VVols use. Do not modify the certificate. Custom SSL certificates are not supported by VASA.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Volume access groups

By creating and using volume access groups, you can control access to a set of volumes. When you associate a set of volumes and a set of initiators with a volume access group, the access group grants those initiators access to that set of volumes.

Volume access groups in NetApp SolidFire storage enable iSCSI initiator IQNs or Fibre Channel WWPNs to access a collection of volumes. Each IQN that you add to an access group can access each volume in the group without using CHAP authentication. Each WWPN that you add to an access group enables Fibre Channel network access to the volumes in the access group.

Volume access groups have the following limits:

- A maximum of 128 initiators per volume access group.
- A maximum of 64 access groups per volume.
- An access group can be made up of a maximum of 2000 volumes.
- An IQN or WWPN can belong to only one volume access group.
- For Fibre Channel clusters, a single volume can belong to a maximum of four access groups.

Initiators

Initiators enable external clients access to volumes in a cluster, serving as the entry point for communication between clients and volumes. You can use initiators for CHAP-based rather than account-based access to storage volumes. A single initiator, when added to a volume access group, allows volume access group members to access all storage volumes added to the group without requiring authentication. An initiator can belong to only one access group.

Data protection

Data protection features include remote replication, volume snapshots, volume cloning, Protection Domains, and high availability with double Helix technology.

Element storage data protection includes the following concepts:

- Remote replication types
- Volume snapshots for data protection
- Volume clones
- Backup and restore process overview for Element storage
- Protection Domains
- Custom Protection Domains
- Double Helix high availability

Remote replication types

Remote replication of data can take the following forms:

- Synchronous and asynchronous replication between clusters
- Snapshot-only replication
- Replication between Element and ONTAP clusters using SnapMirror

For more information, see TR-4741: NetApp Element Software Remote Replication.

Synchronous and asynchronous replication between clusters

For clusters running NetApp Element software, real-time replication enables the quick creation of remote copies of volume data.

You can pair a storage cluster with up to four other storage clusters. You can replicate volume data synchronously or asynchronously from either cluster in a cluster pair for failover and failback scenarios.

Synchronous replication

Synchronous replication continuously replicates data from the source cluster to the target cluster and is affected by latency, packet loss, jitter, and bandwidth.

Synchronous replication is appropriate for the following situations:

- Replication of several systems over a short distance
- A disaster recovery site that is geographically local to the source
- Time-sensitive applications and the protection of databases
- Business continuity applications that require the secondary site to act as the primary site when the primary site is down

Asynchronous replication

Asynchronous replication continuously replicates data from a source cluster to a target cluster without waiting for the acknowledgments from the target cluster. During asynchronous replication, writes are acknowledged to the client (application) after they are committed on the source cluster.

Asynchronous replication is appropriate for the following situations:

- The disaster recovery site is far from the source and the application does not tolerate latencies induced by the network.
- There are bandwidth limitations on the network connecting the source and target clusters.

Snapshot-only replication

Snapshot-only data protection replicates changed data at specific points of time to a remote cluster. Only those snapshots that are created on the source cluster are replicated. Active writes from the source volume are not.

You can set the frequency of the snapshot replications.

Snapshot replication does not affect asynchronous or synchronous replication.

Replication between Element and ONTAP clusters using SnapMirror

With NetApp SnapMirror technology, you can replicate snapshots that were taken using NetApp Element software to ONTAP for disaster recovery purposes. In a SnapMirror relationship, Element is one endpoint and ONTAP is the other.

SnapMirror is a NetApp Snapshot replication technology that facilitates disaster recovery, designed for failover from primary storage to secondary storage at a geographically remote site. SnapMirror technology creates a replica, or mirror, of the working data in secondary storage from which you can continue to serve data if an outage occurs at the primary site. Data is mirrored at the volume level.

The relationship between the source volume in primary storage and the destination volume in secondary storage is called a data protection relationship. The clusters are referred to as endpoints in which the volumes reside and the volumes that contain the replicated data must be peered. A peer relationship enables clusters and volumes to exchange data securely.

SnapMirror runs natively on the NetApp ONTAP controllers and is integrated into Element, which runs on NetApp HCI and SolidFire clusters. The logic to control SnapMirror resides in ONTAP software; therefore, all SnapMirror relationships must involve at least one ONTAP system to perform the coordination work. Users manage relationships between Element and ONTAP clusters primarily through the Element UI; however, some management tasks reside in NetApp ONTAP System Manager. Users can also manage SnapMirror through the CLI and API, which are both available in ONTAP and Element.

See TR-4651: NetApp SolidFire SnapMirror Architecture and Configuration (login required)

You must manually enable SnapMirror functionality at the cluster level by using Element software. SnapMirror functionality is disabled by default, and it is not automatically enabled as part of a new installation or upgrade.

After enabling SnapMirror, you can create SnapMirror relationships from the Data Protection tab in the Element software.

NetApp Element software 10.1 and above supports SnapMirror functionality to copy and restore snapshots with ONTAP systems.

Systems running Element 10.1 and above include code that can communicate directly with SnapMirror on ONTAP systems running 9.3 or higher. The Element API provides methods to enable SnapMirror functionality on clusters, volumes, and snapshots. Additionally, the Element UI includes functionality to manage SnapMirror relationships between Element software and ONTAP systems.

Starting with Element 10.3 and ONTAP 9.4 systems, you can replicate ONTAP originated volumes to Element volumes in specific use cases with limited functionality.

For more information, see Perform replication between NetApp Element software and ONTAP (ONTAP CLI).

Volume snapshots for data protection

A volume snapshot is a point-in-time copy of a volume that you could later use to restore a volume to that specific time.

While snapshots are similar to volume clones, snapshots are simply replicas of volume metadata, so you cannot mount or write to them. Creating a volume snapshot also takes only a small amount of system resources and space, which makes snapshot creation faster than cloning.

You can replicate snapshots to a remote cluster and use them as a backup copy of the volume. This enables you to roll back a volume to a specific point in time by using the replicated snapshot; you can also create a

clone of a volume from a replicated snapshot.

You can back up snapshots from a Element cluster to an external object store, or to another Element cluster. When you back up a snapshot to an external object store, you must have a connection to the object store that allows read/write operations.

You can take a snapshot of an individual volume or multiple for data protection.

Volume clones

A clone of a single volume or multiple volumes is point-in-time copy of the data. When you clone a volume, the system creates a snapshot of the volume and then creates a copy of the data referenced by the snapshot.

This is an asynchronous process, and the amount of time the process requires depends on the size of the volume you are cloning and the current cluster load.

The cluster supports up to two running clone requests per volume at a time and up to eight active volume clone operations at a time. Requests beyond these limits are queued for later processing.

Backup and restore process overview for Element storage

You can back up and restore volumes to other SolidFire storage, as well as to secondary object stores that are compatible with Amazon S3 or OpenStack Swift.

You can back up a volume to the following:

- A SolidFire storage cluster
- An Amazon S3 object store
- An OpenStack Swift object store

When you restore volumes from OpenStack Swift or Amazon S3, you need manifest information from the original backup process. If you are restoring a volume that was backed up on a SolidFire storage system, no manifest information is required.

Protection Domains

A Protection Domain is a node or a set of nodes grouped together such that any part or even all of it might fail, while maintaining data availability. Protection Domains enable a storage cluster to heal automatically from the loss of a chassis (chassis affinity) or an entire domain (group of chassis).

You can manually enable Protection Domain monitoring using the NetApp Element Configuration extension point in the NetApp Element Plug-in for vCenter Server. You can select a Protection Domain threshold based on node or chassis domains. You can also enable Protection Domain monitoring using the Element API or web UI.

A Protection Domain layout assigns each node to a specific Protection Domain.

Two different Protection Domain layouts, called Protection Domain levels, are supported.

- At the node level, each node is in its own Protection Domain.
- At the chassis level, only nodes that share a chassis are in the same Protection Domain.
    - The chassis level layout is automatically determined from the hardware when the node is added to the cluster.

- In a cluster where each node is in a separate chassis, these two levels are functionally identical.

When creating a new cluster, if you are using storage nodes that reside in a shared chassis, you might want to consider designing for chassis-level failure protection using the Protection Domains feature.

Custom Protection Domains

You can define a custom Protection Domain layout that matches your specific chassis and node layout, and where each node is associated with one and only one custom Protection Domain. By default, each node is assigned to the same default custom Protection Domain.

If no custom Protection Domains are assigned:

- Cluster operation is unaffected.
- Custom level is neither tolerant nor resilient.

When you configure custom Protection Domains for a cluster, there are three possible levels of protection, which you can see from the Element web UI dashboard:

- Not protected: The storage cluster is not protected from the failure of one of its custom Protection Domains. To fix this, add additional storage capacity to the cluster or reconfigure the cluster's custom Protection Domains to protect the cluster from possible data loss.
- Fault tolerant: The storage cluster has enough free capacity to prevent data loss after the failure of one of its custom Protection Domains.
- Fault resilient: The storage cluster has enough free capacity to self-heal after the failure of one of its custom Protection Domains. After the healing process has completed, the cluster will be protected from data loss if additional domains were to fail.

If more than one custom Protection Domain is assigned, each subsystem will assign duplicates to separate custom Protection Domains. If this is not possible, it reverts to assigning duplicates to separate nodes. Each subsystem (for example, bins, slices, protocol endpoint providers, and ensemble) does this independently.

You can use the Element UI to configure custom Protection Domains, or you can use the following API methods:

- GetProtectionDomainLayout - shows which chassis and which custom Protection Domain each node is in.
- SetProtectionDomainLayout - enables a custom Protection Domain to be assigned to each node.

Double Helix high availability

Double Helix data protection is a replication method that spreads at least two redundant copies of data across all drives within a system. The "RAID-less" approach enables a system to absorb multiple, concurrent failures across all levels of the storage system and repair quickly.

Performance and quality of service

A SolidFire storage cluster has the ability to provide Quality of Service (QoS) parameters on a per-volume basis. You can guarantee cluster performance measured in inputs and outputs per second (IOPS) using three configurable parameters that define QoS: Min IOPS, Max IOPS, and Burst IOPS.

SolidFire Active IQ has a QoS recommendations page that provides advice on optimal configuration and set up of QoS settings.

Quality of Service parameters

IOPS parameters are defined in the following ways:

- **Minimum IOPS** - The minimum number of sustained inputs and outputs per second (IOPS) that the storage cluster provides to a volume. The Min IOPS configured for a volume is the guaranteed level of performance for a volume. Performance does not drop below this level.
- **Maximum IOPS** - The maximum number of sustained IOPS that the storage cluster provides to a volume. When cluster IOPS levels are critically high, this level of IOPS performance is not exceeded.
- **Burst IOPS** - The maximum number of IOPS allowed in a short burst scenario. If a volume has been running below the Max IOPS, burst credits are accumulated. When performance levels become very high and are pushed to maximum levels, short bursts of IOPS are allowed on the volume.

  Element software uses Burst IOPS when a cluster is running in a state of low cluster IOPS utilization.

  A single volume can accrue Burst IOPS and use the credits to burst above their Max IOPS up to their Burst IOPS level for a set "burst period." A volume can burst for up to 60 seconds if the cluster has the capacity to accommodate the burst. A volume accrues one second of burst credit (up to a maximum of 60 seconds) for every second that the volume runs below its Max IOPS limit.

  Burst IOPS are limited in two ways:

  - A volume can burst above its Max IOPS for a number of seconds equal to the number of burst credits that the volume has accrued.
  - When a volume bursts above its Max IOPS setting, it is limited by its Burst IOPS setting. Therefore, the burst IOPS never exceeds the burst IOPS setting for the volume.
- **Effective Max Bandwidth** - The maximum bandwidth is calculated by multiplying the number of IOPS (based on the QoS curve) by the IO size.

  Example: QoS parameter settings of 100 Min IOPS, 1000 Max IOPS, and 1500 Burst IOPs have the following effects on quality of performance:

  - Workloads are able to reach and sustain a maximum of 1000 IOPS until the condition of workload contention for IOPS becomes apparent on the cluster. IOPS are then reduced incrementally until IOPS on all volumes are within the designated QoS ranges and contention for performance is relieved.
  - Performance on all volumes is pushed toward the Min IOPS of 100. Levels do not drop below the Min IOPS setting but could remain higher than 100 IOPS when workload contention is relieved.
  - Performance is never greater than 1000 IOPS, or less than 100 IOPS for a sustained period. Performance of 1500 IOPS (Burst IOPS) is allowed, but only for those volumes that have accrued burst credits by running below Max IOPS and only allowed for a short periods of time. Burst levels are never sustained.

QoS value limits

Here are the possible minimum and maximum values for QoS.

Parameters	Min value	Default	4 4KB	5 8KB	6 16KB	262KB
Min IOPS	50	50	15,000	9,375*	5556*	385*
Max IOPS	100	15,000	200,000**	125,000	74,074	5128
Burst IOPS	100	15,000	200,000**	125,000	74.074	5128

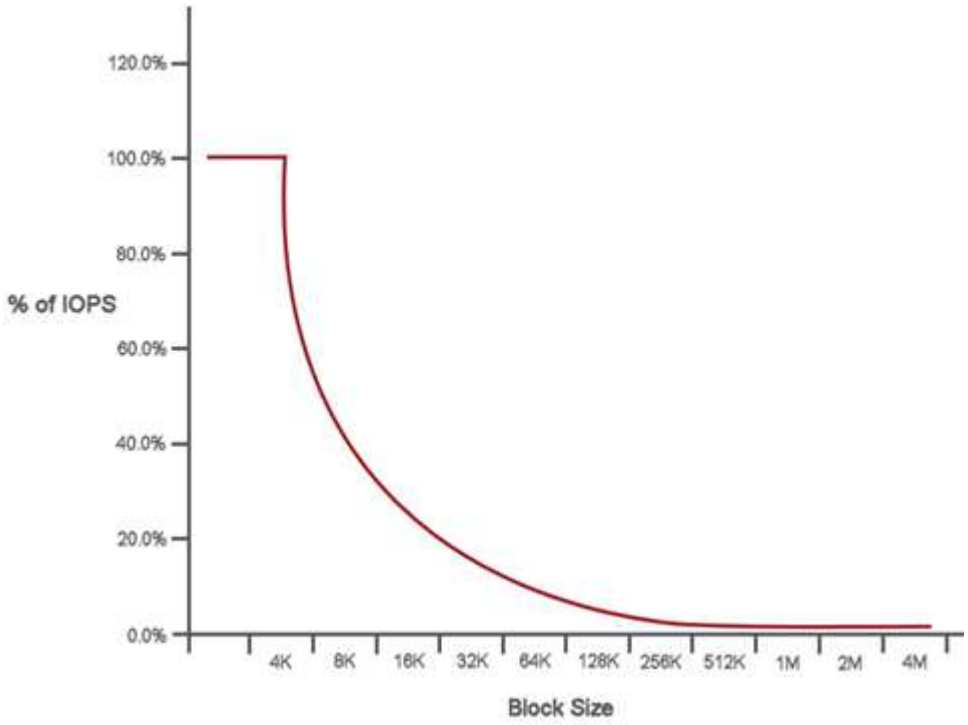
*These estimations are approximate.
**Max IOPS and Burst IOPS can be set as high as 200,000; however, this setting is allowed only to effectively uncap the performance of a volume. Real-world maximum performance of a volume is limited by cluster usage and per-node performance.

QoS performance

The QoS performance curve shows the relationship between block size and the percentage of IOPS.

Block size and bandwidth have a direct impact on the number of IOPS that an application can obtain. Element software takes into account the block sizes it receives by normalizing block sizes to 4k. Based on workload, the system might increase block sizes. As block sizes increase, the system increases bandwidth to a level necessary to process the larger block sizes. As bandwidth increases the number of IOPS the system is able to attain decreases.

The QoS performance curve shows the relationship between increasing block sizes and the decreasing percentage of IOPS:



As an example, if block sizes are 4k, and bandwidth is 4000 KBps, the IOPS are 1000. If block sizes increase to 8k, bandwidth increases to 5000 KBps, and IOPS decrease to 625. By taking block size into account, the system ensures that lower priority workloads that use higher block sizes, such as backups and hypervisor activities, do not take too much of the performance needed by higher priority traffic using smaller block sizes.

QoS policies

A QoS policy enables you to create and save a standardized quality of service setting that can be applied to many volumes.

QoS policies are best for service environments, for example, with database, application, or infrastructure servers that rarely reboot and need constant equal access to storage. Individual volume QoS is best for light use VMs, such as virtual desktops or specialized kiosk-type VMs, that may be rebooted, powered on, or powered off daily or several times a day.

QoS and QoS policies should not be used together. If you are using QoS policies, do not use custom QoS on a volume. Custom QoS will override and adjust QoS policy values for volume QoS settings.

The selected cluster must be Element 10.0 or later to use QoS policies; otherwise, QoS policy functions are not available.

Find more information

- SolidFire and Element Software Documentation

Requirements

Before you get started, you should review the prerequisites to deploy NetApp Element software, including networking and port requirements.

- Networking requirements
- Switch configuration
- Network port requirements

Find more information

- SolidFire and Element Software Documentation

Networking

The network setup for a SolidFire system consists of switch and port requirements. The implementation of these depends on your system.

For more information

- Switch configuration for clusters running Element software
- Network port requirements
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Switch configuration for clusters running Element software

The NetApp Element software system has certain switch requirements and best practices for optimal storage performance.

Storage nodes require 10 or 25GbE Ethernet switches, depending on specific node hardware, for iSCSI storage services and node intra-cluster services communication. 1GbE switches can be used for these types of traffic:

- Management of the cluster and the nodes
- Intra-cluster management traffic between the nodes
- Traffic between the cluster nodes and the management node virtual machine

Best Practice: You should implement the following best practices when configuring Ethernet switches for cluster traffic:

- For non-storage traffic in the cluster, deploy a pair of 1GbE switches to provide high availability and load sharing.
- On the storage network switches, deploy switches in pairs and configure and utilize jumbo frames (an MTU size of 9216 bytes). This ensures a successful installation and eliminates storage network errors due to fragmented packets.

Element deployment requires at least two network segments, one for each of the following types of traffic:

- Management
- Storage/Data

Depending on the NetApp H-Series storage node models and the planned cabling configuration, you can physically separate these networks using separate switches or logically separate them using VLANs. For most deployments, however, you need to logically separate these networks using VLANs.

Storage nodes need to be able to communicate before, during, and after deployment.

If you are implementing separate management networks for storage nodes, ensure that these management networks have network routes between them. These networks must have gateways assigned, and there must be a route between the gateways. Ensure that each new node has a gateway assigned to facilitate communication between nodes and management networks.

NetApp Element requires the following:

- All switch ports connected to NetApp H-Series storage nodes must be configured as spanning tree edge ports.
  - On Cisco switches, depending on the switch model, software version and port type, you can do this with one of the following commands:
    - `spanning-tree port type edge`
    - `spanning-tree port type edge trunk`
    - `spanning-tree portfast`
    - `spanning-tree portfast trunk`
  - On Mellanox switches, you can do this with the `spanning-tree port type edge` command.
- The switches handling storage traffic must support speeds of at least 10GbE per port (up to 25GbE per port is supported).
- The switches handling management traffic must support speeds of at least 1GbE per port.
- You must configure jumbo frames on the switch ports handling storage traffic. Hosts must be able to send 9000 byte packets end-to-end for a successful installation.
- Round-trip network latency between all storage nodes should not exceed 2ms.

Some nodes provide additional out-of-band management capabilities via a dedicated management port. NetApp H300S, H500S, and H700S nodes also allow for IPMI access via Port A. As a best practice, you should ease remote management by configuring out-of-band management for all nodes in your environment.

For more information

- NetApp HCI network and switch requirements
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Network port requirements

You might need to allow the following TCP and UDP ports through your data center's edge firewall so that you can manage the system remotely and allow clients outside of your data center to connect to resources. Some of these ports might not be required, depending on how you use the system.

All ports are TCP unless stated otherwise, and all TCP ports must support three-way handshake communication between the NetApp Support Server, management node, and nodes running Element software. For example, the host on a management node source communicates with the host on a storage cluster MVIP destination through TCP port 443, and the destination host communicates back to the source host through any port.

Enable ICMP between the management node, nodes running Element software, and cluster MVIP.

The following abbreviations are used in the table:

- MIP: Management IP address, a per-node address
- SIP: Storage IP address, a per-node address
- MVIP: Management virtual IP address
- SVIP: Storage virtual IP address

Source	Destination	Port	Description
iSCSI clients	Storage cluster MVIP	443	(Optional) UI and API access
iSCSI clients	Storage cluster SVIP	3260	Client iSCSI communications
iSCSI clients	Storage node SIP	3260	Client iSCSI communications
Management node	`sfsupport.solidfire.com`	22	Reverse SSH tunnel for support access
Management node	Storage node MIP	22	SSH access for support
Management node	DNS servers	53 TCP/UDP	DNS lookup
Management node	Storage node MIP	442	UI and API access to storage node and Element software upgrades

Source	Destination	Port	Description
Management node	Storage cluster MVIP	442	UI and API access to storage node and Element software upgrades
Management node	`monitoring.solidfire.com`	443	Storage cluster reporting to Active IQ
Management node	Storage cluster MVIP	443	UI and API access to storage node and Element software upgrades
Management node	repo.netapp.com	443	Provides access to components necessary to install/update on-premises deployment.
Management node	Storage node BMC/IPMI	623 UDP	RMCP port. This is required to manage IPMI-enabled systems.
Management node	Witness Node	9442	Per-node configuration API service
Management node	vCenter Server	9443	vCenter Plug-in registration. The port can be closed after registration is complete.
SNMP server	Storage cluster MVIP	161 UDP	SNMP polling
SNMP server	Storage node MIP	161 UDP	SNMP polling
Storage node BMC/IPMI	Management node	623 UDP	RMCP port. This is required to manage IPMI-enabled systems.
Storage node MIP	DNS servers	53 TCP/UDP	DNS lookup
Storage node MIP	Management node	80	Element software upgrades
Storage node MIP	S3/Swift endpoint	80	(Optional) HTTP communication to S3/Swift endpoint for backup and recovery
Storage node MIP	NTP server	123 UDP	NTP
Storage node MIP	Management node	162 UDP	(Optional) SNMP traps
Storage node MIP	SNMP server	162 UDP	(Optional) SNMP traps
Storage node MIP	LDAP server	389 TCP/UDP	(Optional) LDAP lookup
Storage node MIP	Management node	443	Element storage firmware upgrades

Source	Destination	Port	Description
Storage node MIP	Remote storage cluster MVIP	443	Remote replication cluster pairing communication
Storage node MIP	Remote storage node MIP	443	Remote replication cluster pairing communication
Storage node MIP	S3/Swift endpoint	443	(Optional) HTTPS communication to S3/Swift endpoint for backup and recovery
Storage node MIP	Management node	514 TCP/UDP 10514 TCP/UDP	Syslog forwarding
Storage node MIP	Syslog server	514 TCP/UDP 10514 TCP/UDP	Syslog forwarding
Storage node MIP	LDAPS server	636 TCP/UDP	LDAPS lookup
Storage node MIP	Remote storage node MIP	2181	Intercluster communication for remote replication
Storage node SIP	Remote storage node SIP	2181	Intercluster communication for remote replication
Storage node SIP	Storage node SIP	3260	Internode iSCSI
Storage node SIP	Remote storage node SIP	4000 through 4020	Remote replication node-to-node data transfer
System administrator PC	Management node	442	HTTPS UI access to management node
System administrator PC	Storage node MIP	442	HTTPS UI and API access to storage node
System administrator PC	Management node	443	HTTPS UI and API access to management node
System administrator PC	Storage cluster MVIP	443	HTTPS UI and API access to storage cluster

Source	Destination	Port	Description
System administrator PC	Storage node baseboard management controller (BMC)/Intelligent Platform Management Interface (IPMI) H410 and H600 series	443	HTTPS UI and API access to node remote control
System administrator PC	Storage node MIP	443	HTTPS storage cluster creation, post-deployment UI access to storage cluster
System administrator PC	Storage node BMC/IPMI H410 and H600 series	623 UDP	Remote Management Control Protocol port. This is required to manage IPMI-enabled systems.
System administrator PC	Witness Node	8080	Witness Node per-node web UI
vCenter Server	Storage cluster MVIP	443	vCenter Plug-in API access
vCenter Server	Remote plug-in	8333	Remote vCenter Plug-in service
vCenter Server	Management node	8443	(Optional) vCenter Plug-in QoSSIOC service.
vCenter Server	Storage cluster MVIP	8444	vCenter VASA provider access (VVols only)
vCenter Server	Management node	9443	vCenter Plug-in registration. The port can be closed after registration is complete.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Try it out

Learn about resources and tools for helping you get started with Element software.

- Lab on Demand for Private Cloud Storage Flexibility with Element (login required): This lab presents concepts of scale-out without limitations, guaranteed workload performance, and automation of storage infrastructure that apply to storage systems that run Element software.
- Try storage features using Element Demo Node: Element Demo Node is a VMware virtual machine version of Element software, which provides an easy way to demonstrate many of the key storage features of NetApp HCI and SolidFire products.

Find more information

- SolidFire All-Flash Storage Resources page

Try storage features using Element Demo Node

Element Demo Node is a VMware virtual machine (VM) version of Element software, which provides an easy way to demonstrate many of the key storage features of NetApp HCI and SolidFire products. The demo node enables developers to code against the Element API without the need for physical hardware. It is packaged as an OVA file for easy VMware deployment.

Supported functionality:

Element Demo Node is only intended for use as a demonstration and development tool. Be aware of the following functional limitations before using the demo node:

- Element Demo Node does not support clustering. It functions as a single-node cluster only.
- It does not support Element upgrades. To demo a newer version of Element, you should install a new demo node VM.
- It is not intended to demonstrate storage performance. Performance observed on the demo node is in no way indicative of the performance on the physical clusters.
- You cannot add demo nodes to NetApp HCI or SolidFire clusters.
- VRF VLANs are not supported (standard tagged VLANs are supported).
- Multi-drive slice service (MDSS) is not supported.
- Element Demo Node is only supported with VMFS datastores. VVols are not supported.
- Hardware-based configuration and monitoring functionality does not work with the demo node.
- It supports a maximum of 10 snapshots per volume.
- It supports a maximum 20 accounts per node/cluster.
- It supports a maximum of 100 volumes per account.
- It supports a maximum of 200 vVols per account.
- It supports a maximum volume size of 100 GiB.
- It supports a sustained cluster limit of 3000 IOPS.

All other Element software limitations apply. See the latest Element software Release Notes for details.

VM requirements

- 240-GB total capacity (The size and number of virtual disks for the VM cannot be changed. Any additional storage presented via the hypervisor is ignored by the guest OS.)
- 60 GB root disk
- Thick provisioned/eager zeroed (one 30-GB metadata drive or three 50-GB block drives) or thin provisioned/eager zeroed (**recommended**) (one 30-GB metadata drive or three 50-GB block drives)
- Two vCPU (fully reserved)
- 16-GB RAM (fully reserved)
- Single HBA for all disks, LSI Logic parallel
- Two vNICs, both vmxnet3 (one management, one storage)

Host requirements

- ESXI 6.0 or 6.5 for Element Demo Node 11.7 VM
- ESXi 6.5 for Element Demo Node 12.0 and 12.2 VMs
- ESXi 6.7 and 7.0 for Element Demo Node 12.3 and 12.5 VMs
- Multi core 64-bit Intel® architecture

Download Element Demo Node

The Element Demo Node software is a set of VMware files that have been packaged in an .ova file.

Install Element Demo Node on VMware ESXi

Installing Element Demo Node on VMware ESXi involves the following tasks:

- Configure network interfaces
- Register the demo node on an ESXi server
- Start the demo node on an ESXi server

Configure network interfaces

The Element Demo Node requires two separate virtual machine networks. One is for storage traffic and the other is for management traffic.
You should configure the storage network to support jumbo frames.

Register the demo node on an ESXi server

To register Element Demo Node on an ESXi server, you should deploy the demo node .ova file using the vSphere Client.

Steps

1. Log in to the vSphere Client, and select the ESXi host from the inventory panel.

2. Select **File** > **Deploy OVF Template**.

   The Deploy OVF Template Wizard is launched.

3. On the **Select template** page, browse to the OVA file you downloaded, and select **Open**.
4. Select **Next**.
5. On the **Name and Location** page, specify a name and location for the deployed template, and then select **Next**.
6. On the **Select a resource** page, browse to the location where you want to run the template, and select **Next**.
7. Verify the details, and select **Next**.
8. On the Select storage page, select where you want to store the virtual machine files and then select **Next**.
9. On the **Select networks** page, map the network used in the OVA file to the two separate virtual machine networks in your inventory, and select **Next**.
10. On the **Ready to complete** page, verify the details about the virtual machine you are creating, and then select **Finish**.

The demo node deployment might take a few minutes to complete.

Start the demo node on an ESXi server

You should start the demo node VM to access Element through the VMware ESXi console.

Steps

1. In the vSphere Client, select the demo node VM that you created.
2. Select the **Summary** tab to view the details about this VM.
3. Select **Power On** to start the VM.
4. Select **Launch Web Console**.
5. Use the TUI to configure the demo node. For more information, see Configure a storage node.

How to get support

Element Demo Node is available on a best-effort volunteer basis. For support, post your questions to the Element Demo Node Forum.

Find more information

- SolidFire All-Flash Storage Resources page
- Element Demo Node download page (login required)

Install and maintain hardware

Learn about installing and maintaining H-series and SF-series hardware.

- H410S and H610S hardware information
- SF-series hardware information
- Return to Factory Image information

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

H410S and H610S hardware information

You can find information about installing and maintaining H-series storage nodes.

Here are the links to the installation and maintenance content:

- Install H-series storage nodes
- Replace a H410S node
- Replace a H610S node
- Replace drives
- Replace a power supply unit

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Install H-series storage nodes

Before you get started with your all-flash storage system, you should install and set up the storage nodes correctly.

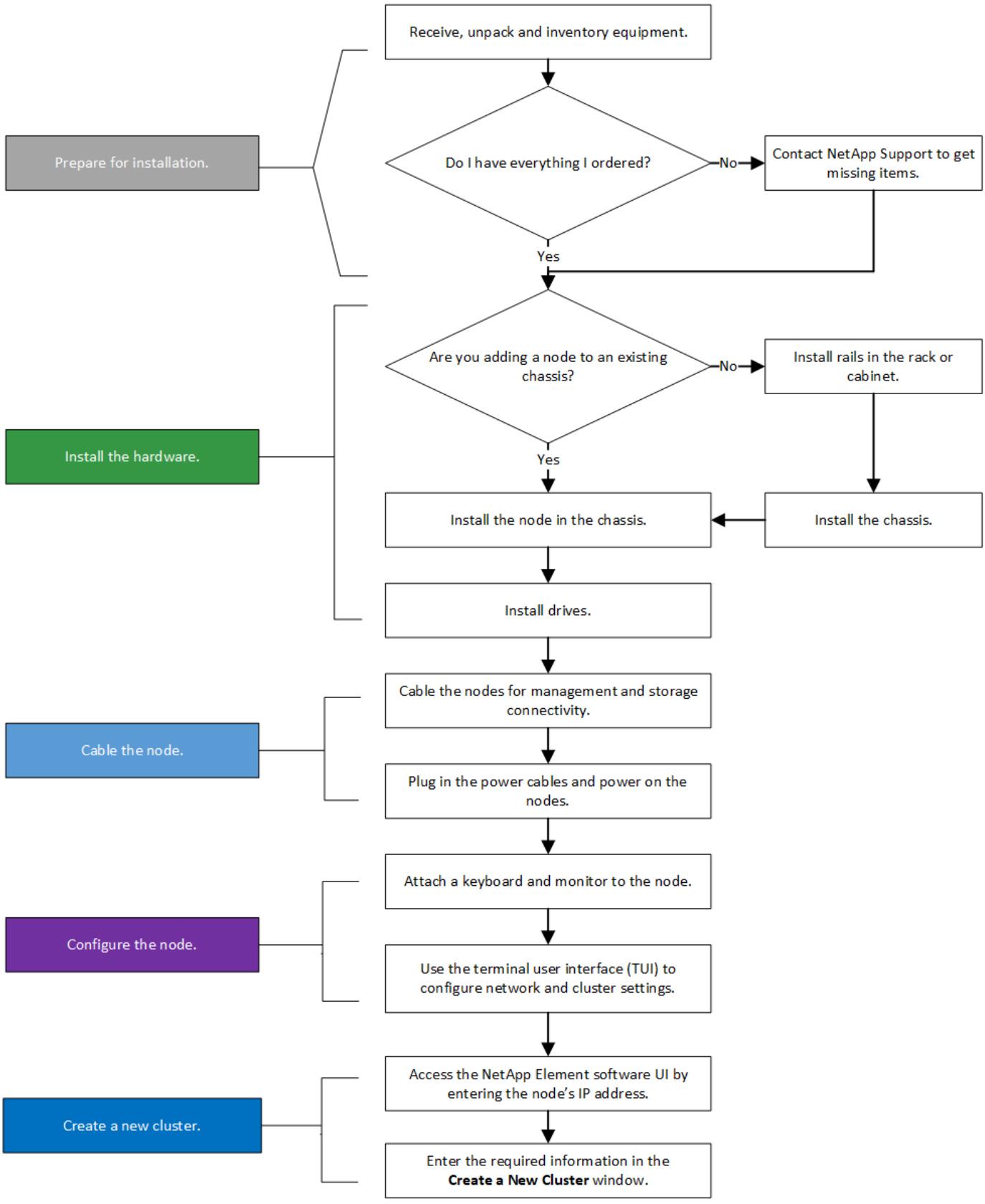
See the poster for a visual representation of the instructions.

- Workflow diagrams
- Prepare for installation
- Install the rails
- Install and cable the nodes
- Configure the nodes
- Create a cluster

Workflow diagrams

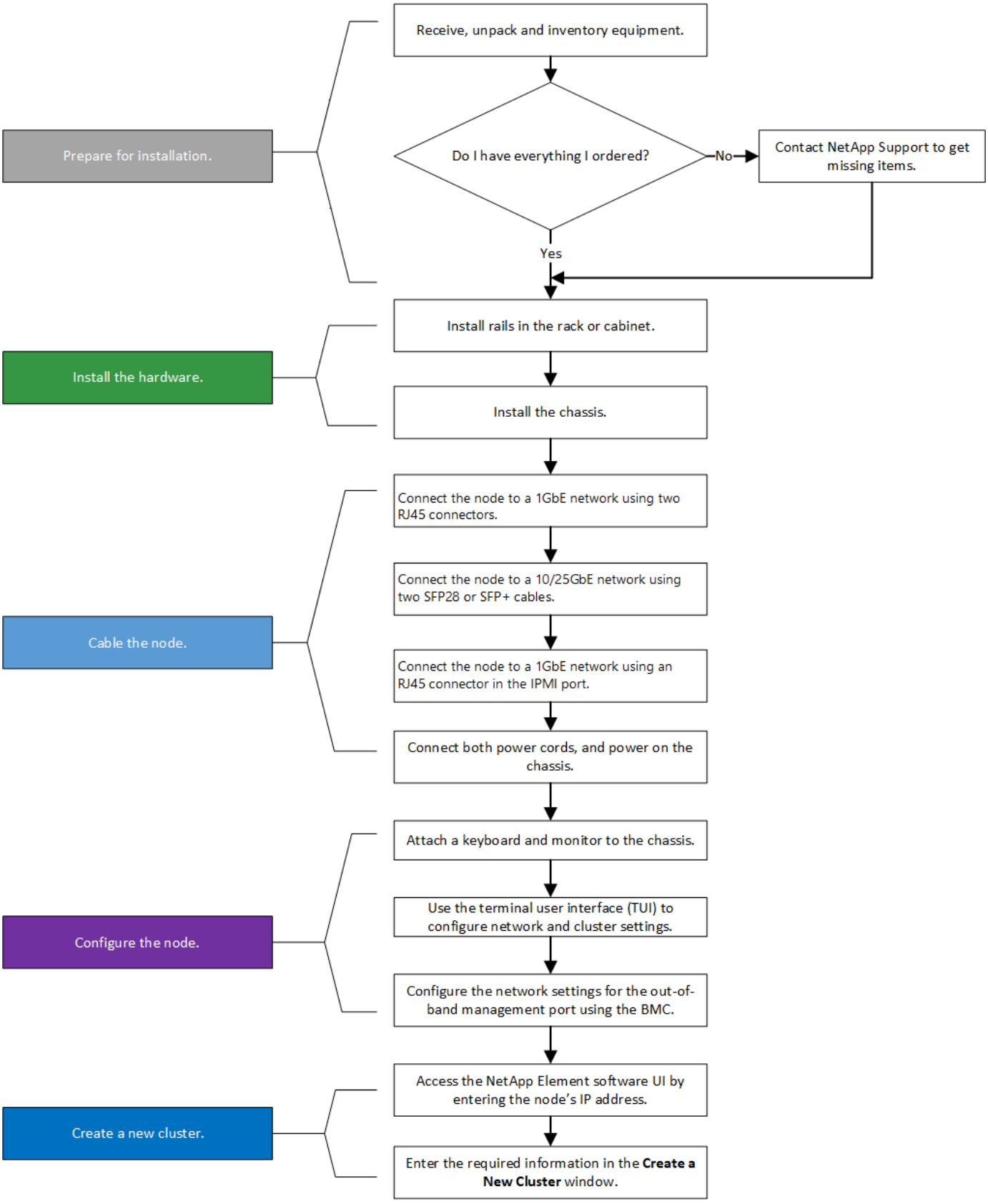
The workflow diagrams here provide a high-level overview of the installation steps. The steps vary slightly depending on the H-series model.

H410S



H610S

The terms "node" and "chassis" are used interchangeably in the case of H610S, because node and chassis are not separate components unlike in the case of a 2U, four-node chassis.



Prepare for installation

In preparation for installation, inventory the hardware that was shipped to you, and contact NetApp Support if any of the items are missing.

Ensure that you have the following items at your installation location:

- Rack space for the system.

Node type	Rack space
H410S nodes	Two rack unit (2U)
H610S nodes	One rack unit (1U)

- SFP28/SFP+ direct-attach cables or transceivers
- CAT5e or higher cables with RJ45 connector
- A keyboard, video, mouse (KVM) switch to configure your system
- USB stick (optional)

The hardware that is shipped to you depends on what you order. A new 2U, four-node order includes the chassis, bezel, slide rail kit, drives, storage nodes, and power cables (two per chassis). If you order H610S storage nodes, the drives come installed in the chassis.

While installing the hardware, ensure that you remove all packing material and wrapping from the unit. This will prevent the nodes from overheating and shutting down.

Install the rails

The hardware order that was shipped to you includes a set of slide rails. You will need a screwdriver to complete the rail installation. The installation steps vary slightly for each node model.

Install hardware from the bottom of the rack up to the top to prevent the equipment from toppling over. If your rack includes stabilizing devices, install them before you install the hardware.

- H410S
- H610S

H410S

H410S nodes are installed in 2U, four-node H-Series chassis, which is shipped with two sets of adapters. If you want to install the chassis in a rack with round holes, use the adapters appropriate for a rack with round holes. The rails for H410S nodes fit a rack between 29 inches and 33.5 inches in depth. When the rail is fully contracted, it is 28 inches long, and the front and rear sections of the rail are held together by only one screw.

If you install the chassis onto a fully contracted rail, the front and rear sections of the rail might separate.

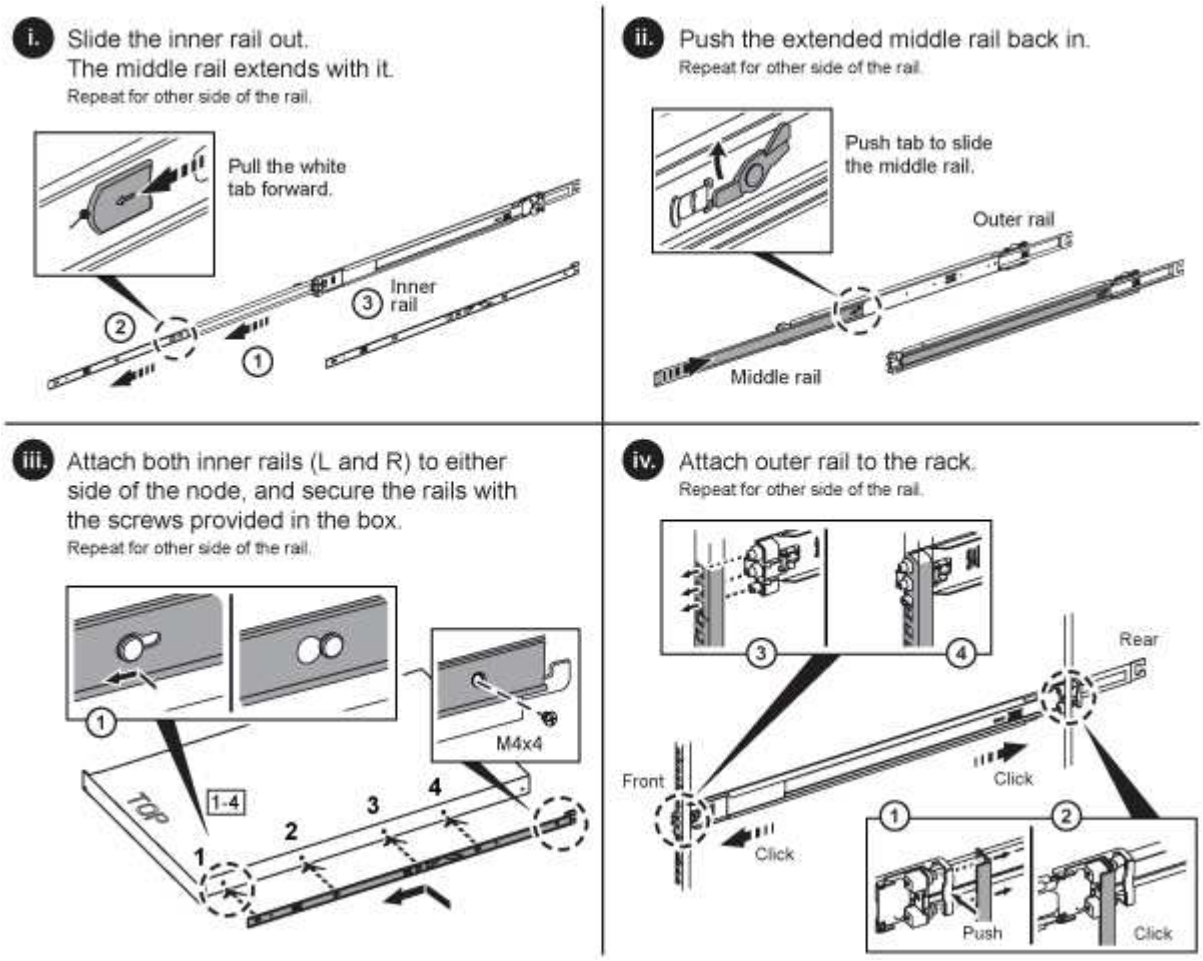
Steps

1. Align the front of the rail with the holes on the front post of the rack.
2. Push the hooks on the front of the rail into the holes on the front post of the rack and then down, until the spring-loaded pegs snap into the rack holes.
3. Attach the rail to the rack with screws. Here is an illustration of the left rail being attached to the front of the rack:

4. Extend the rear section of the rail to the rear post of the rack.
5. Align the hooks on the rear of the rail with the appropriate holes on the rear post ensuring that the front and the back of the rail are on the same level.
6. Mount the rear of the rail onto the rack, and secure the rail with screws.
7. Perform all the above steps for the other side of the rack.

H610S

Here is an illustration for installing rails for an H610S storage node:



There are left and right rails on the H610S. Position the screw hole towards the bottom so that the H610S thumbscrew can secure the chassis to the rail.

Install and cable the nodes

You install the H410S storage node in a 2U, four-node chassis. For H610S, install the chassis/node directly onto the rails in the rack.

Remove all the packing material and wrapping from the unit. This prevents the nodes from overheating and shutting down.

- H410S
- H610S

H410S

Steps

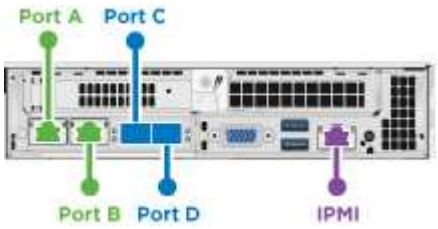
1. Install the H410S nodes in the chassis. Here is a rear-view example of a chassis with four nodes installed:

   Use caution while lifting the hardware and installing it into the rack. An empty two rack unit (2U), four-node chassis weighs 54.45 lb (24.7 kg) and a node weighs 8.0 lb (3.6 kg).

2. Install the drives.

3. Cable the nodes.

   If the airflow vents at the rear of the chassis are blocked by cables or labels, it can lead to premature component failures due to overheating.

   - Connect two CAT5e or higher cables in ports A and B for management connectivity.
   - Connect two SFP28/SFP+ cables or transceivers in ports C and D for storage connectivity.
   - (Optional, recommended) connect a CAT5e cable in the IPMI port for out-of-band management connectivity.

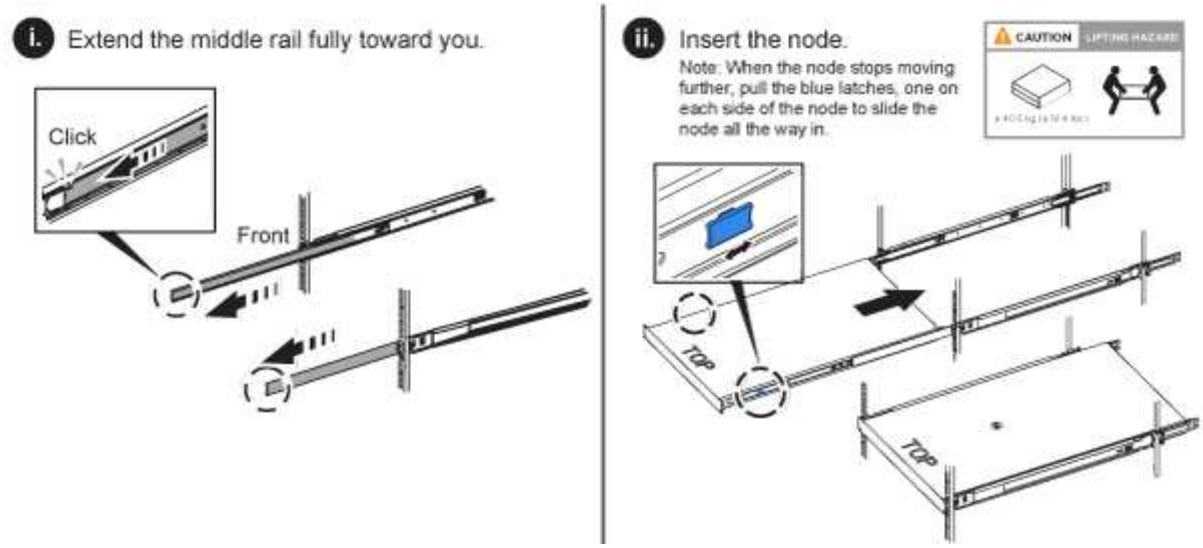
4. Connect the power cords to the two power supply units per chassis and plug them into 240V PDU or power outlet.
5. Power on the nodes.

   It takes approximately six minutes for the node to boot.

H610S

Steps

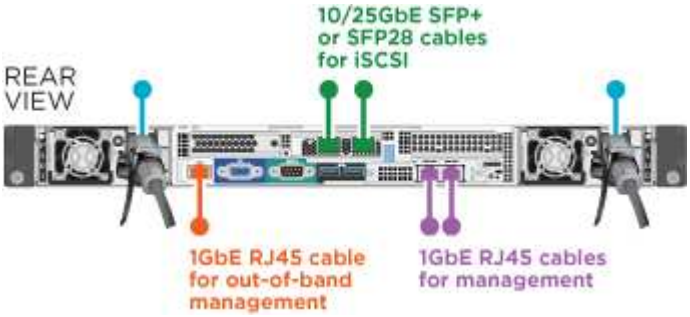
1. Install the H610S chassis. Here is an illustration for installing the node/chassis in the rack:



Use caution while lifting the hardware and installing it into the rack. An H610S chassis weighs 40.5 lb (18.4 kg).

2. Cable the nodes.

If the airflow vents at the rear of the chassis are blocked by cables or labels, it can lead to premature component failures due to overheating.



- Connect the node to a 10/25GbE network using two SFP28 or SFP+ cables.
- Connect the node to a 1GbE network using two RJ45 connectors.
- Connect the node to a 1GbE network using an RJ-45 connector in the IPMI port.
- Connect both power cables to the node.

3. Power on the nodes.

It takes approximately five minutes and 30 seconds for the node to boot.

Configure the nodes

After you rack and cable the hardware, you are ready to configure your new storage resource.

Steps

1. Attach a keyboard and monitor to the node.
2. In the terminal user interface (TUI) that is displayed, configure the network and cluster settings for the node by using the on-screen navigation.

   You should get the IP address of the node from the TUI. You need this when you add the node to a cluster. After you save the settings, the node is in a pending state, and can be added to a cluster. See the <insert link to Setup section>.

3. Configure out-of-band management using the Baseboard Management Controller (BMC). These steps apply **only to H610S** nodes.
   a. Use a web browser and navigate to the default BMC IP address: 192.168.0.120
   b. Log in using **root** as the username and **calvin** as the password.
   c. From the node management screen, navigate to **Settings** > **Network Settings**, and configure the network parameters for the out-of-band management port.

See this KB article (log in required).

Create a cluster

After you add the storage node to your installation and configure the new storage resource, you are ready to create a new storage cluster

Steps

1. From a client on the same network as the newly configured node, access the NetApp Element software UI by entering the node's IP address.
2. Enter the required information in the **Create a New Cluster** window.
   See the setup overview for more information.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

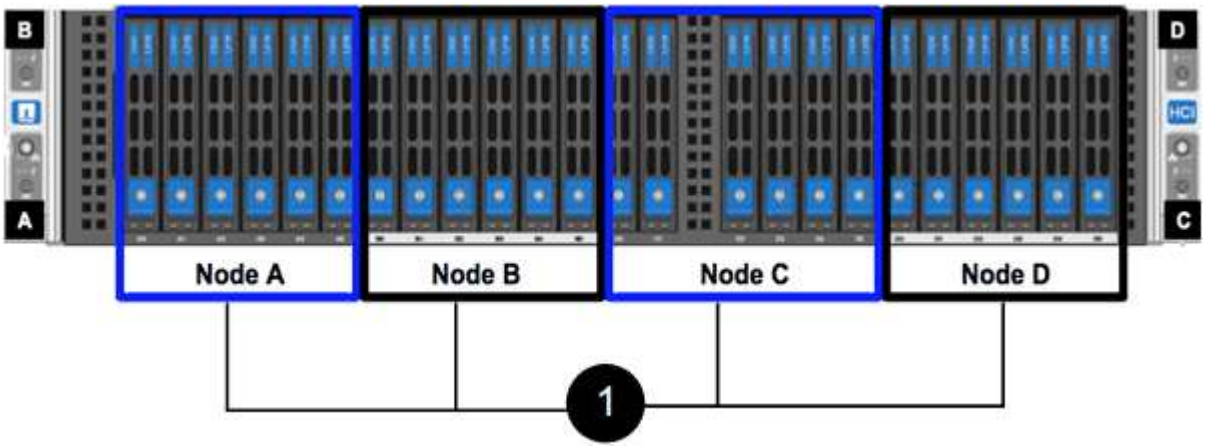
Replace a H410S node

You should replace a storage node in the event of CPU failure, Radian card problems, other motherboard issues, or if it does not power on. The instructions apply to H410S storage nodes.

Alarms in the NetApp Element software UI alert you when a storage node fails. You should use the Element UI to get the serial number (service tag) of the failed node. You need this information to locate the failed node in the cluster.

Here is the back of a two rack unit (2U), four-node chassis with four storage nodes:

Here is the front view of a four-node chassis with H410S nodes, showing the bays that correspond to each node:



What you'll need

- You have verified that your storage node is faulty and needs to be replaced.
- You have obtained a replacement storage node.
- You have an electrostatic discharge (ESD) wristband, or you have taken other antistatic protection.
- You have labeled each cable that is connected to the storage node.

Here is a high-level overview of the steps:

- Prepare to replace the node
- Replace the node in the chassis
- Add the node to the cluster

Prepare to replace the node

You should remove the faulty storage node correctly from the cluster in the NetApp Element software UI before you install the replacement node. You can do this without causing any service interruption. You should obtain the serial number of the faulty storage node from the Element UI and match it with the serial number on the sticker at the back of the node.

Steps

1. In the Element UI, select **Cluster** > **Drives**.
2. Remove the drives from the node using one of the following methods:

Option	Steps
To remove individual drives	a. Click **Actions** for the drive you want to remove. b. Click **Remove**.
To remove multiple drives	a. Select all the drives you want to remove, and click **Bulk Actions**. b. Click **Remove**.

3. Select **Cluster** > **Nodes**.
4. Note the serial number (service tag) of the faulty node. You should match it with the serial number on the sticker at the back of the node.
5. After you note the serial number, remove the node from the cluster as follows:
   a. Select the **Actions** button for the node you want to remove.
   b. Select **Remove**.

Replace the node in the chassis

After you remove the faulty node from the cluster using the NetApp Element software UI you are ready to physically remove the node from the chassis. You should install the replacement node in the same slot in the chassis from which you removed the failed node.

Steps

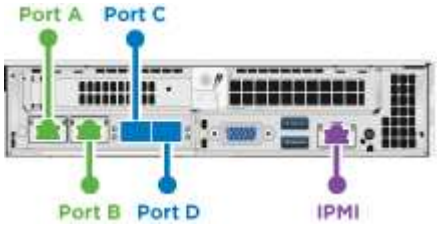
1. Wear antistatic protection before proceeding.
2. Unpack the new storage node, and set it on a level surface near the chassis.

   Keep the packaging material for when you return the faulty node to NetApp.

3. Label each cable that is inserted at the back of the storage node that you want to remove.

   After you install the new storage node, you should insert the cables into the original ports.

   Here is an image showing the back of a storage node:



Port	Details
Port A	1/10GbE RJ45 port
Port B	1/10GbE RJ45 port
Port C	10/25GbE SFP+ or SFP28 port

Port	Details
Port D	10/25GbE SFP+ or SFP28 port
IPMI	1/10GbE RJ45 port

4. Disconnect all the cables from the storage node.
5. Pull down the cam handle on the right side of the node, and pull the node out using both the cam handles.

   The cam handle that you pull down has an arrow on it to indicate the direction in which it moves. The other cam handle does not move and is there to help you pull the node out.

Item	Description
1	Cam handle to help you pull the node out.
2	Cam handle that you pull down before pulling the node out.

Support the node with both your hands when you pull it out of the chassis.

6. Place the node on a level surface.

   You must package the node and return it to NetApp.

7. Install the replacement node in the same slot in the chassis.

Ensure that you do not use excessive force when sliding the node into the chassis.

8. Move the drives from the node you removed and insert them in the new node.
9. Reconnect the cables to the ports from which you originally disconnected them.

   The labels you had on the cables when you disconnected them will help guide you.

1. If the airflow vents at the rear of the chassis are blocked by cables or labels, it can lead to premature component failures due to overheating.
2. Do not force the cables into the ports; you might damage the cables, ports, or both.

Ensure that the replacement node is cabled in the same way as the other nodes in the chassis.

10. Press the button at the front of the node to power it on.

Add the node to the cluster

When you add a node to the cluster or install new drives in an existing node, the drives automatically register as available. You must add the drives to the cluster by using either the Element UI or API before they can participate in the cluster.

The software version on each node in a cluster should be compatible. When you add a node to a cluster, the cluster installs the cluster version of Element software on the new node as needed.

Steps

1. Select **Cluster** > **Nodes**.
2. Select **Pending** to view the list of pending nodes.
3. Do one of the following:
   - To add individual nodes, select the **Actions** icon for the node you want to add.
   - To add multiple nodes, select the check box of the nodes to add, and then **Bulk Actions**.

   If the node you are adding has a different version of Element software than the version running on the cluster, the cluster asynchronously updates the node to the version of Element software running on the cluster master. After the node is updated, it automatically adds itself to the cluster. During this asynchronous process, the node will be in a `pendingActive` state.

4. Select **Add**.

   The node appears in the list of active nodes.

5. From the Element UI, select **Cluster** > **Drives**.
6. Select **Available** to view the list of available drives.
7. Do one of the following:
   - To add individual drives, select the **Actions** icon for the drive you want to add, and then select **Add**.
   - To add multiple drives, select the check boxes of the drives to add, select **Bulk Actions**, and then select **Add**.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Replace a H610S node

You might need to replace the chassis if the fan, central processing unit (CPU), or dual inline memory module (DIMM) fails, or to fix overheating issues or problems with the boot process. The blinking amber LED in the front of the chassis is an indication of a possible need for chassis replacement. You should contact NetApp Support before you proceed.

See the KB article for information about installation requirements for H610S nodes. New and spare H610S storage nodes might have additional installation requirements based on the existing Element software version of the storage cluster. Contact NetApp Support for more information.

The terms "node" and "chassis" are used interchangeably in the case of H610S, which is a one rack unit (1U) chassis.

Best practices for adding and removing drives

You should follow these best practices for adding drives to the cluster:

- Add all the block drives and ensure that block syncing is complete before you add the slice drives.
- For Element software 10.x and later, add all the block drives at once. Ensure that you don’t this for more than three nodes at once.
- For Element software 9.x and earlier, add three drives at once allowing them to completely sync before adding the next group of three.
- Remove the slice drive and ensure that slice syncing is complete before removing the block drives.
- Remove all the block drives from a single node at once. Ensure that all block syncing is complete before you move on to the next node.

What you’ll need

- You have contacted NetApp Support.
  If you are ordering a replacement, you should have a case open with NetApp Support.
- You have obtained the replacement node.
- You have an electrostatic discharge (ESD) wristband, or you have taken other antistatic protection.
- If you need to perform the Return to Factory Image (RTFI) process, you have obtained the USB key. NetApp Support can help you decide if you need to perform the RTFI process.
- You have a keyboard and monitor.
- You have removed the failed node correctly from the cluster.
- If a DIMM has failed, you have removed drives before you remove the node from the cluster.

About this task

Alarms in the Element UI alert you when a host fails. You must match the serial number of the failed host from the VMware vSphere Web Client with the serial number on the sticker at the back of the node.

Steps

1. Locate the service tag at the front of the failed chassis.

2. Verify that the serial number on the service tag matches the NetApp Support case number when you ordered the replacement chassis.
3. Plug in the keyboard and monitor to the back of the failed chassis.
4. Verify the serial number of the failed node with NetApp Support.
5. Power down the chassis.
6. Label the drives in the front and cables at the back with their locations, so that you can put them back in the same locations after the replacement.
   See the following image for the placement of the drives in the chassis:

7. Remove the cables.
8. Remove the chassis by unscrewing the thumbscrews on the mounting ears.
   You should package and return the failed chassis to NetApp.
9. Install the replacement chassis.
10. Remove the drives carefully from the failed chassis, and insert them in the replacement chassis.

    You should insert the drives in the same slots they were in before you removed them.

11. Remove the power supply units from the failed chassis, and insert them in the replacement chassis.
12. Insert the power supply cables, and the network cables in their original ports.
13. Small form-factor pluggable (SFP) transceivers might be inserted in the 10GbE ports of the replacement node. You should remove them before you cable the 10GbE ports.

    See your switch vendor's documentation if your switch does not recognize the cables.

14. Power on the chassis by pressing the power button at the front.
    It takes approximately five minutes and 30 seconds for the node to boot.
15. Perform the configuration steps.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Replace drives

If a drive is faulty or if the drive wear level falls below a threshold, you should replace it. Alarms in the Element software UI notify you when a drive has failed or is going to fail. You can hot-swap a failed drive.

About this task

This procedure is for replacing drives in H410S and H610S storage nodes. Removing a drive takes the drive offline. Any data on the drive is removed and migrated to other drives in the cluster. The data migration to other active drives in the system can take a few minutes to an hour depending on capacity utilization and active I/O on the cluster.
You should follow these best practices for handling drives while removing and replacing them:

- Keep the drive in the ESD bag until you are ready to install it.
- Open the ESD bag by hand or cut the top off with a pair of scissors.
- Always wear an ESD wrist strap grounded to an unpainted surface on your chassis.
- Always use both hands when removing, installing, or carrying a drive.
- Never force a drive into the chassis.
- Always use approved packaging when shipping drives.
- Do not stack drives on top of each other.

Best practices for adding and removing drives

- Add all the block drives and ensure that block syncing is complete before you add the slice drives.
- For Element software 10.x and later, add all the block drives at once. Ensure that you do not do this for more than three nodes at once.
- For Element software 9.x and earlier, add three drives at once allowing them to completely sync before adding the next group of three.
- Remove the slice drive and ensure that slice syncing is complete before removing the block drives.
- Remove all the block drives from a single node at once. Ensure that all block syncing is complete before you move on to the next node.

Steps

1. Remove the drive from the cluster using the NetApp Element software UI:
    a. From the Element UI, Select **Cluster > Drives**.
    b. Select **Failed** to view the list of failed drives.
    c. Make a note of the slot number of the failed drive. You need this information to locate the failed drive in

the chassis.

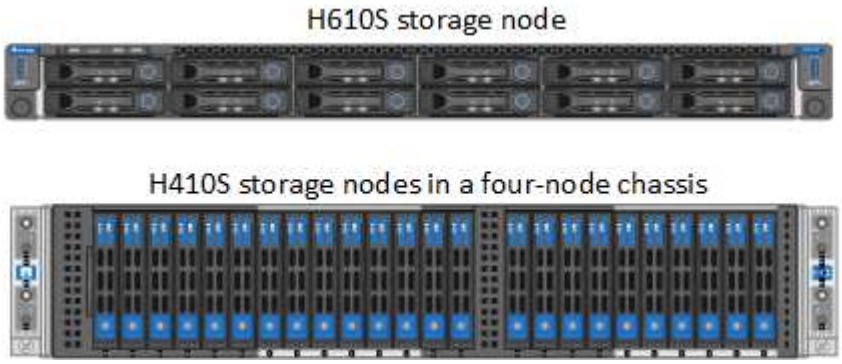
d. Select **Actions** for the drive you want to remove.

e. Select **Remove**.

> If there is not enough capacity to remove active drives before removing a node, an error message appears when you confirm the drive removal. After you resolve the error, you can now physically remove the drive from the chassis.

2. Replace the drive from the chassis:

a. Unpack the replacement drive, and place it on a flat, static-free surface near the rack.
Save the packing materials for when you return the failed drive to NetApp.
Here is the front view of the H610S and H410S storage nodes with the drives:

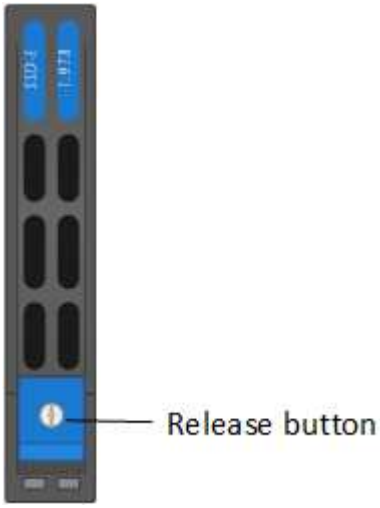


b. (**H410S only**) Perform the following steps:

i. Identify the node by matching the serial number (service tag) with the number you noted down from the Element UI.
The serial number is on a sticker at the back of each node.
After you identify the node, you can use the slot information to identify the slot that the failed drive is in. Drives are arranged alphabetically from A through D and from 0 through 5.

ii. Remove the bezel.

iii. Press the release button on the failed drive:



When you press the release button, the cam handle on the drive springs open partially, and the drive releases from the midplane.

iv. Open the cam handle, and slide the drive out carefully using both hands.

v. Place the drive on an antistatic, level surface.

vi. Insert the replacement drive into the slot all the way into the chassis using both hands.

vii. Press down the cam handle until it clicks.

viii. Reinstall the bezel.

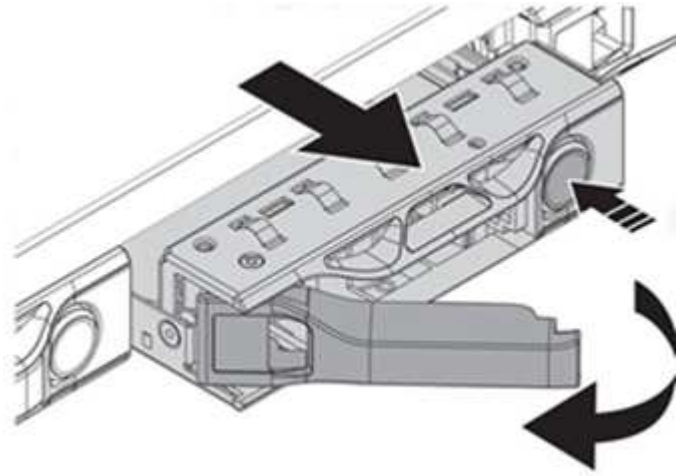
ix. Notify NetApp Support about the drive replacement.
NetApp Support will provide instructions for returning the failed drive.

c. (**H610S only**) Perform the following steps:

i. Match the slot number of the failed drive from the Element UI with the number on the chassis.
The LED on the failed drive is lit amber.

ii. Remove the bezel.

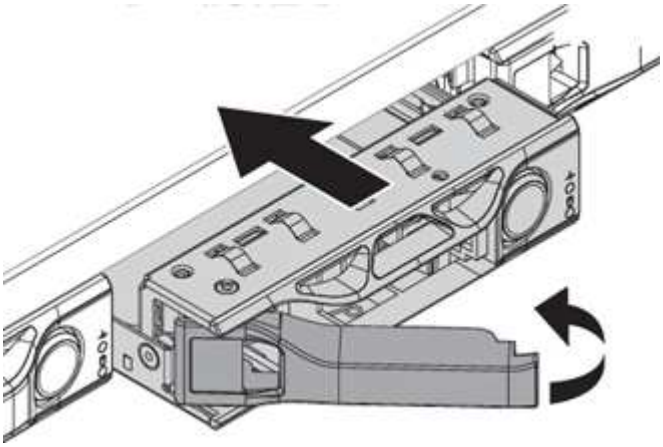
iii. Press the release button, and remove the failed drive as shown in the following illustration:



Ensure that the tray handle is fully open before you attempt to slide the drive out of the chassis.

iv. Slide the drive out, and place it on a static-free, level surface.

v. Press the release button on the replacement drive before you insert it into the drive bay.
The drive tray handle springs open.



vi. Insert the replacement drive without using excessive force.
When the drive is inserted fully, you hear a click.

vii. Close the drive tray handle carefully.

viii. Reinstall the bezel.

ix. Notify NetApp Support about the drive replacement.
NetApp Support will provide instructions for returning the failed drive.

3. Add the drive back to the cluster using the Element UI.

> When you install a new drive in an existing node, the drive automatically registers as **Available** in the Element UI. You should add the drive to the cluster before it can participate in the cluster.

a. From the Element UI, Select **Cluster > Drives**.

b. Select **Available** to view the list of available drives.

c. Select the Actions icon for the drive you want to add, and select **Add**.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Replace a power supply unit

Each chassis includes two power supply units for power redundancy. If a power supply unit is faulty, you should replace it as soon as possible to ensure that the chassis has a redundant power source.

What you'll need

- You have determined that the power supply unit is faulty.
- You have a replacement power supply unit.
- You have verified that the second power supply unit is operating.
- You have an electrostatic discharge (ESD) wristband, or you have taken other antistatic precautions.

About this task

The replacement procedure applies to the following node models:

- Two rack unit (2U), four-node NetApp HCI chassis
- One rack unit (1U) H610S storage chassis

> In the case of H610S, the terms "node" and "chassis" are used interchangeably because node and chassis are not separate components, unlike in the case of the 2U, four-node chassis.

Alarms in the Element UI provide information about the failed power supply unit, referring to it as PS1 or PS2. In a NetApp HCI 2U, four-node chassis, PS1 refers to the unit on the top row of the chassis and PS2 refers to the unit on the bottom row of the chassis. You can replace the faulty power supply unit while your chassis is

powered on and working, as long as the redundant power supply unit is functioning.

> If you are replacing both PSUs in a node, the PSUs must have the same part number and wattage. Mismatched PSUs could damage the system.

Steps

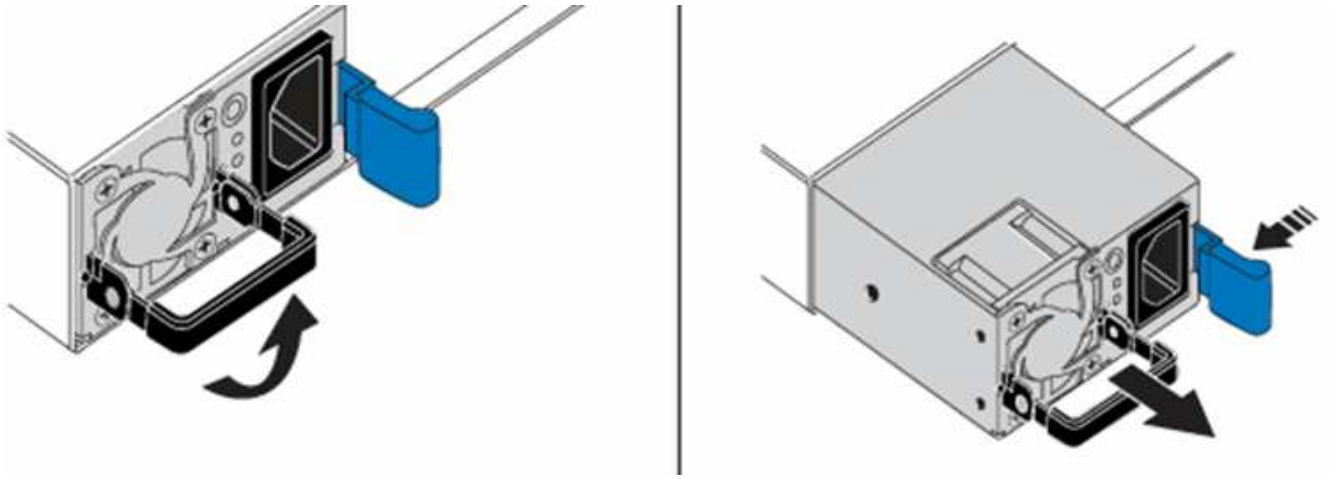
1. Locate the faulty power supply unit in the chassis. The LED on the faulty unit displays amber.

   > The power supply units are located differently based on the type of chassis.

   See the images below for the locations of the power supply units:

Model	Location of the power supply units
2U, four-node NetApp HCI storage chassis	The nodes in your chassis might look different depending on the type of nodes (storage or compute) you have.
H610S chassis	

2. Identify the correct node using the blue pullout tag or serial number. The blue pull-out tag lists the serial number (S/N) and drive layout. Confirm the node's serial number to be serviced.
   - If you are replacing both power supply units, continue to step 3.
   - If you are replacing only one power supply unit, skip to step 4.
3. Confirm that the node has been powered down or is ready to be powered down for service. Note the following:
   - A node that has been powered down does not display any blue power LEDs on the drives or power button.
   - A node that has not yet been shut down displays blue LEDs on the drives and power button.
   - A node that has been shut down and is ready for service displays a blinking PSU LED that blinks on (green) and off (no color).
   - A node that has not yet been shut down displays solid green LEDs on the power supplies.
4. Unplug the power cord from the power supply unit or both power cords, if replacing both units.
5. Lift the cam handle, and press the blue latch to slide out the power supply unit.



The illustration is an example. The location(s) of the power supply unit(s) in the chassis and the color of the release button vary depending on the type of chassis you have.

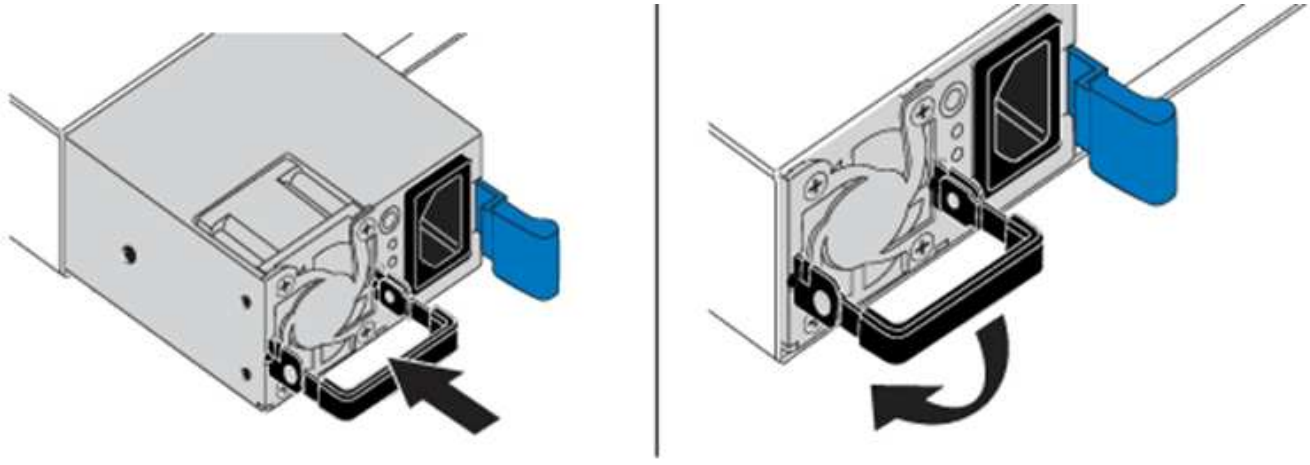
Ensure that you use both hands to support the weight of the power supply unit.

Repeat steps 3, 4, and 5 if replacing a second power supply unit.

6. Locate the label on the power supply unit that you removed from the chassis. The label contains details of the manufacturer and output wattage.

   Do not replace the power supply unit if the wattage of the power supply from your RMA does not match the wattage of the removed power supply. Contact NetApp Support for next steps.

7. Using both hands, align the edges of the power supply unit with the opening in the chassis, gently push the unit into the chassis using the cam handle until it locks into place, and return the cam handle to the upright position.



8. Plug in one or both power cords.
9. If you replaced both power supply units, go to the front of the node and press the power button to power on the nodes. After power on, the power button LED illuminates a solid blue color. The blue LEDs for the drives and the identification button will start blinking.
10. Return the faulty unit to NetApp by following the instructions in the box that was shipped to you.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

SF-series hardware information

You can find information about installing and maintaining SF-series storage nodes.

Here are the links to the installation and maintenance content:

- Install and setup SolidFire C-series nodes
- Install and setup Fibre Channel nodes
- Install and setup SF-series storage nodes
- Replace a chassis
- Replace drives
- Replace a power supply unit

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Replace a chassis

You might need to replace the chassis if the fan, central processing unit (CPU), or dual inline memory module (DIMM) fails, or to fix overheating issues or problems with the boot process. Cluster faults in the NetApp Element software user interface (UI) and the blinking amber light in the front of the chassis are indications of a possible need for chassis replacement. You should contact NetApp Support before you proceed.

What you'll need

- You have contacted NetApp Support.

  If you are ordering a replacement, you must have a case opened with NetApp Support.

- You have obtained the replacement chassis.
- You have an electrostatic discharge (ESD) wristband, or you have taken other antistatic protection.
- If you need to perform the Return to Factory Image (RTFI) process, you have obtained the USB key.

  NetApp Support will help you decide if RTFI is needed. See this KB article (login required).

- You have a keyboard and monitor.

About this task

The instructions in this document apply if you have a one-rack unit (1U) chassis with any of the following nodes:

- SF2405
- SF4805
- SF9605
- SF9608
- SF19210
- SF38410
- SF-FCN-01
- FC0025

Depending on your Element software version, the following nodes are not supported:

- Beginning with Element 12.7, SF2405 and SF9608 storage nodes and FC0025 and SF-FCN-01 FC nodes.
- Beginning with Element 12.0, SF3010, SF6010, and SF9010 storage nodes.

Steps

1. Locate the service tag of the failed chassis and verify that the serial number matches the number on the case you opened with NetApp Support when you ordered the replacement.

   You can locate the service tag from the front of the chassis.

   The following figure is an example of the service tag:

   The above figure is an example. The exact location of the service tag might vary depending on your hardware model.

2. Plug in the keyboard and monitor to the back of the failed chassis.
3. Verify chassis information with NetApp Support.
4. Power down the chassis.
5. Label the drives in the front of the chassis and cables at the back.

Fibre Channel nodes do not have drives in the front.

6. Remove the power supply units and cables.
7. Remove the drives carefully, and place them on an antistatic, level surface.

   If you have a Fibre Channel node, you can skip this step.

8. Remove the chassis by pressing the latch or unscrewing the thumbscrew, based on your hardware model.

   You should package and return the failed chassis to NetApp.

9. **Optional**: Remove the rails and install the new rails that were shipped with your replacement chassis.

   You can choose to reuse the existing rails. If you are reusing the existing rails, you can skip this step.

10. Slide the replacement chassis on to the rails.
11. For storage nodes, insert the drives from the failed chassis to the replacement chassis.

    You should insert the drives in the same slots as they were in the failed chassis.

12. Install the power supply units.
13. Insert the power supply cables, and the 1GbE and 10GbE cables in their original ports.

    Small form-factor pluggable (SFP) transceivers might be inserted in the 10GbE ports of the replacement chassis. You should remove them before you cable the 10GbE ports.

14. If you have determined that you do not need to perform the RTFI process on the node, boot the node, and wait until the terminal user interface (TUI) appears. Proceed to step 16 and allow the cluster to re-image the node automatically when you add it by using the UI.
15. **Optional**: If NetApp Support recommends re-imaging the node with a USB key, perform the following substeps:
    a. Power on the chassis. It boots with the RTFI key image.
    b. At the first prompt, type **Y** to image the storage node.
    c. At the second prompt, type **N** for hardware health checks.

       If the RTFI script detects a problem with a hardware component, it displays an error in the console. If you see an error, contact NetApp Support. After the RTFI process completes, the node shuts down.

    d. Remove the USB key from the USB slot.
    e. Boot the newly imaged node, and wait for the TUI to appear.
16. Configure the network and cluster information from the TUI.

    You can contact NetApp Support for assistance.

17. Add the new node to the cluster using the cluster TUI.
18. Pack and return the failed chassis.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Replace drives for SF-series storage nodes

You can hot-swap a failed solid-state drive with a replacement drive.

What you'll need

- You have a replacement drive.
- You have an electrostatic discharge (ESD) wristband, or you have taken other antistatic precautions.
- You have contacted NetApp Support to verify that the SSD needs to be replaced and for help with the proper resolution procedure.

  You will need the service tag or serial number when you call NetApp Support. Support will work with you to get a replacement drive according to your Service Level Agreement.

About this task

The instructions apply to the following SolidFire storage node models:

- SF2405
- SF4805
- SF9605
- SF9608
- SF19210
- SF38410

Depending on your Element software version, the following nodes are not supported:

- Beginning with Element 12.7, SF2405 and SF9608 storage nodes.
- Beginning with Element 12.0, SF3010, SF6010, and SF9010 storage nodes.

The following figure shows the placement of the drives in an SF9605 chassis:

The above figure is an example. SF9608 has a different drive layout that includes only eight drives that are numbered one through eight, from left to right.

Slot 0 holds the metadata drive for the node. If you are replacing the drive in slot 0, you must attach the sticker included in the shipping box on the replacement drive, so that you can identify it separately from the rest.

Follow these best practices while handling drives:

- Prevent electrostatic discharge (ESD) by keeping the drive in the ESD bag until you are ready to install it.
- Do not insert a metal tool or knife into the ESD bag.
- Open the ESD bag by hand or cut the top off with a pair of scissors.
- Keep the ESD bag and any packing materials in case you must return a drive later.
- Always wear an ESD wrist strap grounded to an unpainted surface on your chassis.
- Always use both hands when removing, installing, or carrying a drive.
- Never force a drive into the chassis.
- Do not stack drives on top of each other.
- Always use approved packaging when shipping drives.

Here is a high-level overview of the steps:

- Remove the drive from the cluster
- Replace the drive from the chassis
- Add the drive to the cluster

Remove the drive from the cluster

The SolidFire system puts a drive in a failed state if the drive's self-diagnostics tells the node it has failed or if communication with the drive stops for five and a half minutes or longer. The system displays a list of the failed drives. You should remove a failed drive from the failed drive list in NetApp Element software.

Steps

1. In the Element UI, select **Cluster** > **Drives**.
2. Select **Failed** to view the list of failed drives.
3. Note the slot number of the failed drive.

   You need this information to locate the failed drive in the chassis.

4. Remove the failed drive using one of the following methods:

Option	Steps
To remove individual drives	a. Select **Actions** for the drive you want to remove. b. Select **Remove**.
To remove multiple drives	a. Select all the drives you want to remove, and the select **Bulk Actions**. b. Select **Remove**.

Replace the drive from the chassis

After you remove a failed drive from the failed drive list in the Element UI, you are ready to physically replace the failed drive from the chassis.

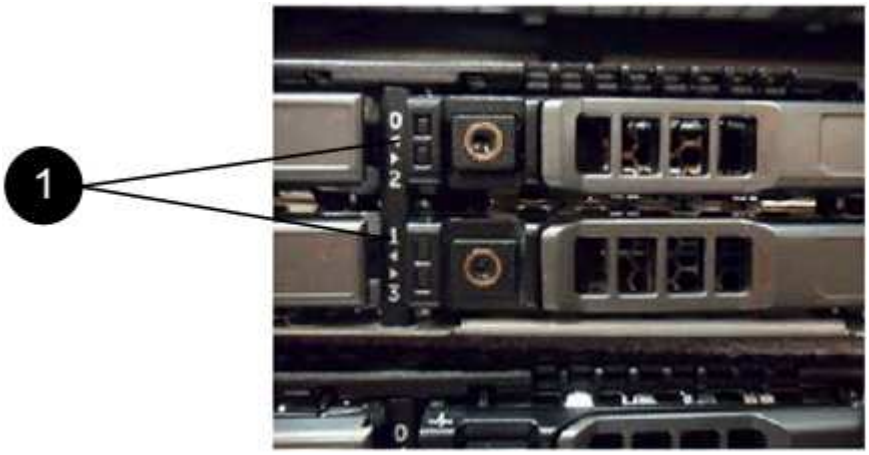
Steps

1. Unpack the replacement drive, and place it on a flat, static-free surface near the rack.

   Save the packing materials for when you return the failed drive to NetApp.

2. Match the slot number of the failed drive from the Element UI with the number on the chassis.

   The following figure is an example to show the numbering of the drive slots:



Item	Description
1	Drive slot numbers

3. Press the red circle on the drive you want to remove to release the drive.

   The latch clicks open.

4. Slide the drive out of the chassis, and place it on a static-free, level surface.
5. Press the red circle on the replacement drive before you slide it into the slot.
6. Insert the replacement drive, and press the red circle to close the latch.
7. Notify NetApp Support about the drive replacement.

   NetApp Support will provide instructions for returning the failed drive.

Add the drive to the cluster

After you install a new drive in the chassis, it registers as available. You should add the drive to the cluster using the Element UI before it can participate in the cluster.

Steps

1. In the Element UI, click **Cluster** > **Drives**.
2. Click **Available** to view the list of available drives.

3. Choose one of the following options to add drives:

Option	Steps
To add individual drives	a. Select the **Actions** button for the drive you want to add. b. Select **Add**.
To add multiple drives	a. Select the check boxes of the drives to add, and then select **Bulk Actions**. b. Select **Add**.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Replace a power supply unit

Each SolidFire chassis includes two power supply units for power redundancy. If a power supply unit fails, you should replace it as soon as possible to ensure that the chassis has a redundant power source.

What you'll need

- You have determined that the power supply unit needs to be replaced.
- You have a replacement power supply unit.
- You have verified that the second power supply unit is operating.
- You have an electrostatic discharge (ESD) wristband, or you have taken other antistatic precautions.

About this task

The instructions apply if you have a one-rack unit (1U) chassis with any of the following nodes:

- SF2405
- SF4805
- SF9605
- SF9608
- SF19210
- SF38410
- SF-FCN-01
- FC0025

Depending on your Element software version, the following nodes are not supported:

- Beginning with Element 12.7, SF2405 and SF9608 storage nodes and FC0025 and SF-FCN-01 FC nodes.
- Beginning with Element 12.0, SF3010, SF6010, and SF9010 storage nodes.

Steps

1. Unplug the power cord from the power supply unit that you are replacing.
2. Press the release button to slide the power supply unit out of the chassis.

   Ensure that you use both hands to support the weight of the power supply unit.

3. Using both hands, align the edges of the replacement power supply unit with the opening in the chassis, and gently push the unit into the chassis.

   Do not use excessive force when sliding the power supply unit into the chassis to prevent damage to the hardware.

4. Plug in the power cord.
5. Return the failed unit to NetApp by following the instructions in the box that was shipped to you.

   You can contact NetApp Support for help with the replacement procedure.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Return to Factory Image information

Configure the Return to Factory Image

NetApp SolidFire storage systems use the Return To Factory Image (RTFI) process to write a software image to a new node or restore a node to its original factory state. The RTFI process securely erases all existing data and configurations (if any) and installs an unconfigured NetApp Element software image. The RTFI process is available for all SolidFire nodes.

SolidFire systems use one RTFI process for all Element software installations. This includes internal manual installations performed by developers, automatic installations by automated framework tests, field installations by service engineers and customers, and installations performed by various integrators and partners. The same RTFI process is used on all SolidFire nodes, regardless of the chassis or node type in use, to automatically fix any issues.

The intended audience for this guide is integrators who install, configure, use, or troubleshoot storage-related issues.

- Linux: You have some background with Linux systems.

- Networking: You have a familiarity with server networking and networked storage, including IP addresses, netmasks, and gateways.

> The RTFI process is data destructive and securely erases all data and configuration details from the node and installs a new operating system. Verify that the node used for the RTFI process is not active as part of a cluster.

Deploy and install the RTFI International Organization for Standardization (ISO) image and perform the RTFI process:

- RTFI deployment and installation options
- Perform the RTFI process
- RTFI options menu

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

RTFI deployment and installation options

The Return To Factory Image (RTFI) process uses a bootable, installable media with a completely self-contained, minimalistic Linux OS to deploy Element software on a node. You can download the RTFI ISO image specific to your Element software version from the NetApp support site.

After you download the RTFI ISO image, you can deploy it according to one of the following commonly used methods:

- **Physical USB key**: You can write a bootable Element software ISO to a USB key. For instructions, see the Knowledge Base article How to create an RTFI key to re-image a SolidFire storage node. Insert the USB key with the ISO into the node and boot from the USB key.
- **Virtual media using the baseboard management controller (BMC) management port**: You can use the BMC to dynamically attach to the ISO located on your client system. The ISO is made available to the host OS as a virtual drive (CD or DVD). For more information, see the Knowledge Base article How to RTFI a node via BMC.
- **Network boot using a Preboot Execution Environment (PXE), Trivial File Transfer Protocol (TFTP), or FTP**: Instead of manually unpacking an ISO image, you can use `autofs` to automatically extract an image when the RTFI process requests it. This deployment mechanism requires more initial setup but allows for correct automation and scalability of installation.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

The RTFI process

You can begin the Return to Factory Image (RTFI) process by interacting with the node

through text console prompts that appear before the system boots.

The RTFI process is data destructive and securely erases all data and configuration details from the node and installs a new operating system. Verify that the node used for the RTFI process is not active as part of a cluster.

The RTFI process performs the following high-level operations:

1. Starts the installation after user confirmation and validates the image.
2. Unlocks all drives on a node.
3. Validates and flashes firmware.
4. Checks hardware.
5. Tests hardware.
6. Secure erases all selected drives.
7. Partitions the root drive and creates file systems.
8. Mounts and unpacks the image.
9. Configures the host name, networking (Dynamic Host Configuration Protocol), default cluster configuration, and GRUB bootloader.
10. Stops all services, collects logs, and reboots.

To configure your node after the RTFI process successfully completes, see the documentation for your Element software version. After a node successfully completes the RTFI process, it transitions to the *available* (unconfigured) state by default.

Perform the RTFI process

Use the following procedure to restore the Element software on your SolidFire node.

For information on creating a USB key or using the BMC to perform the RTFI process, see RTFI deployment and installation options.

Before you begin

Verify that you meet the following requirements:

- You have access to a console for the SolidFire node.
- The node on which you are performing the RTFI process is powered up and connected to a network.
- The node on which you are performing the RTFI process is not part of an active cluster.
- You have access to bootable installation media that contains the image of the relevant Element software version for your configuration.

Contact NetApp Support if you have any concerns before performing the RTFI process.

Steps

1. Connect a monitor and keyboard to the back of the node, or connect to the BMC IP UI, and bring up the **iKVM/HTML5** console from the **Remote Control** tab in the UI.
2. Insert a USB key with an appropriate image in one of the two USB slots in the back of the node.
3. Power on or power reset the node. During the boot-up, select Boot Device by selecting **F11**:

   You must select **F11** multiple times in quick succession because the Boot Device screen goes by quickly.

4. In the Boot Device selection menu, highlight the USB option.

   The options that appear depend on the USB brand that you are using.

   If there are no USB devices listed, go into the BIOS, verify that the USB is listed in the boot order, reboot, and try again.

   If that does not resolve the issue, go into the BIOS, browse to the **Save and Exit** tab, select **Restore to Optimized Defaults**, accept and save the settings, and reboot.

5. A list of the images that are on the highlighted USB device appear. Select the desired version and select enter to start the RTFI process.

   The RTFI image Element software name and version number appear.

6. At the initial prompt, you are notified that the process will remove all data from the node and that data is not recoverable after the process begins. Enter **Yes** to begin.

   All data and configuration details are permanently erased from the node after the process is initiated. If you elect not to proceed, you are directed to the RTFI options menu.

   If you want to watch the console during the RTFI process, you can press the **ALT+F8** keys to toggle to the verbose mode console. Press **ALT+F7** to return to the primary GUI.

7. Enter **No** when prompted to perform extensive hardware tests unless you have a reason to suspect hardware failure or are directed to perform the tests by NetApp Support.

A message indicates that the RTFI process has finished and the system powers off.

8. If necessary, remove all bootable installation media after the node powers off.

   The node is now ready to be powered on and configured. See the Element software set up storage documentation to configure the storage node.

   If you encountered an error message during the RTFI process, see RTFI options menu.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

RTFI options menu

The following options menu appears if the RTFI process is unsuccessful or if you elect not to proceed at the initial RTFI process prompt.

Contact NetApp Support before using any of the following command options.

Option	Description
Reboot	Exits the RTFI process and reboots the node in its current state. No cleanup is performed.
PowerOff	Gracefully powers off the node in its current state. No cleanup is performed.
Exit	Exits the RTFI process and opens a command prompt.
UploadLogs	Collects all logs on the system and uploads a single consolidated log archive to a specified URL.

Upload logs

Collect all logs on the system and upload them to a specified URL according to the following procedure.

Steps

1. At the RTFI options menu prompt, enter **UploadLogs**.
2. Enter the remote directory information:
   a. Type a URL that includes the protocol. For example: `ftp://,scp://,http://,orhttps://`.
   b. (Optional) Add an embedded user name and password. For example: `scp://user:password@URLaddress.com`.

      For a full range of syntax options, see the cURL user manual.

      The log file is uploaded and saved to the specified directory as a `.tbz2` archive.

Use the support tunnel

If you require technical support for your NetApp HCI system or SolidFire all-flash storage system, NetApp Support can connect remotely with your system. To start a session and gain remote access, NetApp Support can open a reverse Secure Shell (SSH) connection to your environment.

You can open a TCP port for an SSH reverse tunnel connection with NetApp Support. This connection enables NetApp Support to log in to your management node.

Before you begin

- For management services 2.18 and later, the capability for remote access is disabled on the management node by default. To enable remote access functionality, see Manage SSH functionality on the management node.
- If your management node is behind a proxy server, the following TCP ports are required in the sshd.config file:

TCP port	Description	Connection direction
443	API calls/HTTPS for reverse port forwarding via open support tunnel to the web UI	Management node to storage nodes
22	SSH login access	Management node to storage nodes or from storage nodes to management node

Steps

- Log in to your management node and open a terminal session.
- At a prompt, enter the following:

  ```
  rst -r sfsupport.solidfire.com -u element -p <port_number>
  ```

- To close the remote support tunnel, enter the following:

  ```
  rst --killall
  ```

- (Optional) Disable remote access functionality again.

SSH remains enabled on the management node if you do not disable it. SSH enabled configuration persists on the management node through updates and upgrades until it is manually disabled.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Storage nodes

The supported firmware versions for H-series and SolidFire storage nodes.

- H610S
- H410S
- SF38410, SF19210, SF9605, and SF4805

H610S

Model number (family portion): H610S
Full model numbers: H610S-1, H610S-1-NE, H610S-2, H610S-2-NE, H610S-4, H610S-4-NE, and H610S-2F

Component firmware managed by a Storage Firmware Bundle

During 11.x timeframe, NetApp Element software was the only way to release firmware. Starting with Element 12.0, the concept of a **Storage Firmware Bundle** was introduced and firmware updates were now possible by an independently released Storage Firmware Bundle or Storage Firmware Bundle included as part of an Element 12.x release.

A dash (**-**) in the following table indicates that the particular hardware component was NOT supported in that given release vehicle.

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (NSED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (NSED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (NSED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SKHynix PE8010 (SED)	Drive SKHynix PE8010 (NSED)
Storage Firmware Bundle 2.182.0	10/17/2024	3B14	4.01.07	122	14.25.1020	16.32.1010	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.5	2.17	CXV8202Q	CXV8501Q	EDA5602Q	EDA5900Q	0109	0109	0108	GDC5A02Q	11093A10	110B3A10
Storage Firmware Bundle 2.175.0	06/15/2023	3B11	3.94.07	122	14.25.1020	16.32.1010	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.5	2.17	CXV8202Q	CXV8501Q	EDA5602Q	EDA5900Q	0109	0109	0108	GDC5602Q	11092A10	110B2A10

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (NSED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (NSED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (NSED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (NSED)
Storage Firmware Bundle 2.164.0	10/20/2022	3B11	3.94.07	122	14.25.1020	16.32.1010	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.3	2.16	CXV8202Q	CXV8501Q	EDA5602Q	EDA5900Q	0109	0109	0108	GDC5602Q	11092A10	110B2A10

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
Storage Firmware Bundle 2.164.0 through NetApp Element 12.7	10/20/2022	3B11	3.94.07	122	14.25.1020	16.32.1010	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.3	2.16	CXV8202Q	CXV8501Q	EDA5602Q	EDA5900Q	0109	0109	0108	GDC5602Q	11092A10	110B2A10

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
Storage Firmware Bundle 2.150.4	06/08/2022	3B11	3.94.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.3	2.16	CXV8202Q	CXV8501Q	EDA5602Q	EDA5900Q	0109	0109	0108	GDC5502Q	11092A10	110B2A10

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SKHynix PE8010 (SED)	Drive SKHynix PE8010 (N-SED)
Storage Firmware Bundle 2.150.4 through NetApp Element 12.5	06/08/2022	3B11	3.94.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.3	2.16	CXV82022Q	CXV85011Q	EDA56022Q	EDA59000Q	0109	0109	0108	GDC55022Q	11092A10	110B2A10

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
Storage Firmware Bundle 2.146.2	02/22/2022	3B11	3.94.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.3	2.16	CXV8202Q	CXV8501Q	EDA5602Q	EDA5900Q	0109	0109	0108	GDC5502Q	11092A10	110B2A10

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
Storage Firmware Bundle 2.9.4 through NetApp Element 12.3.2	09/16/2021	3B06	3.91.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.1	2.16	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
Storage Firmware Bundle 2.9.4 through NetApp Element 12.3.1.165	12/06/2021	3B06	3.91.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.1	2.16	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SKHynix PE8010 (SED)	Drive SKHynix PE8010 (N-SED)
Storage Firmware Bundle 2.99.2	12/06/2021	3B06	3.91.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.1	2.16	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen1)	Cache NVDIMM NVDIMM module Smart (Gen2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen2)	Cache NVDIMM NVDIMM module Micron (Gen1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen1)	Cache NVDIMM NVDIMM module Micron (Gen2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
StorageFirmwareBundle2.9.1throughNetAppElement12.3.1.103	09/16/2021	3B06	3.86.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.1	2.16	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SKHynix PE8010 (SED)	Drive SKHynix PE8010 (N-SED)
Storage Firmware Bundle 2.9 through NetApp Element 12.3	04/15/2021	3B06	3.86.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.1	2.16	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SKHynix PE8010 (SED)	Drive SKHynix PE8010 (N-SED)
Storage Firmware Bundle 2.7.6.8	02/03/2021	3B06	3.86.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-
Storage Firmware Bundle 2.7.1	09/29/2020	3B03	3.84.07	122	14.02.1002	-	3.1	2.16	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5302Q	EDA5600Q	0108	0108	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
Storage Firmware Bundle 2.7 6.8 through NetApp Element 12. 2.1	06/02/2021	3B06	3.86.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.1	2.16	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SKHynix PE8010 (SED)	Drive SKHynix PE8010 (N-SED)
Storage Firmware Bundle 2.21 through NetApp Element 12.2	09/29/2020	3B03	3.84.07	122	14.22.1002	-	3.1	2.16	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5302Q	EDA5600Q	0108	0108	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
Storage Firmware Bundle 2.7 6.8 through NetApp Element 12.0.1	06/02/2021	3B06	3.86.07	122	14.25.1020	-	3.1	2.16	26.2C	1.30	25.3C	1.40	1.10	3.1	2.16	CXV8202Q	CXV8501Q	EDA5402Q	EDA5700Q	0109	0109	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Aging at ech (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Aging at ech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Aging at ech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SKHynix PE8010 (SED)	Drive SKHynix PE8010 (N-SED)
Storage Firmware Bundle 1.2.17 through NetApp Element 12.0	03/20/2020	3B03	3.78.07	122	14.22.1002	-	3.1	2.16	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	0108	0108	0108	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
NetApp Element 11.8	03/11/2020	3B03	3.78.07	122	14.22.1002	-	3.1	2.16	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	0108	0108	0107	-	-	-
NetApp Element 11.7	11/21/2019	3A10	3.76.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	0108	0108	0107	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
NetApp Element 11.5.1	02/20/2020	3A08	3.76.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	0108	0108	0107	-	-	-
NetApp Element 11.5	09/26/2019	3A08	3.76.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	-	-	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	-	-	0107	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
NetApp Element 11.3.2	02/19/2020	3A08	3.76.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	0108	0108	-	-	-	-
NetApp Element 11.3.1	08/19/2019	3A08	3.76.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	-	-	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	-	-	-	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module Smart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 1)	Cache NVDIMM NVDIMM module Smart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) Smart (Gen 2)	Cache NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (N-SED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (N-SED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (N-SED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (N-SED)
NetApp Element 11.1.1	02/19/2020	3A06	3.70.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	0108	0108	-	-	-	-
NetApp Element 11.1	04/25/2019	3A06	3.70.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	-	-	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	-	-	-	-	-	-

Release Vehicle	Release Date	BIOS	BMC	CPLD	10/25 GbE NIC CX4	10/25 GbE NIC CX5	Cache NVDIMM NVDIMM module SMart (Gen 1)	Cache NVDIMM NVDIMM Energy Source (BPM) SMart (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module SMart (Gen 2)	Cache NVDIMM NVDIMM Energy Source (BPM) SMart (Gen 2)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 1)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 1)	Cache NVDIMM NVDIMM NVDIMM module Micron (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 2)	Cache NVDIMM NVDIMM Energy Source (PGEM) Agigatech (Gen 3)	Drive Samsung PM963 (SED)	Drive Samsung PM963 (NSED)	Drive Samsung PM983 (SED)	Drive Samsung PM983 (NSED)	Drive Kioxia CD5 (SED)	Drive Kioxia CD5 (NSED)	Drive CD5 (FIPS)	Drive Samsung PM9A3 (SED)	Drive SK Hynix PE8010 (SED)	Drive SK Hynix PE8010 (NSED)
NetApp Element 11.0.2	02/19/2020	3A06	3.70.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	25.3C	1.40	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	0108	0108	-	-	-	-
NetApp Element 11	11/29/2018	3A06	3.70.07	117	14.22.1002	-	2.C	2.07	26.2C	1.30	-	-	-	-	-	CXV8202Q	CXV8501Q	EDA5202Q	EDA5200Q	-	-	-	-	-	-

Component firmware not managed by a Storage Firmware Bundle

The following firmware is not managed by a Storage Firmware Bundle:

Component	Current version
1/10 GbE NIC	3.2d 0x80000b4b
Boot device	M161225i

H410S

Model Number (Family portion): H410S
Full Model Numbers: H410S-0, H410S-1, H410S-1-NE, and H410S-2

Component firmware managed by a Storage Firmware Bundle

Component firmware managed by a Storage Firmware Bundle.

Release Vehicle	Release Date	BIOS	BMC	10/25 GbE NIC SMCI Mellanox	Cache NVDIMM RMS200	Cache NVDIMM RMS300	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
Storage Firmware Bundle 2.182.0	10/17/2024	NAT3.6	07.02.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.175.0	06/15/2023	NAT3.4	07.02.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.164.0 through NetApp Element 12.7	10/20/2022	NAT3.4	6.98.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.164.0	10/20/2022	NAT3.4	6.98.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q

Release Vehicle	Release Date	BIOS	BMC	10/25 GbE NIC SMCI Mellanox	Cache NVDIMM RMS200	Cache NVDIMM RMS300	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
Storage Firmware Bundle 2.164.0 through NetApp Element 12.7	10/20/2022	NAT3.4	6.98.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.150.4 through NetApp Element 12.5	06/08/2022	NAT3.4	6.98.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.99 through NetApp Element 12.3	04/15/2021	NA2.1	6.84.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 2.76.8 through NetApp Element 12.2.1	06/02/2021	NA2.1	6.84.00	14.25.1020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q

Release Vehicle	Release Date	BIOS	BMC	10/25 GbE NIC SMCI Mellanox	Cache NVDIMM RMS200	Cache NVDIMM RMS300	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
Storage Firmware Bundle 1.2.17 through NetApp Element 12.0	03/20/2020	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.8.2	02/22/2022	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.8.1	06/02/2021	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.8	03/11/2020	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.7	11/21/2019	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.5.1	02/19/2020	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.5	09/26/2019	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.3.2	02/19/2020	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.3.1	08/19/2019	NA2.1	3.25	14.21.1000	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.1.1	02/19/2020	NA2.1	3.25	14.17.2020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.1	04/25/2019	NA2.1	3.25	14.17.2020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q

Release Vehicle	Release Date	BIOS	BMC	10/25 GbE NIC SMCI Mellanox	Cache NVDIMM RMS200	Cache NVDIMM RMS300	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
NetApp Element 11.0.2	02/19/2020	NA2.1	3.25	14.17.2020	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
NetApp Element 11.0	11/29/2018	NA2.1	3.25	14.17.2020	ae3b8cc	-	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q

Component firmware not managed by a Storage Firmware Bundle

The following firmware is not managed by a Storage Firmware Bundle:

Component	Current version
CPLD	01.A1.06
SAS Adapter	16.00.01.00
Microcontroller Unit (MCU)	1.18
SIOM 1/10 GbE NIC	1.93
Power Supply	1.3
Boot Device SSDSCKJB240G7	N2010121
Boot Device MTFDDAV240TCB1AR	DOMU037

SF38410, SF19210, SF9605, and SF4805

Full Model Numbers: SF38410, SF19210, SF9605, and SF4805

Component firmware managed by a Storage Firmware Bundle

During 11.x timeframe, NetApp Element software was the only way to release firmware. Starting with Element 12.0, the concept of a **Storage Firmware Bundle** was introduced and firmware updates were now possible by an independently released Storage Firmware Bundle or Storage Firmware Bundle included as part of an Element 12.x release.

ⓘ A dash (**-**) in the following table indicates that the particular hardware component was NOT supported in that given release vehicle.

Release Vehicle	Release Date	NIC	Cache NVDIMM RMS200 (RMS200)	Cache NVDIMM RMS200 (RMS300)	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
Storage Firmware Bundle 2.164.0	10/20/2022	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.164.0 through NetApp Element 12.7	10/20/2022	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.150.4	06/08/2022	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.150.4 through NetApp Element 12.5	06/08/2022	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.146.2	02/22/2022	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7A04Q
Storage Firmware Bundle 2.99.4 through NetApp Element 12.3.2	09/16/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 2.99.4 through NetApp Element 12.3.1.165	12/06/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q

Release Vehicle	Release Date	NIC	Cache NVDIMM RMS200 (RMS200)	Cache NVDIMM RMS200 (RMS300)	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
Storage Firmware Bundle 2.99.2	08/03/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 2.99.1 through NetApp Element 12.3.1.103	09/16/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 2.99 through NetApp Element 12.3	04/15/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 2.76.8	02/03/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 2.27.1	09/29/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
Storage Firmware Bundle 2.76.8 through NetApp Element 12.2.1	06/02/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 2.21 through NetApp Element 12.2	09/29/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q

Release Vehicle	Release Date	NIC	Cache NVDIMM RMS200 (RMS200)	Cache NVDIMM RMS200 (RMS300)	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
Storage Firmware Bundle 2.76.8 through NetApp Element 12.0.1	06/02/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7904Q
Storage Firmware Bundle 1.2.17 through NetApp Element 12.0	03/20/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.8.2	02/22/2022	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.8.1	06/02/2021	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.8	03/11/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.7	11/21/2019	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.5.1	02/19/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.5	09/26/2019	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.3.2	02/19/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.3.1	08/19/2019	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q

Release Vehicle	Release Date	NIC	Cache NVDIMM RMS200 (RMS200)	Cache NVDIMM RMS200 (RMS300)	Drive Samsung PM863 (SED)	Drive Samsung PM863 (N-SED)	Drive Toshiba Hawk-4 (SED)	Drive Toshiba Hawk-4 (N-SED)	Drive Samsung PM883 (SED)
NetApp Element 11.1.1	02/19/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.1	04/25/2019	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11.0.2	02/19/2020	7.10.18	ae3b8cc	7d8422bc	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q
NetApp Element 11	11/29/2018	7.10.18	ae3b8cc	-	GXT5404Q	GXT5103Q	8ENP7101	8ENP6101	HXT7104Q

Component firmware not managed by a Storage Firmware Bundle

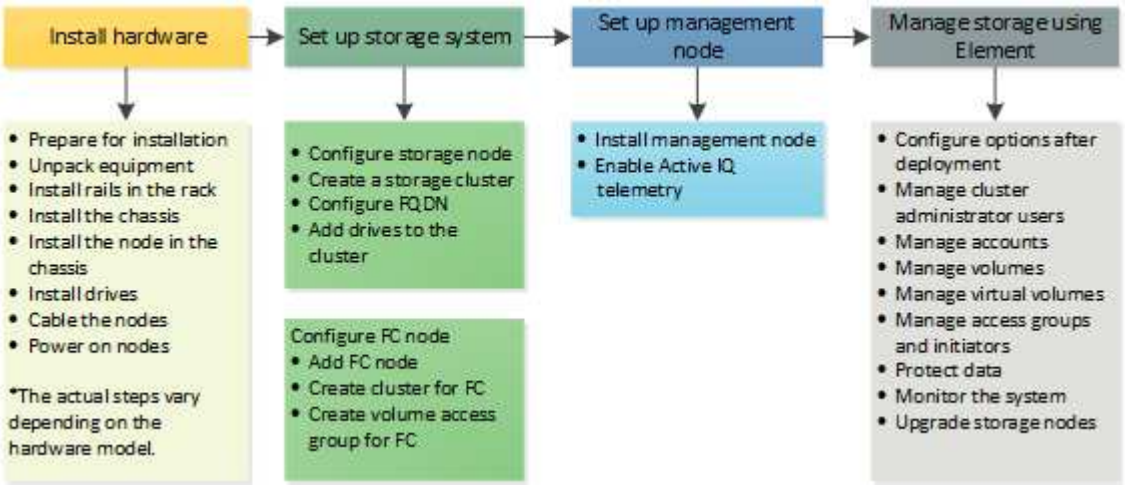
The following firmware is not managed by a Storage Firmware Bundle:

Component	Current version
BIOS	2.8.0
iDRAC	2.75.75.75
Identity Module	N41WC 1.02
SAS Adapter	16.00.01.00
Power Supply	1.3
Boot Device	M161225i

Setup overview

At this point, you should have installed the hardware. The hardware also includes Element software.

Next, you'll need to set up the storage system for your environment. You can set up a cluster with storage nodes or Fibre Channel nodes and manage it using Element software after you install and cable nodes in a rack unit and power them on.



Steps to set up storage

1. Select one of the following:
    - Set up cluster with storage nodes
    - Set up cluster with Fibre Channel nodes
2. Determine which SolidFire components to install
3. Set up a management node and enable Active IQ telemetry

Find more information

- Discover next steps for using storage
- SolidFire and Element Software Documentation

Setting up a cluster with Element storage nodes

You can set up a cluster with storage nodes and manage it using Element software after you install and cable nodes in a rack unit and power them on. You can then install and configure additional components in your storage system.

Steps

1. Configure a storage node
2. Create a storage cluster
3. Log in to the Element software user interface
4. Add drives to the cluster

5. Determine which SolidFire components to install
6. Set up a management node

Find more information

- SolidFire and Element Software Documentation

Configure a storage node

You must configure individual nodes before you can add them to a cluster. After you install and cable a node in a rack unit and power it on, you can configure the node network settings using the per-node UI or the node terminal user interface (TUI). Ensure that you have the necessary network configuration information for the node before proceeding.

There are two options for configuring storage nodes:

- Per-node UI: Use the per-node UI (https://<node_management_IP>:442) to configure node network settings.
- TUI: Use the node terminal user interface (TUI) to configure the node.

You cannot add a node with DHCP-assigned IP addresses to a cluster. You can use the DHCP IP address to initially configure the node in the per-node UI, TUI, or API. During this initial configuration, you can add static IP address information so that you can add the node to a cluster.

After initial configuration, you can access the node using the node's management IP address. You can then change the node settings, add it to a cluster, or use the node to create a cluster. You can also configure a new node using Element software API methods.

Beginning in Element version 11.0, nodes can be configured with IPv4, IPv6, or both addresses for their management network. This applies to both storage nodes and management nodes, except for management node 11.3 and later which does not support IPv6. When you create a cluster, only a single IPv4 or IPv6 address can be used for the MVIP and the corresponding address type must be configured on all nodes.

Configure a storage node using the per-node UI

You can configure nodes using the per-node user interface.

About this task

- You can configure the node to have either an IPv4 or IPv6 address.
- You need the DHCP address displayed in the TUI to access a node. You cannot use DHCP addresses to add a node to a cluster.

You should configure the management (Bond1G) and storage (Bond10G) interfaces for separate subnets. Bond1G and Bond10G interfaces configured for the same subnet cause routing problems when storage traffic is sent via the Bond1G interface. If you must use the same subnet for management and storage traffic, manually configure management traffic to use the Bond10G interface. You can do this for each node using the **Cluster Settings** page of the per-node UI.

Steps

1. In a browser window, enter the DHCP IP address of a node.

   You must add the extension `:442` to access the node; for example, https://172.25.103.6:442.

   The **Network Settings** tab opens with the **Bond1G** section.

2. Enter the 1G management network settings.
3. Click **Apply Changes**.
4. Click **Bond10G** to display the 10G storage network settings.
5. Enter the 10G storage network settings.
6. Click **Apply Changes**.
7. Click **Cluster Settings**.
8. Enter the hostname for the 10G network.
9. Enter the cluster name.

   This name must be added to the configuration for all nodes before a cluster can be created. All the nodes in a cluster must have identical cluster names. Cluster names are case-sensitive.

10. Click **Apply Changes**.

Configure a storage node using the TUI

You can use the terminal user interface (TUI) to perform initial configuration for new nodes.

You should configure the Bond1G (Management) and Bond10G (Storage) interfaces for separate subnets. Bond1G and Bond10G interfaces configured for the same subnet causes routing problems when storage traffic is sent via the Bond1G interface. If you must use the same subnet for management and storage traffic, manually configure management traffic to use the Bond10G interface. You can do this for each node using the **Cluster** > **Nodes** page of the Element UI.

Steps

1. Attach a keyboard and monitor to the node and then power on the node.

   The NetApp Storage Main menu of the TUI appears on the tty1 terminal.

   If the node cannot reach your configuration server, the TUI displays an error message. Check your configuration server connection or the networking connection to resolve the error.

2. Select **Network** > **Network Config**.

   To navigate through the menu, press the Up or Down arrow keys. To move to another button or to the fields from the buttons, press **Tab**. To navigate between fields, use the Up or Down arrow keys.

3. Select **Bond1G (Management)** or **Bond10G (Storage)** to configure the 1G and 10G network settings for the node.

4. For the Bond mode and Status fields, press **Tab** to select the Help button and identify the available options.

   All the nodes in a cluster must have identical cluster names. Cluster names are case-sensitive. If a DHCP server is running on the network with available IP addresses, the 1GbE address appears in the Address field.

5. Press **Tab** to select the **OK** button and save the changes.

   The node is put in a pending state and can be added to an existing cluster or a new cluster.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Create a storage cluster

You can create a storage cluster after you have configured all of the individual nodes. When you create a cluster, a cluster administrator user account is automatically created for you. The cluster administrator has permission to manage all cluster attributes and can create other cluster administrator accounts.

What you'll need

- You have installed the management node.
- You have configured all of the individual nodes.

About this task

During new node configuration, 1G or 10G Management IP (MIP) addresses are assigned to each node. You must use one of the node IP addresses created during configuration to open the Create a New Cluster page. The IP address you use depends on the network you have chosen for cluster management.

If you want to enable cluster-wide software encryption at rest for SolidFire all-flash storage clusters, you must do so during cluster creation. Beginning with Element 12.5, you must enable software encryption at rest in the Create Cluster UI during cluster creation. For Element 12.3.x and earlier, you must create the cluster using the CreateCluster API method and change the enableSoftwareEncryptionAtRest parameter to `true`. After software encryption at rest is enabled on the cluster, it cannot be disabled. You can enable and disable Hardware-based encryption at rest after cluster creation.

When creating a new cluster, consider the following:

- If you are using storage nodes that reside in a shared chassis, you might want to consider designing for chassis-level failure protection using the protection domains feature.
- If a shared chassis in not in use, you can define a custom protection domain layout.

Steps

1. In a browser window, enter `https://MIP:443`, where MIP is the management node IP address.
2. In Create a New Cluster, enter the following information:
   - Management VIP: Routable virtual IP on the 1GbE or 10GbE network for network management tasks.

> You can create a new cluster using IPv4 or IPv6 addressing.

- iSCSI (storage) VIP: Virtual IP on the 10GbE network for storage and iSCSI discovery.

> You cannot change the MVIP, SVIP, or cluster name after you create the cluster.

- User name: The primary cluster administrator user name for authenticated access to the cluster. You must save the user name for future reference.

> You can use uppercase and lowercase letters, special characters, and numbers for the user name and password.

- Password: Password for authenticated access to the cluster. You must save the password for future reference.
  Two-way data protection is enabled by default. You cannot change this setting.

3. Read the End User License Agreement, and select **I Agree**.
4. **Optional**: In the Nodes list, ensure that the check boxes for nodes that should not be included in the cluster are not selected.
5. Select **Create Cluster**.

   The system might take several minutes to create the cluster depending on the number of nodes in the cluster. On a properly configured network, a small cluster of five nodes should take less than one minute. After the cluster is created, the Create a New Cluster window is redirected to the MVIP URL address for the cluster and displays the Element UI.

For more information

- Managing storage with the Element API
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Access the Element software user interface

You can access the Element UI by using the management virtual IP (MVIP) address of the primary cluster node.

You must ensure that popup blockers and NoScript settings are disabled in your browser.

You can access the UI using IPv4 or IPv6 addressing, depending on configuration during cluster creation.

Steps

1. Choose one of the following:
   - IPv6: Enter `https://[IPv6_MVIP_address]`. For example:

     ```
     https://[fd20:8b1e:b256:45a::1234]/
     ```

   - IPv4: Enter `https://[IPv4 MVIP address]`. For example:

```
https://10.123.456.789/
```

2. For DNS, enter the host name.
3. Click through any authentication certificate messages.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Add drives to a cluster

When you add a node to the cluster or install new drives in an existing node, the drives automatically register as available. You must add the drives to the cluster by using either the Element UI or API before they can participate in the cluster.

Drives are not displayed in the Available Drives list when the following conditions exist:

- Drives are in Active, Removing, Erasing, or Failed state.
- The node of which the drive is a part of is in Pending state.

Steps

1. From the Element user interface, select **Cluster** > **Drives**.
2. Click **Available** to view the list of available drives.
3. Do one of the following:
   - To add individual drives, click the **Actions** icon for the drive you want to add and click **Add**.
   - To add multiple drives, select the check boxes of the drives to add, click **Bulk Actions**, and click **Add**.

```
== Find more information
* https://docs.netapp.com/us-en/element-software/index.html[SolidFire
and Element Software Documentation]
* https://docs.netapp.com/us-en/vcp/index.html[NetApp Element Plug-in
for vCenter Server^]
```

Set up a cluster with Fibre Channel nodes

Configure a Fibre Channel node

Fibre Channel nodes enable you to connect the cluster to a Fibre Channel network fabric. Fibre Channel nodes are added in pairs, and operate in active-active mode (all nodes actively process traffic for the cluster). Clusters running Element software version 9.0 and later support up to four nodes; clusters running previous versions support a maximum of two nodes.

You must ensure that the following conditions are met before you configure a Fibre Channel node:

- At least two Fibre Channel nodes are connected to Fibre Channel switches.
- All SolidFire Fibre Channel ports should be connected to your Fibre Channel fabric. The four SolidFire Bond10G network connections should be connected in one LACP bond group at the switch level. This will enable the best overall performance from the Fibre Channel systems.
- Review and validate all best practices for Fibre Channel clusters included in this NetApp Knowledge Base article.

  SolidFire FC cluster best practice

Network and cluster configuration steps are the same for Fibre Channel nodes and storage nodes.

When you create a new cluster with Fibre Channel nodes and SolidFire storage nodes, the worldwide port name (WWPN) addresses for the nodes are available in the Element UI. You can use the WWPN addresses to zone the Fibre Channel switch.

WWPNs are registered in the system when you create a new cluster with nodes. In the Element UI, you can find the WWPN addresses from the WWPN column of the FC Ports tab, which you access from the Cluster tab.

Find more information

Add Fibre Channel nodes to a cluster

Create a new cluster with Fibre Channel nodes

Create a new cluster with Fibre Channel nodes

You can create a new cluster after you have configured the individual Fibre Channel nodes. When you create a cluster, a cluster administrator user account is automatically created for you. The cluster administrator has permission to manage all cluster attributes and can create other cluster administrator accounts.

During new node configuration, 1G or 10G Management IP (MIP) addresses are assigned to each node. You must use one of the node IP addresses created during configuration to open the Create a New Cluster page. The IP address you use depends on the network you have chosen for cluster management.

What you'll need

You have configured the individual Fibre Channel nodes.

Steps

1. In a browser window, enter a node MIP address.
2. In Create a New Cluster, enter the following information:
   - Management VIP: Routable virtual IP on the 1GbE or 10GbE network for network management tasks.
   - iSCSI (storage) VIP: Virtual IP on the 10GbE network for storage and iSCSI discovery.

     You cannot change the SVIP after you create the cluster.

   - User name: The primary Cluster Admin user name for authenticated access to the cluster. You must save the user name for future reference.

> You can use uppercase and lowercase letters, special characters, and numbers for the user name.

- Password: Password for authenticated access to the cluster. You must save the user name for future reference.
  Two-way data protection is enabled by default. You cannot change this setting.

3. Read the End User License Agreement, and click **I Agree**.
4. **Optional**: In the Nodes list, ensure that the check boxes for nodes that should not be included in the cluster are not selected.
5. Click **Create Cluster**.

   The system might take several minutes to create the cluster depending on the number of nodes in the cluster. On a properly configured network, a small cluster of five nodes should take less than one minute. After the cluster is created, the Create a New Cluster window is redirected to the MVIP URL address for the cluster and displays the web UI.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Add Fibre Channel nodes to a cluster

You can add Fibre Channel nodes to a cluster when more storage is needed or during cluster creation. Fibre Channel nodes require initial configuration when they are first powered on. After the node is configured, it appears in the list of pending nodes and you can add it to a cluster.

The software version on each Fibre Channel node in a cluster must be compatible. When you add a Fibre Channel node to a cluster, the cluster installs the cluster version of Element on the new node as needed.

Steps

1. Select **Cluster** > **Nodes**.
2. Click **Pending** to view the list of pending nodes.
3. Do one of the following:
   - To add individual nodes, click the **Actions** icon for the node you want to add.
   - To add multiple nodes, select the check box of the nodes to add, and then **Bulk Actions**.

   > If the node you are adding has a different version of Element than the version running on the cluster, the cluster asynchronously updates the node to the version of Element running on the cluster master. After the node is updated, it automatically adds itself to the cluster. During this asynchronous process, the node will be in a pendingActive state.

4. Click **Add**.

   The node appears in the list of active nodes.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Set up zones for Fibre Channel nodes

When you create a new cluster with Fibre Channel nodes and SolidFire storage nodes, the worldwide port name (WWPN) addresses for the nodes are available in the web UI. You can use the WWPN addresses to zone the Fibre Channel switch.

WWPNs are registered in the system when you create a new cluster with nodes. In the Element UI, you can find the WWPN addresses from the WWPN column of the FC Ports tab, which you access from the Cluster tab.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Create a volume access group for Fibre Channel clients

Volume access groups enable communication between Fibre Channel clients and volumes on a SolidFire storage system. Mapping Fibre Channel client initiators (WWPN) to the volumes in a volume access group enables secure data I/O between a Fibre Channel network and a SolidFire volume.

You can also add iSCSI initiators to a volume access group; this gives the initiators access to the same volumes in the volume access group.

Steps

1. Click **Management** > **Access Groups**.
2. Click **Create Access Group**.
3. Enter a name for the volume access group in the **Name** field.
4. Select and add the Fibre Channel initiators from the **Unbound Fibre Channel Initiators** list.

   You can add or delete initiators at a later time.

5. **Optional:** Select and add an iSCSI initiator from the **Initiators** list.
6. To attach volumes to the access group, perform the following steps:
   a. Select a volume from the **Volumes** list.
   b. Click **Attach Volume**.
7. Click **Create Access Group**.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Determine which SolidFire components to install

You might want to check which SolidFire components, such as the management node, Active IQ and the NetApp Monitoring Agent (NMA), that you should install, depending on configuration and deployment choices.

The following table lists the additional components and indicates whether you should install them.

Component	Standalone SolidFire storage cluster	NetApp HCI cluster
Management node	Recommended	Installed by default, required
Active IQ	Recommended*	Recommended*
NetApp Monitoring Agent	Not supported	Recommended

*Active IQ is required for capacity-licensed SolidFire storage clusters.

Steps

1. Determine which components should be installed.
2. Complete the installation according to the install the management node procedure.

> To set up Active IQ, use the `--telemetry_active` parameter in the setup script to enable data collection for analytics by Active IQ.

3. For NetApp Monitoring Agent information, see this procedure.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Set up a management node

You can install the NetApp Element software management node (mNode) to upgrade and provide system services, manage cluster assets and settings, run system tests and utilities, and enable NetApp Support access for troubleshooting.

1. See the install the management node documentation.

> To set up Active IQ, use the `--telemetry_active` parameter in the setup script to enable data collection for analytics by Active IQ.

Find more information

- SolidFire and Element Software Documentation

- NetApp Element Plug-in for vCenter Server

Configure Fully Qualified Domain Name web UI access

SolidFire all-flash storage with NetApp Element software 12.2 or later enables you to access storage cluster web interfaces using the Fully Qualified Domain Name (FQDN). If you want to use the FQDN to access web user interfaces such as the Element web UI, per-node UI, or management node UI, you must first add a storage cluster setting to identify the FQDN used by the cluster.

This process enables the cluster to properly redirect a login session and improves integration with external services such as key managers and identity providers for multi-factor authentication.

What you'll need

- This feature requires Element 12.2 or later.
- Configuring this feature using NetApp Hybrid Cloud Control REST APIs requires management services 2.15 or later.
- Configuring this feature using the NetApp Hybrid Cloud Control UI requires management services 2.19 or later.
- To use REST APIs, you must have deployed a management node running version 11.5 or later.
- You need fully qualified domain names for the management node and each storage cluster that resolve correctly to the management node IP address and each storage cluster IP address.

You can configure or remove FQDN web UI access using NetApp Hybrid Cloud Control and the REST API. You can also troubleshoot incorrectly configured FQDNs.

- Configure FQDN web UI access using NetApp Hybrid Cloud Control
- Configure FQDN web UI access using the REST API
- Remove FQDN web UI access using NetApp Hybrid Cloud Control
- Remove FQDN web UI access using the REST API
- Troubleshooting

Configure FQDN web UI access using NetApp Hybrid Cloud Control

Steps

1. Open the IP address of the management node in a web browser:

   ```
   https://<ManagementNodeIP>
   ```

2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
3. Select the menu icon at the top right of the page.
4. Select **Configure**.
5. In the **Fully Qualified Domain Names** pane, select **Set Up**.
6. In the resulting window, enter the FQDNs for the management node and each storage cluster.

7. Select **Save**.

   The **Fully Qualified Domain Names** pane lists each storage cluster with its associated MVIP and FQDN.

   > Only connected storage clusters with the FQDN set are listed in the **Fully Qualified Domain Names** pane.

Configure FQDN web UI access using the REST API

Steps

1. Ensure that the Element storage nodes and the mNode have DNS configured correctly for the network environment so that FQDNs in the environment can be resolved. To set DNS, go to the per-node UI for storage nodes and to the management node, then select **Network Settings** > **Management Network**.
   a. Per-node UI for storage nodes: `https://<storage_node_management_IP>:442`
   b. Per-node UI for the management node: `https://<management_node_IP>:442`
2. Change the storage cluster settings using the Element API.
   a. Access the Element API and create the following cluster interface preference using the CreateClusterInterfacePreference API method, inserting the cluster MVIP FQDN for the preference value:
      - Name: `mvip_fqdn`
      - Value: `Fully Qualified Domain Name for the Cluster MVIP`

        In this example, FQDN=storagecluster.my.org:

        ```
        https://<Cluster_MVIP>/json-rpc/12.2?
        method=CreateClusterInterfacePreference&name=mvip_fqdn&value=stora
        gecluster.my.org
        ```
3. Change the management node settings using the REST API on the management node:
   a. Access the REST API UI for the management node by entering the management node IP address followed by `/mnode/2/`

      For example:

      `https://<management_node_IP>/mnode/2/`
   b. Click **Authorize** or any lock icon and enter the cluster user name and password.
   c. Enter the client ID as `mnode-client`.
   d. Click **Authorize** to begin the session and then close the window.
   e. From the server list, select `mnode2`.
   f. Click **GET /settings**.
   g. Click **Try it out**.
   h. Click **Execute**.

i. Record any proxy settings reported in the response body.

j. Click **PUT/setttings**.

k. Click **Try it out**.

l. In the request body area, enter the management node FQDN as the value for the `mnode_fqdn` parameter.

m. Enter any proxy setting values you recorded earlier in the remaining parameters in the request body. If you leave the proxy parameters empty or do not include them in the request body, existing proxy settings will be removed.

n. Click **Execute**.

Remove FQDN web UI access using NetApp Hybrid Cloud Control

You can use this procedure to remove FQDN web access for the management node and the storage clusters.

Steps

1. In the **Fully Qualified Domain Names** pane, select **Edit**.
2. In the resulting window, delete the contents in the **FQDN** text field.
3. Select **Save**.

   The window closes and the FQDN is no longer listed in the **Fully Qualified Domain Names** pane.

Remove FQDN web UI access using the REST API

Steps

1. Change the storage cluster settings using the Element API.
   a. Access the Element API and delete the following cluster interface preference using the `DeleteClusterInterfacePreference` API method:
      - Name: `mvip_fqdn`

        For example:

        ```
        https://<Cluster_MVIP>/json-
        rpc/12.2?method=DeleteClusterInterfacePreference&name=mvip_fqdn
        ```

2. Change the management node settings using the REST API on the management node:
   a. Access the REST API UI for the management node by entering the management node IP address followed by `/mnode/2/`. For example:

      ```
      https://<management_node_IP>/mnode/2/
      ```

   b. Select **Authorize** or any lock icon and enter the Element cluster user name and password.

   c. Enter the client ID as `mnode-client`.

   d. Select **Authorize** to begin a session.

e. Close the window.

f. Select **PUT /settings**.

g. Select **Try it out**.

h. In the request body area, do not enter a value for the `mnode_fqdn` parameter. Also specify whether the proxy should be used (`true` or `false`) for the `use_proxy` parameter.

```
{
 "mnode_fqdn": "",
 "use_proxy": false
}
```

i. Select **Execute**.

Troubleshooting

If FQDNs are configured incorrectly, you might have problems accessing either the management node, a storage cluster, or both. Use the following information to help troubleshoot the issue.

Issue	Cause	Resolution
• You get a browser error when attempting to access either the management node or the storage cluster using the FQDN. • You cannot log in to either the management node or the storage cluster using an IP address.	The management node FQDN and storage cluster FQDN are both incorrectly configured.	Use the REST API instructions on this page to remove the management node and storage cluster FQDN settings and configure them again.
• You get a browser error when attempting to access the storage cluster FQDN. • You cannot log in to either the management node or the storage cluster using an IP address.	The management node FQDN is correctly configured, but the storage cluster FQDN is incorrectly configured.	Use the REST API instructions on this page to remove the storage cluster FQDN settings and configure them again
• You get a browser error when attempting to access the management node FQDN. • You can log in to the management node and storage cluster using an IP address.	The management node FQDN is incorrectly configured, but the storage cluster FQDN is correctly configured.	Log in to NetApp Hybrid Cloud Control to correct the management node FQDN settings in the UI, or use the REST API instructions on this page to correct the settings.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

What's next

After you set up Element software, you manage storage by completing some of the following options:

- Access the Element software user interface
- Configure SolidFire system options after deployment
- Manage accounts
- Manage your system
- Manage volumes and virtual volumes
- Protect your data
- Troubleshoot your system

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products
- NetApp Element Plug-in for vCenter Server

Manage storage with Element software

Use Element software to set up SolidFire storage, monitor cluster capacity and performance, and manage storage activity across a multi-tenant infrastructure.

Element is the storage operating system at the heart of a SolidFire cluster. Element software runs independently on all nodes in the cluster and enables the nodes of the cluster to combine resources and present as a single storage system to external clients. Element software is responsible for all cluster coordination, scale and management of the system as a whole.

The software interface is built upon the Element API.

- Access the Element software user interface
- Configure SolidFire system options after deployment
- Upgrade storage system components
- Use basic options in the Element software UI
- Manage accounts
- Manage your system
- Manage volumes and virtual volumes
- Protect your data
- Troubleshoot your system

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Access the Element software user interface

You can access the Element UI by using the management virtual IP (MVIP) address of the primary cluster node.

You must ensure that popup blockers and NoScript settings are disabled in your browser.

You can access the UI using IPv4 or IPv6 addressing, depending on configuration during cluster creation.

1. Choose one of the following:
   - IPv6: Enter https://[IPv6 MVIP address] For example:

     ```
     https://[fd20:8b1e:b256:45a::1234]/
     ```

   - IPv4: Enter https://[IPv4 MVIP address] For example:

     ```
     https://10.123.456.789/
     ```

2. For DNS, enter the host name.
3. Click through any authentication certificate messages.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Configure SolidFire system options after deployment

After you set up your SolidFire system, you might want to perform some optional tasks.

If you change credentials in the system, you might want to know the impact on other components.

Additionally, you can configure settings for multi-factor authentication, external key management, and Federal Information Processing Standards (FIPS) security.
You should also look at updating passwords when needed.

Find more information

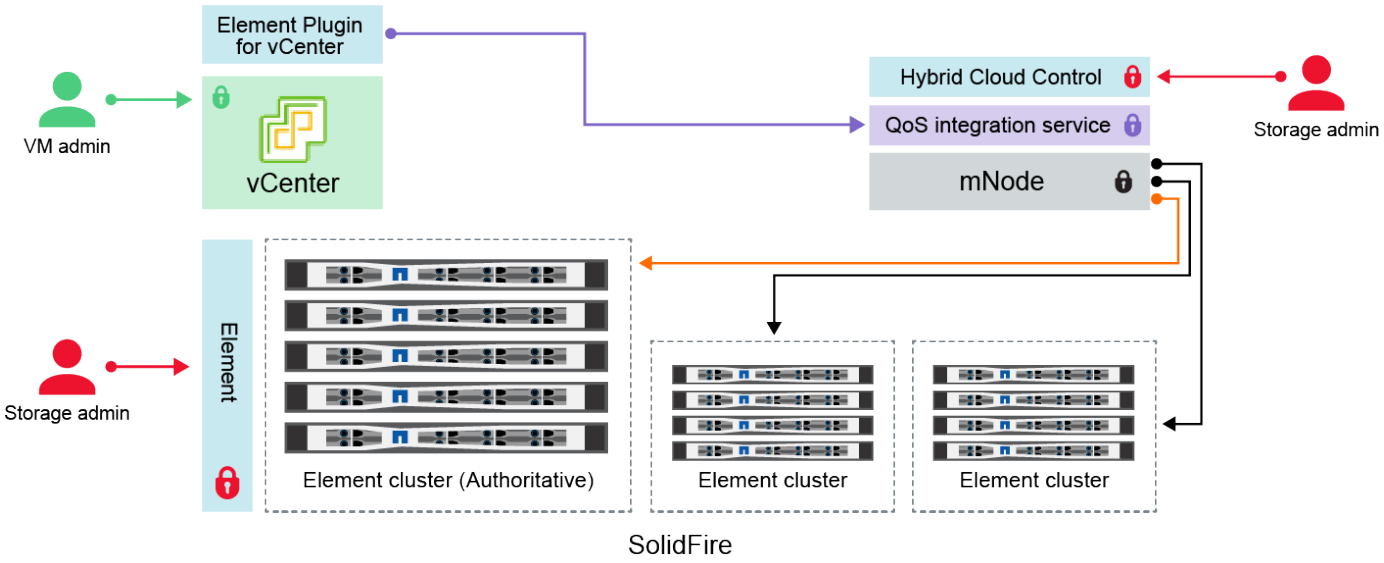
- Change credentials in NetApp HCI and NetApp SolidFire
- Change the Element software default SSL certificate
- Change the IPMI password for nodes
- Enable multi-factor authentication
- Get started with external key management
- Create a cluster supporting FIPS drives

Change credentials in NetApp HCI and NetApp SolidFire

Depending on the security policies in the organization that deployed NetApp HCI or NetApp SolidFire, changing credentials or passwords is commonly part of the security practices. Before you change passwords, you should be aware of the impact on other software components in the deployment.

If you change credentials for one component of a NetApp HCI or NetApp SolidFire deployment, the following table provides guidance as to the impact on other components.

NetApp SolidFire component interactions:



Credential Type and Icon	Usage by Admin	See these instructions
Element credentials 	**Applies to**: NetApp HCI and SolidFire Admins use these credentials to log into: • Element user interface on the Element storage cluster • Hybrid Cloud Control on the management node (mnode) When Hybrid Cloud Control manages multiple storage clusters, it accepts only the admin credentials for the storage clusters, known as the *authoritative cluster* that the mnode was initially set up for. For storage clusters later added to Hybrid Cloud Control, the mnode securely stores admin credentials. If credentials for subsequently added storage clusters are changed, the credentials must also be updated in the mnode using the mnode API.	• Update the storage cluster admin passwords. • Update the storage cluster admin credentials in the mnode using the modifyclusteradmin API.

Credential Type and Icon	Usage by Admin	See these instructions
vSphere Single Sign-on credentials 	**Applies to**: NetApp HCI only Admins use these credentials to log into the VMware vSphere Client. When vCenter is part of the NetApp HCI installation, credentials are configured in the NetApp Deployment Engine as the following: • username@vsphere.local with the specified password, and • administrator@vsphere.local with the specified password. When an existing vCenter is used to deploy NetApp HCI, the vSphere Single Sign-on credentials are managed by the IT VMware admins.	Update vCenter and ESXi credentials.
Baseboard management controller (BMC) credentials 	**Applies to**: NetApp HCI only Administrators use these credentials to log in to the BMC of the NetApp compute nodes in a NetApp HCI deployment. The BMC provides basic hardware monitoring and virtual console capabilities. BMC (sometimes referred to as *IPMI*) credentials for each NetApp compute node are stored securely on the mnode in NetApp HCI deployments. NetApp Hybrid Cloud Control uses BMC credentials in a service account capacity to communicate with the BMC in the compute nodes during compute node firmware upgrades. When the BMC credentials are changed, the credentials for the respective compute nodes must be updated also on the mnode to retain all Hybrid Cloud Control functionality.	• Configure IPMI for each node on NetApp HCI. • For H410C, H610C, and H615C nodes, change default IPMI password. • For H410S and H610S nodes, change default IPM password. • Change BMC credentials on the management node.
ESXi credentials 	**Applies to**: NetApp HCI only Admins can log into ESXi hosts using either SSH or the local DCUI with a local root account. In NetApp HCI deployments, the username is 'root' and the password was specified during the initial installation of that compute node in NetApp Deployment Engine. ESXi root credentials for each NetApp compute node are stored securely on the mnode in NetApp HCI deployments. NetApp Hybrid Cloud Control uses the credentials in a service account capacity to communicate with ESXi hosts directly during compute node firmware upgrades and health checks. When the ESXi root credentials are changed by a VMware admin, the credentials for the respective compute nodes must be updated on the mnode to retain Hybrid Cloud Control functionality.	Update credentials for vCenter and ESXi hosts.

Credential Type and Icon	Usage by Admin	See these instructions
QoS integration password 	**Applies to**: NetApp HCI and optional in SolidFire Not used for interactive logins by admins. The QoS integration between VMware vSphere and Element Software is enabled via: • Element Plug-in for vCenter Server, and • QoS service on the mnode. For authentication, the QoS service uses a password that is exclusively used in this context. The QoS password is specified during the initial installation of the Element Plug-in for vCenter Server, or auto-generated during NetApp HCI deployment. No impact on other components.	Update QoSSIOC credentials in the NetApp Element Plug-in for vCenter Server. The NetApp Element Plug-in for vCenter Server SIOC password is also known as the *QoSSIOC password*. Review the Element Plug-in for vCenter Server KB article.
vCenter Service Appliance credentials 	**Applies to**: NetApp HCI only if set up by NetApp Deployment Engine Admins can log into the vCenter Server appliance virtual machines. In NetApp HCI deployments, the username is 'root' and the password was specified during the initial installation of that compute node in the NetApp Deployment Engine. Depending on the VMware vSphere version deployed, certain admins in the vSphere Single Sign-on domain can also log in to the appliance. No impact on other components.	No changes needed.
NetApp Management Node admin credentials 	**Applies to**: NetApp HCI and optional in SolidFire Admins can log into the NetApp management node virtual machines for advanced configuration and troubleshooting. Depending on the management node version deployed, login via SSH is not enabled by default. In NetApp HCI deployments, the username and password was specified by the user during the initial installation of that compute node in NetApp Deployment Engine. No impact on other components.	No changes needed.

Find more information

- Change the Element software default SSL certificate
- Change the IPMI password for nodes
- Enable multi-factor authentication

- Get started with external key management
- Create a cluster supporting FIPS drives

Change the Element software default SSL certificate

You can change the default SSL certificate and private key of the storage node in the cluster using the NetApp Element API.

When a NetApp Element software cluster is created, the cluster creates a unique self-signed Secure Sockets Layer (SSL) certificate and private key that is used for all HTTPS communication via the Element UI, per-node UI, or APIs. Element software supports self-signed certificates as well as certificates that are issued and verified by a trusted Certificate Authority (CA).

You can use the following API methods to get more information about the default SSL certificate and make changes.

- **GetSSLCertificate**

  You can use the GetSSLCertificate method to retrieve information about the currently installed SSL certificate including all certificate details.

- **SetSSLCertificate**

  You can use the SetSSLCertificate method to set the cluster and per-node SSL certificates to the certificate and private key you supply. The system validates the certificate and private key to prevent an invalid certificate from being applied.

- **RemoveSSLCertificate**

  The RemoveSSLCertificate method removes the currently installed SSL certificate and private key. The cluster then generates a new self-signed certificate and private key.

> The cluster SSL certificate is automatically applied to all new nodes added to the cluster. Any node removed from the cluster reverts to a self-signed certificate and all user-defined certificate and key information is removed from the node.

Find more information

- Change the management node default SSL certificate
- What are the requirements around setting custom SSL certificates in Element Software?
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Change default IPMI password for nodes

You can change the default Intelligent Platform Management Interface (IPMI) administrator password as soon as you have remote IPMI access to the node. You might want to do this if there were any installation updates.

For details about configuring IPM access for nodes, see Configure IPMI for each node.

You can change the IPM password for these nodes:

- H410S nodes
- H610S nodes

Change the default IPMI password for H410S nodes

You should change the default password for the IPMI administrator account on each storage node as soon as you configure the IPMI network port.

What you'll need

You should have configured the IPMI IP address for each storage node.

Steps

1. Open a web browser on a computer that can reach the IPMI network and browse to the IPMI IP address for the node.
2. Enter the user name `ADMIN` and password `ADMIN` in the login prompt.
3. Upon logging in, click the **Configuration** tab.
4. Click **Users**.
5. Select the `ADMIN` user and click **Modify User**.
6. Select the **Change Password** check box.
7. Enter a new password in the **Password** and **Confirm Password** fields.
8. Click **Modify**, and then click **OK**.
9. Repeat this procedure for any other H410S nodes with default IPMI passwords.

Change the default IPMI password for H610S nodes

You should change the default password for the IPMI administrator account on each storage node as soon as you configure the IPMI network port.

What you'll need

You should have configured the IPMI IP address for each storage node.

Steps

1. Open a web browser on a computer that can reach the IPMI network and browse to the IPMI IP address for the node.
2. Enter the user name `root` and password `calvin` in the login prompt.
3. Upon logging in, click the menu navigation icon at the top left of the page to open the sidebar drawer.
4. Click **Settings**.
5. Click **User Management**.
6. Select the **Administrator** user from the list.
7. Enable the **Change Password** check box.
8. Enter a new, strong password in the **Password** and **Confirm Password** fields.
9. Click **Save** at the bottom of the page.
10. Repeat this procedure for any other H610S nodes with default IPMI passwords.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Use basic options in the Element software UI

The NetApp Element software web user interface (Element UI) enables you to monitor and perform common tasks on your SolidFire system.

Basic options include viewing API commands activated by UI activity and providing feedback.

- View API activity
- Icons in the Element interface
- Provide feedback

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

View API activity

The Element system uses the NetApp Element API as the foundation for its features and functionality. The Element UI enables you to view various types of real-time API activity on the system as you use the interface. With the API log, you can view user-initiated and background system API activity, as well as API calls made on the page you are currently viewing.

You can use the API log to identify what API methods are used for certain tasks, and see how to use the API methods and objects to build custom applications.

For information about each method, see Element Software API reference.

1. From the Element UI navigation bar, click **API Log**.
2. To modify the type of API activity displayed in the API Log window, perform the following steps:
   a. Select **Requests** to display API request traffic.
   b. Select **Responses** to display API response traffic.
   c. Filter the types of API traffic by selecting one of the following:
      - **User Initiated**: API traffic by your activities during this web UI session.
      - **Background Polling**: API traffic generated by background system activity.
      - **Current Page**: API traffic generated by tasks on the page you are currently viewing.

Find more information

- Managing storage with the Element API

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Interface refresh rate impacted by cluster load

Depending on API response times, the cluster might automatically adjust the data refresh interval for certain portions of the NetApp Element software page you are viewing.

The refresh interval is reset to the default when you reload the page in your browser. You can see the current refresh interval by clicking the cluster name in the upper-right of the page. Note that the interval controls how often API requests are made, not how quickly the data comes back from the server.

When a cluster is under heavy load, it might queue API requests from the Element UI. In rare circumstances, when system response is significantly delayed, such as a slow network connection combined with a busy cluster, you might be logged out of the Element UI if the system does not respond to queued API requests quickly enough. If you are redirected to the logout screen, you can log in again after dismissing any initial browser authentication prompt. Upon returning to the overview page, you might be prompted for cluster credentials if they are not saved by your browser.

Icons in the Element interface

The NetApp Element software interface displays icons to represent actions you can take on system resources.

The following table provides a quick reference:

Icon	Description
	Actions
	Backup to
	Clone or copy
	Delete or purge
	Edit
	Filter
	Pair

	Refresh
	Restore
	Restore from
	Rollback
	Snapshot

Provide feedback

You can help improve the Element software web user interface and address any UI issues by using the feedback form that is accessible throughout the UI.

1. From any page in the Element UI, click the **Feedback** button.
2. Enter relevant information in the Summary and Description fields.
3. Attach any helpful screenshots.
4. Enter a name and email address.
5. Select the check box to include data about your current environment.
6. Click **Submit**.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Manage accounts

In SolidFire storage systems, tenants can use accounts to enable clients to connect to volumes on a cluster. When you create a volume, it is assigned to a specific account. You can also manage cluster administrator accounts for a SolidFire storage system.

- Work with accounts using CHAP
- Manage cluster administrator user accounts

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Work with accounts using CHAP

In SolidFire storage systems, tenants can use accounts to enable clients to connect to volumes on a cluster. An account contains the Challenge-Handshake Authentication Protocol (CHAP) authentication required to access the volumes assigned to it. When you create a volume, it is assigned to a specific account.

An account can have up to two-thousand volumes assigned to it, but a volume can belong to only one account.

CHAP algorithms

Beginning with Element 12.7, secure FIPS compliant CHAP algorithms SHA1, SHA-256, and SHA3-256 are supported. With Element 12.7, when a host iSCSI initiator is creating an iSCSI session with an Element iSCSI target, it requests a list of CHAP algorithms to use. The Element iSCSI target chooses the first algorithm that it supports from the list requested by the host iSCSI initiator. To confirm that the Element iSCSI target chooses the most secure algorithm, you must configure the host iSCSI initiator to send a list of algorithms ordered from most secure, for example, SHA3-256, to least secure, for example, SHA1 or MD5. When SHA algorithms are not requested by the host iSCSI initiator, the Element iSCSI target chooses MD5, assuming the proposed algorithm list from the host contains MD5. You might need to update the host iSCSI initiator configuration to enable support for the secure algorithms.

During an Element 12.7 upgrade, if you have already updated the host iSCSI initiator configuration to send a session request with a list that includes SHA algorithms, as the storage nodes reboot, the new secure algorithms are activated and new or reconnected iSCSI sessions are established using the most secure protocol. All existing iSCSI sessions transition from MD5 to SHA during the upgrade. If you do not update the host iSCSI initiator configuration to request SHA, the existing iSCSI sessions will continue to use MD5. At a later date, after you update the host iSCSI initiator CHAP algorithms, the iSCSI sessions should transition gradually from MD5 to SHA over time based on maintenance activities that result in iSCSI session reconnects.

For example, the default host iSCSI initiator in Red Hat Enterprise Linux (RHEL) 8.3 has the `node.session.auth.chap_algs = SHA3-256,SHA256,SHA1,MD5` setting commented out which results in the iSCSI initiator only using MD5. Uncommenting this setting on the host and restarting the iSCSI initiator triggers iSCSI sessions from that host to start using SHA3-256.

If required, you can use the ListISCSISessions API method to see the CHAP algorithms being used for each session.

Create an account

You can create an account to allow access to volumes.

Each account name in the system must be unique.

1. Select **Management** > **Accounts**.
2. Click **Create Account**.
3. Enter a **Username**.
4. In the **CHAP Settings** section, enter the following information:

   Leave the credential fields blank to auto-generate either password.

   - **Initiator Secret** for CHAP node session authentication.

- **Target Secret** for CHAP node session authentication.

5. Click **Create Account**.

View account details

You can view performance activity for individual accounts in a graphical format.

The graph information provides I/O and throughput information for the account. The Average and Peak activity levels are shown in increments of 10-second reporting periods. These statistics include activity for all volumes assigned to the account.

1. Select **Management** > **Accounts**.
2. Click the Actions icon for an account.
3. Click **View Details**.

Here are some of the details:

- **Status**: The status of the account. Possible values:
  - active: An active account.
  - locked: A locked account.
  - removed: An account that has been deleted and purged.
- **Active Volumes**: The number of active volumes assigned to the account.
- **Compression**: The compression efficiency score for the volumes assigned to the account.
- **Deduplication**: The deduplication efficiency score for the volumes assigned to the account.
- **Thin Provisioning**: The thin provisioning efficiency score for the volumes assigned to the account.
- **Overall Efficiency**: The overall efficiency score for the volumes assigned to the account.

Edit an account

You can edit an account to change the status, change the CHAP secrets, or modify the account name.

Modifying CHAP settings in an account or removing initiators or volumes from an access group can cause initiators to lose access to volumes unexpectedly. To verify that volume access will not be lost unexpectedly, always log out iSCSI sessions that will be affected by an account or access group change, and verify that initiators can reconnect to volumes after any changes to initiator settings and cluster settings have been completed.

> Persistent volumes that are associated with management services are assigned to a new account that is created during installation or upgrade. If you are using persistent volumes, do not modify or delete their associated account.

1. Select **Management** > **Accounts**.
2. Click the Actions icon for an account.
3. In the resulting menu, select **Edit**.
4. **Optional:** Edit the **Username**.
5. **Optional:** Click the **Status** drop-down list and select a different status.

Changing the status to **locked** terminates all iSCSI connections to the account, and the account is no longer accessible. Volumes associated with the account are maintained; however, the volumes are not iSCSI discoverable.

6. **Optional:** Under **CHAP Settings**, edit the **Initiator Secret** and **Target Secret** credentials used for node session authentication.

   If you do not change the **CHAP Settings** credentials, they remain the same. If you make the credentials fields blank, the system generates new passwords.

7. Click **Save Changes**.

Delete an account

You can delete an account when it is no longer needed.

Delete and purge any volumes associated with the account before you delete the account.

Persistent volumes that are associated with management services are assigned to a new account that is created during installation or upgrade. If you are using persistent volumes, do not modify or delete their associated account.

1. Select **Management** > **Accounts**.
2. Click the Actions icon for the account you want to delete.
3. In the resulting menu, select **Delete**.
4. Confirm the action.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Manage cluster administrator user accounts

You can manage cluster administrator accounts for a SolidFire storage system by creating, deleting, and editing cluster administrator accounts, changing the cluster administrator password, and configuring LDAP settings to manage system access for users.

Storage cluster administrator account types

There are two types of administrator accounts that can exist in a storage cluster running NetApp Element software: the primary cluster administrator account and a cluster administrator account.

- **Primary cluster administrator account**

  This administrator account is created when the cluster is created. This account is the primary administrative account with the highest level of access to the cluster. This account is analogous to a root user in a Linux system. You can change the password for this administrator account.

- **Cluster administrator account**

  You can give a cluster administrator account a limited range of administrative access to perform specific tasks within a cluster. The credentials assigned to each cluster administrator account are used to authenticate API and Element UI requests within the storage system.

  A local (non-LDAP) cluster administrator account is required to access active nodes in a cluster via the per-node UI. Account credentials are not required to access a node that is not yet part of a cluster.

View cluster admin details

1. To create a cluster-wide (non-LDAP) cluster administrator account, perform the following actions:
   a. Click **Users** > **Cluster Admins**.
2. On the Cluster Admins page of the Users tab, you can view the following information.
   - **ID**: Sequential number assigned to the cluster administrator account.
   - **Username**: The name given to the cluster administrator account when it was created.
   - **Access**: The user permissions assigned to the user account. Possible values:
     - read
     - reporting
     - nodes
     - drives
     - volumes
     - accounts
     - clusterAdmins
     - administrator
     - supportAdmin

     All permissions are available to the administrator access type.
   - **Type**: The type of cluster administrator. Possible values:
     - Cluster
     - Ldap
   - **Attributes**: If the cluster administrator account was created using the Element API, this column shows any name-value pairs that were set using that method.

     See NetApp Element Software API Reference.

Create a cluster administrator account

You can create new cluster administrator accounts with permissions to allow or restrict access to specific areas of the storage system. When you set cluster administrator account permissions, the system grants read-only rights for any permissions you do not assign to the cluster administrator.

If you want to create an LDAP cluster administrator account, ensure that LDAP is configured on the cluster

before you begin.

Enable LDAP authentication with the Element user interface

You can later change cluster administrator account privileges for reporting, nodes, drives, volumes, accounts, and cluster-level access. When you enable a permission, the system assigns write access for that level. The system grants the administrator user read-only access for the levels that you do not select.

You can also later remove any cluster administrator user account created by a system administrator. You cannot remove the primary cluster administrator account that was created when the cluster was created.

1. To create a cluster-wide (non-LDAP) cluster administrator account, perform the following actions:
   a. Click **Users** > **Cluster Admins**.
   b. Click **Create Cluster Admin**.
   c. Select the **Cluster** user type.
   d. Enter a user name and password for the account and confirm password.
   e. Select user permissions to apply to the account.
   f. Select the check box to agree to the End User License Agreement.
   g. Click **Create Cluster Admin**.
2. To create a cluster administrator account in the LDAP directory, perform the following actions:
   a. Click **Cluster** > **LDAP**.
   b. Ensure that LDAP Authentication is enabled.
   c. Click **Test User Authentication** and copy the distinguished name that appears for the user or one of the groups of which the user is a member so that you can paste it later.
   d. Click **Users** > **Cluster Admins**.
   e. Click **Create Cluster Admin**.
   f. Select the LDAP user type.
   g. In the Distinguished Name field, follow the example in the text box to enter a full distinguished name for the user or group. Alternatively, paste it from the distinguished name you copied earlier.

      If the distinguished name is part of a group, then any user that is a member of that group on the LDAP server will have permissions of this admin account.

      To add LDAP Cluster Admin users or groups the general format of the username is “LDAP:<Full Distinguished Name>”.

   h. Select user permissions to apply to the account.
   i. Select the check box to agree to the End User License Agreement.
   j. Click **Create Cluster Admin**.

Edit cluster administrator permissions

You can change cluster administrator account privileges for reporting, nodes, drives, volumes, accounts, and cluster-level access. When you enable a permission, the system assigns write access for that level. The system grants the administrator user read-only access for the levels that you do not select.

1. Click **Users** > **Cluster Admins**.

2. Click the Actions icon for the cluster administrator you want to edit.
3. Click **Edit**.
4. Select user permissions to apply to the account.
5. Click **Save Changes**.

Change passwords for cluster administrator accounts

You can use the Element UI to change cluster administrator passwords.

1. Click **Users** > **Cluster Admins**.
2. Click the Actions icon for the cluster administrator you want to edit.
3. Click **Edit**.
4. In the Change Password field, enter a new password and confirm it.
5. Click **Save Changes**.

Find more information

- Enable LDAP authentication with the Element user interface
- Disable LDAP
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Manage LDAP

You can set up the Lightweight Directory Access Protocol (LDAP) to enable secure, directory-based login functionality to SolidFire storage. You can configure LDAP at the cluster level and authorize LDAP users and groups.

Managing LDAP involves setting up LDAP authentication to a SolidFire cluster using an existing Microsoft Active Directory environment and testing the configuration.

You can use both IPv4 and IPv6 addresses.

Enabling LDAP involves the following high-level steps, described in detail:

1. **Complete pre-configuration steps for LDAP support**. Validate that you have all of the details required to configure LDAP authentication.
2. **Enable LDAP authentication**. Use either the Element UI or the Element API.
3. **Validate the LDAP configuration**. Optionally, check that the cluster is configured with the correct values by running the GetLdapConfiguration API method or by checking the LCAP configuration using the Element UI.
4. **Test the LDAP authentication** (with the `readonly` user). Test that the LDAP configuration is correct either by running the TestLdapAuthentication API method or by using the Element UI. For this initial test, use the username “sAMAccountName” of the `readonly` user. This will validate that your cluster is configured correctly for LDAP authentication and also validate that the `readonly` credentials and access are correct. If this step fails, repeat steps 1 through 3.

5. **Test the LDAP authentication** (with a user account that you want to add). Repeat setp 4 with a user account that you want to add as an Element cluster admin. Copy the `distinguished` name (DN) or the user (or the group). This DN will be used in step 6.
6. **Add the LDAP cluster admin** (copy and paste the DN from the Test LDAP authentication step). Using either the Element UI or the AddLdapClusterAdmin API method, create a new cluster admin user with the appropriate access level. For the username, paste in the full DN you copied in Step 5. This assures that the DN is formatted correctly.
7. **Test the cluster admin access**. Log in to the cluster using the newly created LDAP cluster admin user. If you added an LDAP group, you can log in as any user in that group.

Complete pre-configuration steps for LDAP support

Before you enable LDAP support in Element, you should set up a Windows Active Directory Server and perform other pre-configuration tasks.

Steps

1. Set up a Windows Active Directory Server.
2. **Optional:** Enable LDAPS support.
3. Create users and groups.
4. Create a read-only service account (such as “sfreadonly”) to be used for searching the LDAP directory.

Enable LDAP authentication with the Element user interface

You can configure storage system integration with an existing LDAP server. This enables LDAP administrators to centrally manage storage system access for users.

You can configure LDAP with either the Element user interface or the Element API. This procedure describes how to configure LDAP using the Element UI.

This example shows how to configure LDAP authentication on SolidFire and it uses `SearchAndBind` as the authentication type. The example uses a single Windows Server 2012 R2 Active Directory Server.

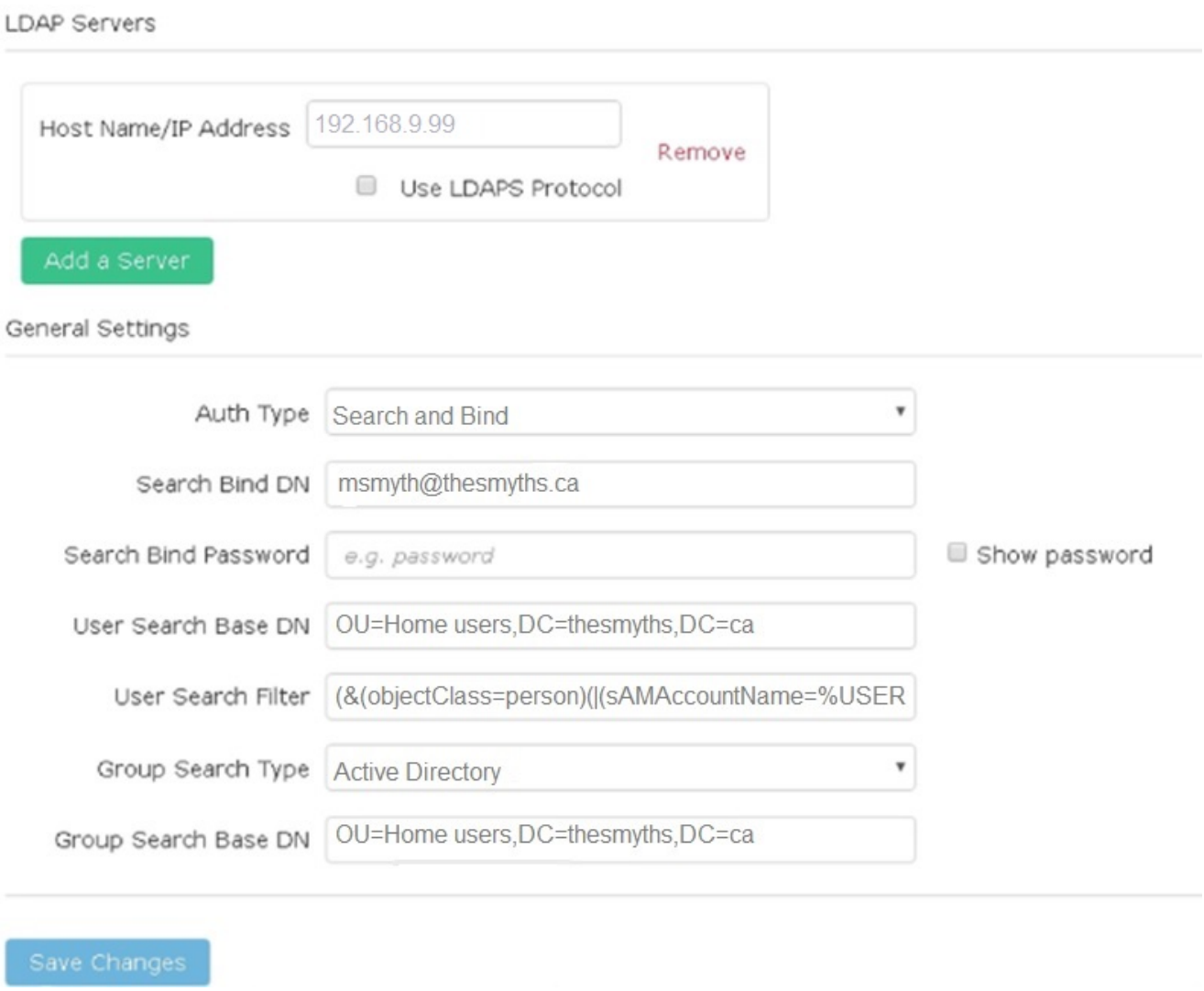
Steps

1. Click **Cluster** > **LDAP**.
2. Click **Yes** to enable LDAP authentication.
3. Click **Add a Server**.
4. Enter the **Host Name/IP Address**.

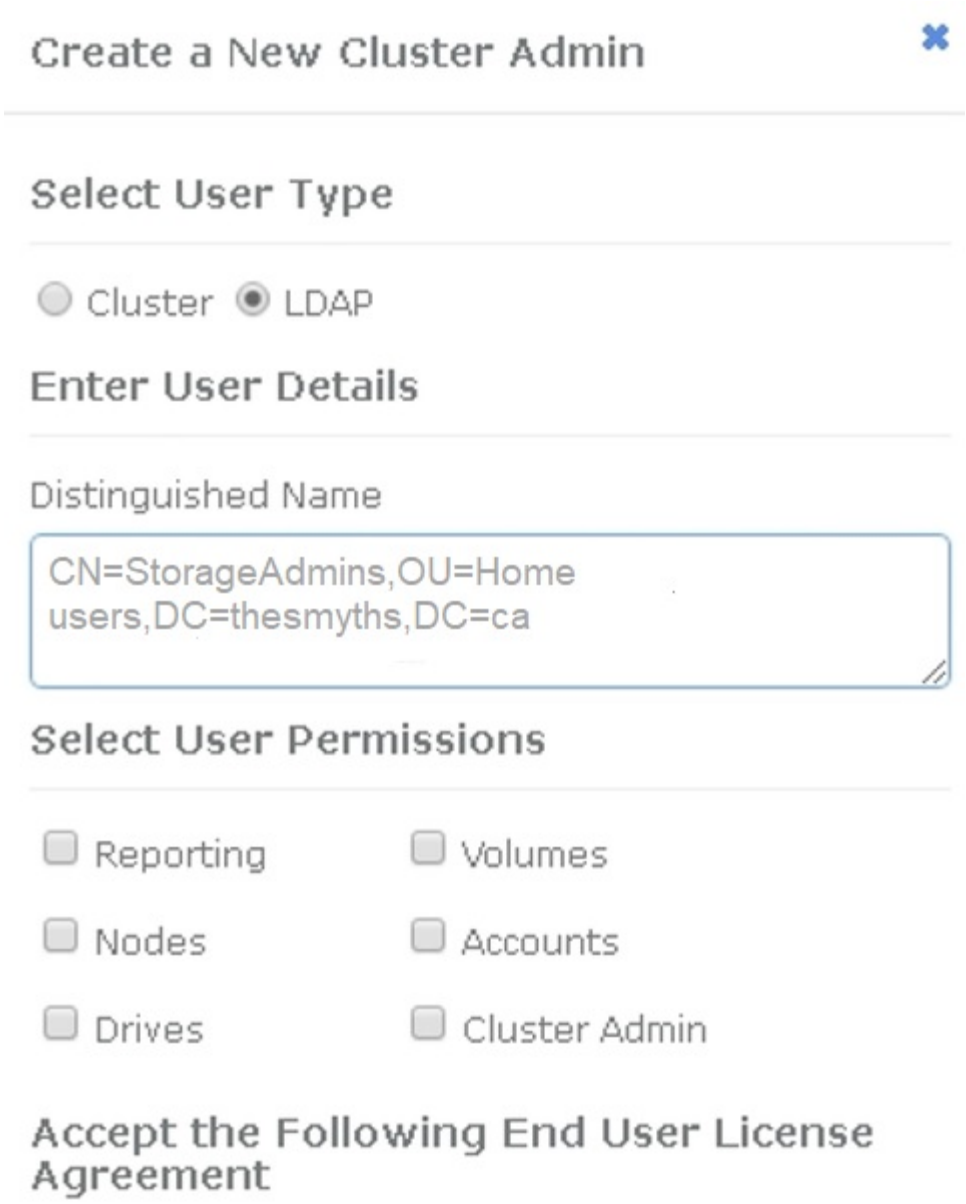
   An optional custom port number can also be entered.

   For example, to add a custom port number, enter <host name or ip address>:<port number>

5. **Optional:** Select **Use LDAPS Protocol**.
6. Enter the required information in **General Settings**.



7. Click **Enable LDAP**.
8. Click **Test User Authentication** if you want to test the server access for a user.
9. Copy the distinguished name and user group information that appears for use later when creating cluster administrators.
10. Click **Save Changes** to save any new settings.
11. To create a user in this group so that anyone can log in, complete the following:
    a. Click **User** > **View**.



b. For the new user, click **LDAP** for the User Type, and paste the group you copied to the Distinguished Name field.
c. Select the permissions, typically all permissions.
d. Scroll down to the End User License Agreement and click **I accept**.
e. Click **Create Cluster Admin**.

   Now you have a user with the value of an Active Directory group.

To test this, log out of the Element UI and log back in as a user in that group.

Enable LDAP authentication with the Element API

You can configure storage system integration with an existing LDAP server. This enables LDAP administrators to centrally manage storage system access for users.

You can configure LDAP with either the Element user interface or the Element API. This procedure describes how to configure LDAP using the Element API.

To leverage LDAP authentication on a SolidFire cluster, you enable LDAP authentication first on the cluster using the `EnableLdapAuthentication` API method.

Steps

1. Enable LDAP authentication first on the cluster using the `EnableLdapAuthentication` API method.
2. Enter the required information.

```
{
     "method":"EnableLdapAuthentication",
     "params":{
          "authType": "SearchAndBind",
          "groupSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
          "groupSearchType": "ActiveDirectory",
          "searchBindDN": "SFReadOnly@prodtest.solidfire.net",
          "searchBindPassword": "ReadOnlyPW",
          "userSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net ",
          "userSearchFilter":
"(&(objectClass=person)(sAMAccountName=%USERNAME%))"
          "serverURIs": [
               "ldap://172.27.1.189",
          [
     },
  "id":"1"
}
```

3. Change the values of the following parameters:

Parameters used	Description
authType: SearchAndBind	Dictates that the cluster will use the readonly service account to first search for the user being authenticated and subsequently bind that user if found and authenticated.
groupSearchBaseDN: dc=prodtest,dc=solidfire,dc=net	Specifies the location in the LDAP tree to begin searching for groups. For this example, we've used the root of our tree. If your LDAP tree is very large, you might want to set this to a more granular sub-tree to decrease search times.
userSearchBaseDN: dc=prodtest,dc=solidfire,dc=net	Specifies the location in the LDAP tree to begin searching for users. For this example, we've used the root of our tree. If your LDAP tree is very large, you might want to set this to a more granular sub-tree to decrease search times.

Parameters used	Description
groupSearchType: ActiveDirectory	Uses the Windows Active Directory server as the LDAP server.

`userSearchFilter: "(&(objectClass=person)(sAMAccountName=%USERNAME%))"`
 	
To use the userPrincipalName (email address for login) you could change the userSearchFilter to:
 	
`"(&(objectClass=person)(userPrincipalName=%USERNAME%))"`
 	
Or, to search both userPrincipalName and sAMAccountName, you can use the following userSearchFilter:
 	
`"(&(objectClass=person)(`
	(sAMAccountName=%USERNAME%)(userPrincipalName=%USERNAME%)))"
	---- 	

Leverages the sAMAccountName as our username for logging in to the SolidFire cluster. These settings tell LDAP to search for the username specified during login in the sAMAccountName attribute and also limit the search to entries that have “person” as a value in the objectClass attribute.	searchBindDN
This is the distinguished name of readonly user that will be used to search the LDAP directory. For active directory it’s usually easiest to use the userPrincipalName (email address format) for the user.	searchBindPassword

To test this, log out of the Element UI and log back in as a user in that group.

View LDAP details

View LDAP information on the LDAP page on the Cluster tab.

> You must enable LDAP to view these LDAP configuration settings.

1. To view LDAP details with the Element UI, click **Cluster** > **LDAP**.
   - **Host Name/IP Address**: Address of an LDAP or LDAPS directory server.

- **Auth Type**: The user authentication method. Possible values:
  - Direct Bind
  - Search And Bind
- **Search Bind DN**: A fully qualified DN to log in with to perform an LDAP search for the user (needs bind-level access to the LDAP directory).
- **Search Bind Password**: Password used to authenticate access to the LDAP server.
- **User Search Base DN**: The base DN of the tree used to start the user search. The system searches the subtree from the specified location.
- **User Search Filter**: Enter the following using your domain name:

```
(&(objectClass=person)(|(sAMAccountName=%USERNAME%)(userPrincipalName=%USERNAME%)))
```

- **Group Search Type**: Type of search that controls the default group search filter used. Possible values:
  - Active Directory: Nested membership of all of a user's LDAP groups.
  - No Groups: No group support.
  - Member DN: Member DN-style groups (single-level).
- **Group Search Base DN**: The base DN of the tree used to start the group search. The system searches the subtree from the specified location.
- **Test User Authentication**: After LDAP is configured, use this to test the user name and password authentication for the LDAP server. Enter an account that already exists to test this. The distinguished name and user group information appears, which you can copy for later use when creating cluster administrators.

Test the LDAP configuration

After configuring LDAP, you should test it by using either the Element UI or the Element API `TestLdapAuthentication` method.

Steps

1. To test the LDAP configuration with the Element UI, do the following:
   a. Click **Cluster** > **LDAP**.
   b. Click **Test LDAP Authentication**.
   c. Resolve any issues by using the information in the table below:

Error message	Description
`xLDAPUserNotFound`	• The user being tested was not found in the configured `userSearchBaseDN` subtree. • The `userSearchFilter` is configured incorrectly.

Error message	Description
`xLDAPBindFailed (Error: Invalid credentials)`	• The username being tested is a valid LDAP user, but the password provided is incorrect. • The username being tested is a valid LDAP user, but the account is currently disabled.
`xLDAPSearchBindFailed (Error: Can't contact LDAP server)`	The LDAP server URI is incorrect.
`xLDAPSearchBindFailed (Error: Invalid credentials)`	The read-only username or password is configured incorrectly.
`xLDAPSearchFailed (Error: No such object)`	The `userSearchBaseDN` is not a valid location within the LDAP tree.
`xLDAPSearchFailed (Error: Referral)`	• The `userSearchBaseDN` is not a valid location within the LDAP tree. • The `userSearchBaseDN` and `groupSearchBaseDN` are in a nested OU. This can cause permission issues. The workaround is to include the OU in the user and group base DN entries, (for example: `ou=storage, cn=company, cn=com`)

2. To test the LDAP configuration with the Element API, do the following:
   a. Call the TestLdapAuthentication method.

```
{
  "method":"TestLdapAuthentication",
     "params":{
        "username":"admin1",
        "password":"admin1PASS
      },
      "id": 1
}
```

   b. Review the results. If the API call is successful, the results include the specified user's distinguished name and a list of groups in which the user is a member.

```
{
"id": 1
     "result": {
         "groups": [

"CN=StorageMgmt,OU=PTUsers,DC=prodtest,DC=solidfire,DC=net"
         ],
         "userDN": "CN=Admin1
Jones,OU=PTUsers,DC=prodtest,DC=solidfire,DC=net"
     }
}
```

Disable LDAP

You can disable LDAP integration using the Element UI.

Before you begin, you should note all the configuration settings, because disabling LDAP erases all settings.

Steps

1. Click **Cluster** > **LDAP**.
2. Click **No**.
3. Click **Disable LDAP**.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Manage your system

You can manage your system in the Element UI. This includes enabling multi-factor authentication, managing cluster settings, supporting Federal Information Processing Standards (FIPS), and using external key management.

- Enable multi-factor authentication
- Configure cluster settings
- Create a cluster supporting FIPS drives
- Get started with external key management

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Enable multi-factor authentication

Multi-factor authentication (MFA) uses a third-party Identity Provider (IdP) via the Security Assertion Markup Language (SAML) to manage user sessions. MFA enables administrators to configure additional factors of authentication as required, such as password and text message, and password and email message.

Set up multi-factor authentication

You can use these basic steps via the Element API to set up your cluster to use multi-factor authentication.

Details of each API method can be found in the Element API Reference.

1. Create a new third-party Identity Provider (IdP) configuration for the cluster by calling the following API method and passing the IdP metadata in JSON format: `CreateIdpConfiguration`

   IdP metadata, in plain text format, is retrieved from the third-party IdP. This metadata needs to be validated to ensure that it is correctly formatted in JSON. There are numerous JSON formatter applications available that you can use, for example:https://freeformatter.com/json-escape.html.

2. Retrieve cluster metadata, via spMetadataUrl, to copy to the third-party IdP by calling the following API method: `ListIdpConfigurations`

   spMetadataUrl is a URL used to retrieve service provider metadata from the cluster for the IdP in order to establish a trust relationship.

3. Configure SAML assertions on the third-party IdP to include the “NameID” attribute to uniquely identify a user for audit logging and for Single Logout to function properly.
4. Create one or more cluster administrator user accounts authenticated by a third-party IdP for authorization by calling the following API method:`AddIdpClusterAdmin`

   > The username for the IdP cluster Administrator should match the SAML attribute Name/Value mapping for the desired effect, as shown in the following examples:

   - email=bob@company.com — where the IdP is configured to release an email address in the SAML attributes.
   - group=cluster-administrator - where the IdP is configured to release a group property in which all users should have access.
     Note that the SAML attribute Name/Value pairing is case-sensitive for security purposes.
5. Enable MFA for the cluster by calling the following API method: `EnableIdpAuthentication`

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Additional information for multi-factor authentication

You should be aware of the following caveats in relation to multi-factor authentication.

- In order to refresh IdP certificates that are no longer valid, you will need to use a non-IdP admin user to call the following API method: `UpdateIdpConfiguration`
- MFA is incompatible with certificates that are less than 2048 bits in length. By default, a 2048-bit SSL certificate is created on the cluster. You should avoid setting a smaller sized certificate when calling the API method: `SetSSLCertificate`

> If the cluster is using a certificate that is less than 2048 bits pre-upgrade, the cluster certificate must be updated with a 2048-bit or greater certificate after upgrade to Element 12.0 or later.

- IdP admin users cannot be used to make API calls directly (for example, via SDKs or Postman) or used for other integrations (for example, OpenStack Cinder or vCenter Plug-in). Add either LDAP cluster admin users or local cluster admin users if you need to create users that have these abilities.

Find more information

- Managing storage with the Element API
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Configure cluster settings

You can view and change cluster-wide settings and perform cluster-specific tasks from the Cluster tab of the Element UI.

You can configure settings such as cluster fullness threshold, support access, encryption at rest, virtual volumes, SnapMirror, and NTP broadcast client.

Options

- Work with virtual volumes
- Use SnapMirror replication between Element and ONTAP clusters
- Set the cluster full threshold
- Enable and disable support access
- How are the blockSpace thresholds calculated for Element
- Enable and disable encryption for a cluster
- Manage the Terms of Use banner
- Configure Network Time Protocol servers for the cluster to query
- Manage SNMP
- Manage drives
- Manage nodes
- Manage virtual networks
- View Fibre Channel ports details

Find more information

- SolidFire and Element Software Documentation

- NetApp Element Plug-in for vCenter Server

Enable and disable encryption at rest for a cluster

With SolidFire clusters, you can encrypt all at-rest data stored on cluster drives. You can enable cluster-wide protection of self-encrypting drives (SED) using either hardware or software-based encryption at rest.

You can enable hardware encryption at rest using the Element UI or API. Enabling the hardware encryption at rest feature does not affect performance or efficiency on the cluster. You can enable software encryption at rest using the Element API only.

Hardware-based encryption at rest is not enabled by default during cluster creation and can be enabled and disabled from the Element UI.

> For SolidFire all-flash storage clusters, software encryption at rest must be enabled during cluster creation and cannot be disabled after the cluster has been created.

What you'll need

- You have cluster administrator privileges to enable or change encryption settings.
- For hardware-based encryption at rest, you have ensured that the cluster is in a healthy state before changing encryption settings.
- If you are disabling encryption, two nodes must be participating in a cluster to access the key to disable encryption on a drive.

Check encryption at rest status

To see the current status of encryption at rest and/or software encryption at rest on the cluster, use the GetClusterInfo method. You can use the GetSoftwareEncryptionAtRestInfo method to get information the cluster uses to encrypt data at rest.

> The Element software UI dashboard at `https://<MVIP>/` currently only shows encryption at rest status for hardware-based encryption.

Options

- Enable hardware-based encryption at rest
- Enable software-based encryption at rest
- Disable hardware-based encryption at rest

Enable hardware-based encryption at rest

> To enable encryption at rest using an external key management configuration, you must enable encryption at rest via the API. Enabling using the existing Element UI button will revert to using internally generated keys.

1. From the Element UI, select **Cluster** > **Settings**.
2. Select **Enable Encryption at Rest**.

Enable software-based encryption at rest

Software encryption at rest cannot be disabled after it is enabled on the cluster.

1. During cluster creation, run the create cluster method with `enableSoftwareEncryptionAtRest` set to `true`.

Disable hardware-based encryption at rest

1. From the Element UI, select **Cluster** > **Settings**.
2. Select **Disable Encryption at Rest**.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Set the cluster full threshold

You can change the level at which the system generates a block cluster fullness warning using the steps below. In addition, you can use the ModifyClusterFullThreshold API method to change the level at which the system generates a block or metadata warning.

What you'll need

You must have cluster administrator privileges.

Steps

1. Click **Cluster** > **Settings**.
2. In the Cluster Full Settings section, enter a percentage in **Raise a warning alert when _% capacity remains before Helix could not recover from a node failure**.
3. Click **Save Changes**.

Find more information

How are the blockSpace thresholds calculated for Element

Enable and disable support access

You can enable support access to temporarily allow NetApp support personnel access to storage nodes via SSH for troubleshooting.

You must have cluster admin privileges to change support access.

1. Click **Cluster** > **Settings**.
2. In the Enable / Disable Support Access section, enter the duration (in hours) that you want to allow support to have access.
3. Click **Enable Support Access**.
4. **Optional:** To disable support access, click **Disable Support Access**.

Manage the Terms of Use banner

You can enable, edit, or configure a banner that contains a message for the user.

Options

Enable the Terms of Use banner
Edit the Terms of Use banner
Disable the Terms of Use banner

Enable the Terms of Use banner

You can enable a Terms of Use banner that appears when a user logs in to the Element UI. When the user clicks on the banner, a text dialog box appears containing the message you have configured for the cluster. The banner can be dismissed at any time.

You must have cluster administrator privileges to enable Terms of Use functionality.

1. Click **Users** > **Terms of Use**.
2. In the **Terms of Use** form, enter the text to be displayed for the Terms of Use dialog box.

   Do not exceed 4096 characters.

3. Click **Enable**.

Edit the Terms of Use banner

You can edit the text that a user sees when they select the Terms of Use login banner.

What you'll need

- You must have cluster administrator privileges to configure Terms of Use.
- Ensure that the Terms of Use feature is enabled.

Steps

1. Click **Users** > **Terms of Use**.
2. In the **Terms of Use** dialog box, edit the text that you want to appear.

   Do not exceed 4096 characters.

3. Click **Save Changes**.

Disable the Terms of Use banner

You can disable the Terms of Use banner. With the banner disabled, the user is no longer requested to accept the terms of use when using the Element UI.

What you'll need

- You must have cluster administrator privileges to configure Terms of Use.
- Ensure that Terms of Use is enabled.

Steps

1. Click **Users** > **Terms of Use**.

2. Click **Disable**.

Set the Network Time Protocol

Setting up the Network Time Protocol (NTP) can be achieved in one of two ways: either instruct each node in a cluster to listen for broadcasts or or instruct each node to query an NTP server for updates.

The NTP is used to synchronize clocks over a network. Connection to an internal or external NTP server should be part of the initial cluster setup.

Configure Network Time Protocol servers for the cluster to query

You can instruct each node in a cluster to query a Network Time Protocol (NTP) server for updates. The cluster contacts only configured servers and requests NTP information from them.

Configure NTP on the cluster to point to a local NTP server. You can use the IP address or the FQDN host name. The default NTP server at cluster creation time is set to us.pool.ntp.org; however, a connection to this site cannot always be made depending on the physical location of the SolidFire cluster.

Using the FQDN depends on whether the individual storage node's DNS settings are in place and operational. To do so, configure the DNS servers on every storage node and ensure that the ports are open by reviewing the Network Port Requirements page.

You can enter up to five different NTP servers.

You can use both IPv4 and IPv6 addresses.

What you'll need

You must have cluster administrator privileges to configure this setting.

Steps

1. Configure a list of IPs and/or FQDNs in the server settings.
2. Ensure that the DNS is set properly on the nodes.
3. Click **Cluster** > **Settings**.
4. Under Network Time Protocol Settings, select **No**, which uses the standard NTP configuration.
5. Click **Save Changes**.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Configure the cluster to listen for NTP broadcasts

By using the broadcast mode, you can instruct each node in a cluster to listen on the network for Network Time Protocol (NTP) broadcast messages from a particular server.

What you'll need

- You must have cluster administrator privileges to configure this setting.
- You must configure an NTP server on your network as a broadcast server.

Steps

1. Click **Cluster** > **Settings**.
2. Enter the NTP server or servers that are using broadcast mode into the server list.
3. Under Network Time Protocol Settings, select **Yes** to use a broadcast client.
4. To set the broadcast client, in the **Server** field, enter the NTP server you configured in broadcast mode.
5. Click **Save Changes**.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Manage SNMP

You can configure Simple Network Management Protocol (SNMP) in your cluster.

You can select an SNMP requestor, select which version of SNMP to use, identify the SNMP User-based Security Model (USM) user, and configure traps to monitor the SolidFire cluster. You can also view and access management information base files.

You can use both IPv4 and IPv6 addresses.

SNMP details

On the SNMP page of the Cluster tab, you can view the following information.

- **SNMP MIBs**

  The MIB files that are available for you to view or download.

- **General SNMP Settings**

  You can enable or disable SNMP. After you enable SNMP, you can choose which version to use. If using version 2, you can add requestors, and if using version 3, you can set up USM users.

- **SNMP Trap Settings**

  You can identify which traps you want to capture. You can set the host, port, and community string for each trap recipient.

Configure an SNMP requestor

When SNMP version 2 is enabled, you can enable or disable a requestor, and configure requestors to receive authorized SNMP requests.

1. Click **Cluster › SNMP**.

2. Under **General SNMP Settings**, click **Yes** to enable SNMP.
3. From the **Version** list, select **Version 2**.
4. In the **Requestors** section, enter the **Community String** and **Network** information.

   By default, the community string is public, and the network is localhost. You can change these default settings.

5. **Optional:** To add another requestor, click **Add a Requestor** and enter the **Community String** and **Network** information.
6. Click **Save Changes**.

Find more information

- Configure SNMP traps
- View managed object data using management information base files

Configure an SNMP USM user

When you enable SNMP version 3, you need to configure a USM user to receive authorized SNMP requests.

1. Click **Cluster** > **SNMP**.
2. Under **General SNMP Settings**, click **Yes** to enable SNMP.
3. From the **Version** list, select **Version 3**.
4. In the **USM Users** section, enter the name, password, and passphrase.
5. **Optional:** To add another USM user, click **Add a USM User** and enter the name, password, and passphrase.
6. Click **Save Changes**.

Configure SNMP traps

System administrators can use SNMP traps, also referred to as notifications, to monitor the health of the SolidFire cluster.

When SNMP traps are enabled, the SolidFire cluster generates traps associated with event log entries and system alerts. To receive SNMP notifications, you need to choose the traps that should be generated and identify the recipients of the trap information. By default, no traps are generated.

1. Click **Cluster** > **SNMP**.
2. Select one or more types of traps in the **SNMP Trap Settings** section that the system should generate:
   - Cluster Fault Traps
   - Cluster Resolved Fault Traps
   - Cluster Event Traps
3. In the **Trap Recipients** section, enter the host, port, and community string information for a recipient.
4. **Optional**: To add another trap recipient, click **Add a Trap Recipient** and enter host, port, and community string information.

5. Click **Save Changes**.

View managed object data using management information base files

You can view and download the management information base (MIB) files used to define each of the managed objects. The SNMP feature supports read-only access to those objects defined in the SolidFire-StorageCluster-MIB.

The statistical data provided in the MIB shows system activity for the following:

- Cluster statistics
- Volume statistics
- Volumes by account statistics
- Node statistics
- Other data such as reports, errors, and system events

The system also supports access to the MIB file containing the upper level access points (OIDS) to SF-Series products.

Steps

1. Click **Cluster** > **SNMP**.
2. Under **SNMP MIBs**, click the MIB file you want to download.
3. In the resulting download window, open or save the MIB file.

Manage drives

Each node contains one or more physical drives that are used to store a portion of the data for the cluster. The cluster utilizes the capacity and performance of the drive after the drive has been successfully added to a cluster. You can use the Element UI to manage drives.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Drives details

The Drives page on the Cluster tab provides a list of the active drives in the cluster. You can filter the page by selecting from the Active, Available, Removing, Erasing, and Failed tabs.

When you first initialize a cluster, the active drives list is empty. You can add drives that are unassigned to a cluster and listed in the Available tab after a new SolidFire cluster is created.

The following elements appear in the list of active drives.

- **Drive ID**

The sequential number assigned to the drive.

- **Node ID**

  The node number assigned when the node is added to the cluster.

- **Node Name**

  The name of the node that houses the drive.

- **Slot**

  The slot number where the drive is physically located.

- **Capacity**

  The size of the drive, in GB.

- **Serial**

  The serial number of the drive.

- **Wear Remaining**

  The wear level indicator.

  The storage system reports the approximate amount of wear available on each solid-state drive (SSD) for writing and erasing data. A drive that has consumed 5 percent of its designed write and erase cycles reports 95 percent wear remaining. The system does not refresh drive wear information automatically; you can refresh or close and reload the page to refresh the information.

- **Type**

  The type of drive. The type can be either block or metadata.

Manage nodes

You can manage SolidFire storage and Fibre Channel nodes from the Nodes page of the Cluster tab.

If a newly added node accounts for more than 50 percent of the total cluster capacity, some of the capacity of this node is made unusable ("stranded"), so that it complies with the capacity rule. This remains the case until more storage is added. If a very large node is added that also disobeys the capacity rule, the previously stranded node will no longer be stranded, while the newly added node becomes stranded. Capacity should always be added in pairs to avoid this happening. When a node becomes stranded, an appropriate cluster fault is thrown.

Find more information

Add a node to a cluster

Add a node to a cluster

You can add nodes to a cluster when more storage is needed or after cluster creation.

Nodes require initial configuration when they are first powered on. After the node is configured, it appears in the list of pending nodes and you can add it to a cluster.

The software version on each node in a cluster must be compatible. When you add a node to a cluster, the cluster installs the cluster version of NetApp Element software on the new node as needed.

You can add nodes of smaller or larger capacities to an existing cluster. You can add larger node capacities to a cluster to allow for capacity growth. Larger nodes added to a cluster with smaller nodes must be added in pairs. This allows for sufficient space for Double Helix to move the data should one of the larger nodes fail. You can add smaller node capacities to a larger node cluster to improve performance.

If a newly added node accounts for more than 50 percent of the total cluster capacity, some of the capacity of this node is made unusable ("stranded"), so that it complies with the capacity rule. This remains the case until more storage is added. If a very large node is added that also disobeys the capacity rule, the previously stranded node will no longer be stranded, while the newly added node becomes stranded. Capacity should always be added in pairs to avoid this happening. When a node becomes stranded, the strandedCapacity cluster fault is thrown.

NetApp video: Scale on Your Terms: Expanding a SolidFire Cluster

You can add nodes to NetApp HCI appliances.

Steps

1. Select **Cluster** > **Nodes**.
2. Click **Pending** to view the list of pending nodes.

   When the process for adding nodes is complete, they appear in the Active nodes list. Until then, pending nodes appear in the Pending Active list.

   SolidFire installs the Element software version of the cluster on the pending nodes when you add them to a cluster. This might take a few minutes.

3. Do one of the following:
   - To add individual nodes, click the **Actions** icon for the node you want to add.
   - To add multiple nodes, select the check box of the nodes to add, and then **Bulk Actions**.
     Note: If the node you are adding has a different version of Element software than the version running on the cluster, the cluster asynchronously updates the node to the version of Element software running on the cluster master. After the node is updated, it automatically adds itself to the cluster. During this asynchronous process, the node will be in a pendingActive state.
4. Click **Add**.

   The node appears in the list of active nodes.

Find more information

Node versioning and compatibility

Node versioning and compatibility

Node compatibility is based on the Element software version installed on a node. Element software-based storage clusters automatically image a node to the Element software

version on the cluster if the node and cluster are not at compatible versions.

The following list describes the software release significance levels that make up the Element software version number:

- **Major**

  The first number designates a software release. A node with one major component number cannot be added to a cluster containing nodes of a different major-patch number, nor can a cluster be created with nodes of mixed major versions.

- **Minor**

  The second number designates smaller software features or enhancements to existing software features that have been added to a major release. This component is incremented within a major version component to indicate that this incremental release is not compatible with any other Element software incremental releases with a different minor component. For example, 11.0 is not compatible with 11.1, and 11.1 is not compatible with 11.2.

- **Micro**

  The third number designates a compatible patch (incremental release) to the Element software version represented by the major.minor components. For example, 11.0.1 is compatible with 11.0.2, and 11.0.2 is compatible with 11.0.3.

Major and minor version numbers must match for compatibility. Micro numbers do not have to match for compatibility.

Cluster capacity in a mixed node environment

You can mix different types of nodes in a cluster. The SF-Series 2405, 3010, 4805, 6010, 9605, 9010, 19210, 38410 and the H-Series can coexist in a cluster.

The H-Series consists of H610S-1, H610S-2, H610S-4, and H410S nodes. These nodes are both 10GbE and 25GbE capable.

It is best to not intermix non-encrypted and encrypted nodes. In a mixed node cluster, no node can be larger than 33 percent of the total cluster capacity. For instance, in a cluster with four SF-Series 4805 nodes, the largest node that can be added alone is an SF-Series 9605. The cluster capacity threshold is calculated based on the potential loss of the largest node in this situation.

Depending on your Element software version, the following SF-series storage nodes are not supported:

Beginning with…	Storage node not supported…
Element 12.7	• SF2405 • SF9608
Element 12.0	• SF3010 • SF6010 • SF9010

If you attempt to upgrade one of these nodes to an unsupported Element version, you will see an error stating that this node is not supported by Element 12.x.

View node details

You can view details for individual nodes such as service tags, drive details, and graphics for utilization and drive statistics. The Nodes page of the Cluster tab provides the Version column where you can view the software version of each node.

Steps

1. Click **Cluster** > **Nodes**.
2. To view the details for a specific node, click the **Actions** icon for a node.
3. Click **View Details**.
4. Review the node details:
   - **Node ID**: The system-generated ID for the node.
   - **Node Name**: The host name for the node.
   - **Available 4k IOPS**: The IOPS configured for the node.
   - **Node Role**: The role that the node has in the cluster. Possible values:
     - Cluster Master: The node that performs cluster-wide administrative tasks and contains the MVIP and SVIP.
     - Ensemble Node: A node that participates in the cluster. There are either 3 or 5 ensemble nodes depending on cluster size.
     - Fibre Channel: A node in the cluster.
   - **Node Type**: The model type of the node.
   - **Active Drives**: The number of active drives in the node.
   - **Management IP**: The management IP (MIP) address assigned to node for 1GbE or 10GbE network admin tasks.
   - **Cluster IP**: The cluster IP (CIP) address assigned to the node used for the communication between nodes in the same cluster.
   - **Storage IP**: The storage IP (SIP) address assigned to the node used for iSCSI network discovery and all data network traffic.
   - **Management VLAN ID**: The virtual ID for the management local area network.
   - **Storage VLAN ID**: The virtual ID for the storage local area network.
   - **Version**: The version of software running on each node.
   - **Replication Port**: The port used on nodes for remote replication.
   - **Service Tag**: The unique service tag number assigned to the node.

View Fibre Channel ports details

You can view details of Fibre Channel ports such as its status, name, and port address from the FC Ports page.

View information about the Fibre Channel ports that are connected to the cluster.

Steps

1. Click **Cluster** > **FC Ports**.
2. To filter information on this page, click **Filter**.
3. Review the details:
   - **Node ID**: The node hosting the session for the connection.
   - **Node Name**: System-generated node name.
   - **Slot**: Slot number where the Fibre Channel port is located.
   - **HBA Port**: Physical port on the Fibre Channel host bus adapter (HBA).
   - **WWNN**: The world wide node name.
   - **WWPN**: The target world wide port name.
   - **Switch WWN**: World wide name of the Fibre Channel switch.
   - **Port State**: Current state of the port.
   - **nPort ID**: The node port ID on the Fibre Channel fabric.
   - **Speed**: The negotiated Fibre Channel speed. Possible values are as follows:
     - 4Gbps
     - 8Gbps
     - 16Gbps

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Manage virtual networks

Virtual networking in SolidFire storage enables traffic between multiple clients that are on separate logical networks to be connected to one cluster. Connections to the cluster are segregated in the networking stack through the use of VLAN tagging.

Find more information

- Add a virtual network
- Enable virtual routing and forwarding
- Edit a virtual network
- Edit VRF VLANs
- Delete a virtual network

Add a virtual network

You can add a new virtual network to a cluster configuration to enable a multi-tenant environment connection to a cluster running Element software.

What you'll need

- Identify the block of IP addresses that will be assigned to the virtual networks on the cluster nodes.

- Identify a storage network IP (SVIP) address that will be used as an endpoint for all NetApp Element storage traffic.

You must consider the following criteria for this configuration:

- VLANs that are not VRF-enabled require initiators to be in the same subnet as the SVIP.
- VLANs that are VRF-enabled do not require initiators to be in the same subnet as the SVIP, and routing is supported.
- The default SVIP does not require initiators to be in the same subnet as the SVIP, and routing is supported.

When a virtual network is added, an interface for each node is created and each requires a virtual network IP address. The number of IP addresses you specify when creating a new virtual network must be equal to or greater than the number of nodes in the cluster. Virtual network addresses are bulk provisioned by and assigned to individual nodes automatically. You do not need to manually assign virtual network addresses to the nodes in the cluster.

Steps

1. Click **Cluster** > **Network**.
2. Click **Create VLAN**.
3. In the **Create a New VLAN** dialog box, enter values in the following fields:
   - **VLAN Name**
   - **VLAN Tag**
   - **SVIP**
   - **Netmask**
   - (Optional) **Description**
4. Enter the **Starting IP** address for the range of IP addresses in **IP Address Blocks**.
5. Enter the **Size** of the IP range as the number of IP addresses to include in the block.
6. Click **Add a Block** to add a non-continuous block of IP addresses for this VLAN.
7. Click **Create VLAN**.

View virtual network details

Steps

1. Click **Cluster** > **Network**.
2. Review the details.
   - **ID**: Unique ID of the VLAN network, which is assigned by the system.
   - **Name**: Unique user-assigned name for the VLAN network.
   - **VLAN Tag**: VLAN tag assigned when the virtual network was created.
   - **SVIP**: Storage virtual IP address assigned to the virtual network.
   - **Netmask**: Netmask for this virtual network.
   - **Gateway**: Unique IP address of a virtual network gateway. VRF must be enabled.
   - **VRF Enabled**: Indication of whether virtual routing and forwarding is enabled or not.
   - **IPs Used**: The range of virtual network IP addresses used for the virtual network.

Enable virtual routing and forwarding

You can enable virtual routing and forwarding (VRF), which allows multiple instances of a routing table to exist in a router and work simultaneously. This functionality is available for storage networks only.

You can enable VRF only at the time of creating a VLAN. If you want to switch back to non-VRF, you must delete and re-create the VLAN.

1. Click **Cluster** > **Network**.
2. To enable VRF on a new VLAN, select **Create VLAN**.
   a. Enter relevant information for the new VRF/VLAN. See Adding a virtual network.
   b. Select the **Enable VRF** check box.
   c. **Optional**: Enter a gateway.
3. Click **Create VLAN**.

Find more information

Add a virtual network

Edit a virtual network

You can change VLAN attributes, such as VLAN name, netmask, and size of the IP address blocks. The VLAN tag and SVIP cannot be modified for a VLAN. The gateway attribute is not a valid parameter for non-VRF VLANs.

If any iSCSI, remote replication, or other network sessions exist, the modification might fail.

When managing the size of VLAN IP address ranges, you should note the following limitations:

- You can only remove IP addresses from the initial IP address range assigned at the time the VLAN was created.
- You can remove an IP address block that was added after the initial IP address range, but you cannot resize an IP block by removing IP addresses.
- When you try to remove IP addresses, from either the initial IP address range or in an IP block, that are in use by nodes in the cluster, the operation might fail.
- You cannot reassign specific in-use IP addresses to other nodes in the cluster.

You can add an IP address block by using the following procedure:

1. Select **Cluster** > **Network**.
2. Select the Actions icon for the VLAN you want to edit.
3. Select **Edit**.
4. In the **Edit VLAN** dialog box, enter the new attributes for the VLAN.
5. Select **Add a Block** to add a non-continuous block of IP addresses for the virtual network.
6. Select **Save Changes**.

Link to troubleshooting KB articles

Link to the Knowledge Base articles for help with troubleshooting issues with managing your VLAN IP address ranges.

- Duplicate IP warning after adding a storage node in VLAN on Element cluster
- How to determine which VLAN IP's are in use and which nodes those IP's are assigned to in Element

Edit VRF VLANs

You can change VRF VLAN attributes, such as VLAN name, netmask, gateway, and IP address blocks.

1. Click **Cluster** > **Network**.
2. Click the Actions icon for the VLAN you want to edit.
3. Click **Edit**.
4. Enter the new attributes for the VRF VLAN in the **Edit VLAN** dialog box.
5. Click **Save Changes**.

Delete a virtual network

You can remove a virtual network object. You must add the address blocks to another virtual network before you remove a virtual network.

1. Click **Cluster** > **Network**.
2. Click the Actions icon for the VLAN you want to delete.
3. Click **Delete**.
4. Confirm the message.

Find more information

Edit a virtual network

Create a cluster supporting FIPS drives

Security is becoming increasingly critical for the deployment of solutions in many customer environments. Federal Information Processing Standards (FIPS) are standards for computer security and interoperability. FIPS 140-2 certified encryption for data at rest is a component of the overall security solution.

- Avoid mixing nodes for FIPS drives
- Enable encryption at rest
- Identify whether nodes are ready for the FIPS drives feature
- Enable the FIPS drives feature
- Check the FIPS drive status
- Troubleshoot the FIPS drive feature

Avoid mixing nodes for FIPS drives

To prepare for enabling the FIPS drives feature, you should avoid mixing nodes where some are FIPS drives capable and some are not.

A cluster is considered FIPS drives compliant based on the following conditions:

- All drives are certified as FIPS drives.
- All nodes are FIPS drives nodes.
- Encryption at Rest (EAR) is enabled.
- The FIPS drives feature is enabled. All drives and nodes must be FIPS capable and Encryption at Rest must be enabled in order to enable the FIPS drive feature.

Enable encryption at rest

You can enable and disable cluster-wide encryption at rest. This feature is not enabled by default. To support FIPS drives, you must enable encryption at rest.

1. In the NetApp Element software UI, click **Cluster** > **Settings**.
2. Click **Enable Encryption at Rest**.

Find more information

- Enable and disable encryption for a cluster
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Identify whether nodes are ready for the FIPS drives feature

You should check to see if all nodes in the storage cluster are ready to support FIPS drives by using the NetApp Element software GetFipsReport API method.

The resulting report shows one of the following statuses:

- None: Node is not capable of supporting the FIPS drives feature.
- Partial: Node is FIPS capable, but not all drives are FIPS drives.
- Ready: Node is FIPS capable and all drives are FIPS drives or no drives are present.

Steps

1. Using the Element API, check to see if the nodes and drives in the storage cluster are capable of FIPS drives by entering:

   ```
   GetFipsReport
   ```

2. Review the results, noting any nodes that did not display a status of Ready.
3. For any nodes that did not display a Ready status, check to see if the drive is capable of supporting the FIPS drives feature:
   - Using the Element API, enter: `GetHardwareList`

- Note the value of the **DriveEncryptionCapabilityType**. If it is "fips," the hardware can support the FIPS drives feature.

  See details about `GetFipsReport` or `ListDriveHardware` in the Element API Reference.

4. If the drive cannot support the FIPS drives feature, replace the hardware with FIPS hardware (either node or drives).

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Enable the FIPS drives feature

You can enable the FIPS drives feature by using the NetApp Element software `EnableFeature` API method.

Encryption at Rest must be enabled on the cluster and all nodes and drives must be FIPS capable, as indicated when the GetFipsReport displays a Ready status for all nodes.

Step

1. Using the Element API, enable FIPS on all drives by entering:

   ```
   EnableFeature params: FipsDrives
   ```

Find more information

- Manage storage with the Element API
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Check the FIPS drive status

You can check whether the FIPS drives feature is enabled on the cluster by using the NetApp Element software `GetFeatureStatus` API method, which shows whether the FIPS Drives Enabled Status is true or false.

1. Using the Element API, check the FIPS drives feature on the cluster by entering:

   ```
   GetFeatureStatus
   ```

2. Review the results of the `GetFeatureStatus` API call. If the FIPS Drives enabled value is True, the FIPS drives feature is enabled.

   ```
   {"enabled": true,
   "feature": "FipsDrives"
   }
   ```

Find more information

- Manage storage with the Element API
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Troubleshoot the FIPS drive feature

Using the NetApp Element software UI, you can view alerts for information about cluster faults or errors in the system related to the FIPS drives feature.

1. Using the Element UI, select **Reporting** > **Alerts**.
2. Look for cluster faults including:
   - FIPS drives mismatched
   - FIPS drives out of compliance
3. For resolution suggestions, see Cluster Fault code information.

Find more information

- Cluster fault codes
- Manage storage with the Element API
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Enable FIPS 140-2 for HTTPS on your cluster

You can use the EnableFeature API method to enable the FIPS 140-2 operating mode for HTTPS communications.

With NetApp Element software, you can choose to enable Federal Information Processing Standards (FIPS) 140-2 operating mode on your cluster. Enabling this mode activates the NetApp Cryptographic Security Module (NCSM) and leverages FIPS 140-2 Level 1 certified encryption for all communication via HTTPS to the NetApp Element UI and API.

> After you enable FIPS 140-2 mode, it cannot be disabled. When FIPS 140-2 mode is enabled, each node in the cluster reboots and runs through a self-test ensuring that the NCSM is correctly enabled and operating in the FIPS 140-2 certified mode. This causes an interruption to both management and storage connections on the cluster. You should plan carefully and only enable this mode if your environment needs the encryption mechanism it offers.

For more information, see the Element API information.

The following is an example of the API request to enable FIPS:

```
{
  "method": "EnableFeature",
    "params": {
        "feature" : "fips"
    },
    "id": 1
}
```

After this operating mode is enabled, all HTTPS communication uses the FIPS 140-2 approved ciphers.

Find more information

- SSL ciphers
- Manage storage with the Element API
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

SSL ciphers

SSL ciphers are encryption algorithms used by hosts to establish a secure communication. There are standard ciphers that Element software supports and non-standard ones when FIPS 140-2 mode is enabled.

The following lists provide the standard Secure Socket Layer (SSL) ciphers supported by Element software and the SSL ciphers supported when FIPS 140-2 mode is enabled:

- **FIPS 140-2 disabled**

  TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048) - A

  TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A

  TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048) - A

  TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A

  TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (secp256r1) - A

  TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A

  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (secp256r1) - A

  TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp256r1) - A

  TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C

  TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A

  TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A

TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A

TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A

TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A

TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A

TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (rsa 2048) - A

TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A

TLS_RSA_WITH_IDEA_CBC_SHA (rsa 2048) - A

TLS_RSA_WITH_RC4_128_MD5 (rsa 2048) - C

TLS_RSA_WITH_RC4_128_SHA (rsa 2048) - C

TLS_RSA_WITH_SEED_CBC_SHA (rsa 2048) - A

- **FIPS 140-2 enabled**

TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (dh 2048) - A

TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A

TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (dh 2048) - A

TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (sect571r1) - A

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (secp256r1) - A

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (sect571r1) - A

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (sect571r1) - A

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (secp256r1) - A

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp256r1) - A

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (sect571r1) - A

TLS_RSA_WITH_3DES_EDE_CBC_SHA (rsa 2048) - C

TLS_RSA_WITH_AES_128_CBC_SHA (rsa 2048) - A

TLS_RSA_WITH_AES_128_CBC_SHA256 (rsa 2048) - A

TLS_RSA_WITH_AES_128_GCM_SHA256 (rsa 2048) - A

TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A

TLS_RSA_WITH_AES_256_CBC_SHA256 (rsa 2048) - A

TLS_RSA_WITH_AES_256_GCM_SHA384 (rsa 2048) - A

Find more information

Enable FIPS 140-2 for HTTPS on your cluster

Get started with external key management

External key management (EKM) provides secure Authentication Key (AK) management in conjunction with an off-cluster external key server (EKS). The AKs are used to lock and unlock Self Encrypting Drives (SEDs) when encryption at rest is enabled on the cluster. The EKS provides secure generation and storage of the AKs. The cluster utilizes the Key Management Interoperability Protocol (KMIP), an OASIS defined standard protocol, to communicate with the EKS.

- Set up external management
- Rekey software encryption at rest master key
- Recover inaccessible or invalid authentication keys
- External key management API commands

Find more information

- CreateCluster API that can be used to enable software encryption at rest
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Set up external key management

You can follow these steps and use the Element API methods listed to set up your external key management feature.

What you'll need

- If you are setting up external key management in combination with software encryption at rest, you have enabled software encryption at rest using the CreateCluster method on a new cluster that does not contain volumes.

Steps

1. Establish a trust relationship with the External Key Server (EKS).
   a. Create a public/private key pair for the Element cluster that is used to establish a trust relationship with the key server by calling the following API method: CreatePublicPrivateKeyPair
   b. Get the certificate sign request (CSR) which the Certification Authority needs to sign. The CSR enables the key server to verify that the Element cluster that will be accessing the keys is authenticated as the Element cluster. Call the following API method: GetClientCertificateSignRequest
   c. Use the EKS/Certificate Authority to sign the retrieved CSR. See third-party documentation for more information.
2. Create a server and provider on the cluster to communicate with the EKS. A key provider defines where a key should be obtained, and a server defines the specific attributes of the EKS that will be communicated

with.

a. Create a key provider where the key server details will reside by calling the following API method: CreateKeyProviderKmip

b. Create a key server providing the signed certificate and the public key certificate of the Certification Authority by calling the following API methods: CreateKeyServerKmip
TestKeyServerKmip

   If the test fails, verify your server connectivity and configuration. Then repeat the test.

c. Add the key server into the key provider container by calling the following API methods:
AddKeyServerToProviderKmip
TestKeyProviderKmip

   If the test fails, verify your server connectivity and configuration. Then repeat the test.

3. Do one of the following as a next step for encryption at rest:

a. (For hardware encryption at rest) Enable hardware encryption at rest by providing the ID of the key provider that contains the key server used for storing the keys by calling the EnableEncryptionAtRest API method.

> You must enable encryption at rest via the API. Enabling encryption at rest using the existing Element UI button will cause the feature to revert to using internally generated keys.

b. (For software encryption at rest) In order for software encryption at rest to utilize the newly created key provider, pass the key provider ID to the RekeySoftwareEncryptionAtRestMasterKey API method.

Find more information

- Enable and disable encryption for a cluster
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Rekey software encryption at rest master key

You can use the Element API to rekey an existing key. This process creates a new replacement master key for your external key management server. Master keys are always replaced by new master keys and never duplicated or overwritten.

You might need to rekey as part of one of the following procedures:

- Create a new key as part of a change from internal key management to external key management.
- Create a new key as a reaction to or as protection against a security-related event.

> This process is asynchronous and returns a response before the rekey operation is complete. You can use the GetAsyncResult method to poll the system to see when the process has completed.

What you'll need

- You have enabled software encryption at rest using the CreateCluster method on a new cluster that does

not contain volumes and has no I/O. Use `GetSoftwareEncryptionatRestInfo` to confirm that the state is `enabled` before proceeding.

- You have established a trust relationship between the SolidFire cluster and an External Key Server (EKS). Run the TestKeyProviderKmip method to verify that a connection to the key provider is established.

Steps

1. Run the ListKeyProvidersKmip command and copy the key provider ID (`keyProviderID`).
2. Run the RekeySoftwareEncryptionAtRestMasterKey with the `keyManagementType` parameter as `external` and `keyProviderID` as the ID number of the key provider from the previous step:

```
{
 "method": "rekeysoftwareencryptionatrestmasterkey",
 "params": {
   "keyManagementType": "external",
   "keyProviderID": "<ID number>"
 }
}
```

3. Copy the `asyncHandle` value from the `RekeySoftwareEncryptionAtRestMasterKey` command response.
4. Run the GetAsyncResult command with the `asyncHandle` value from the previous step to confirm the change in configuration. From the command response, you should see that the older master key configuration has been updated with new key information. Copy the new key provider ID for use in a later step.

```
{
   "id": null,
   "result": {
     "createTime": "2021-01-01T22:29:18Z",
     "lastUpdateTime": "2021-01-01T22:45:51Z",
     "result": {
       "keyToDecommission": {
         "keyID": "<value>",
         "keyManagementType": "internal"
     },
     "newKey": {
       "keyID": "<value>",
       "keyManagementType": "external",
       "keyProviderID": <value>
     },
     "operation": "Rekeying Master Key. Master Key management being
transferred from Internal Key Management to External Key Management with
keyProviderID=<value>",
     "state": "Ready"
   },
   "resultType": "RekeySoftwareEncryptionAtRestMasterKey",
   "status": "complete"
}
```

5. Run the `GetSoftwareEncryptionatRestInfo` command to confirm that new key details, including the `keyProviderID`, have been updated.

```
{
   "id": null,
   "result": {
     "masterKeyInfo": {
       "keyCreatedTime": "2021-01-01T22:29:18Z",
       "keyID": "<updated value>",
       "keyManagementType": "external",
       "keyProviderID": <value>
     },
     "rekeyMasterKeyAsyncResultID": <value>
     "status": "enabled",
     "version": 1
   },
}
```

Find more information

- Manage storage with the Element API
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Recover inaccessible or invalid authentication keys

Occasionally, an error can occur that requires user intervention. In the event of an error, a cluster fault (referred to as a cluster fault code) will be generated. The two most likely cases are described here.

The cluster is unable to unlock the drives due to a KmipServerFault cluster fault.

This can occur when the cluster first boots up and the key server is inaccessible or the required key is unavailable.

1. Follow the recovery steps in the cluster fault codes (if any).

A sliceServiceUnhealthy fault might be set because the metadata drives have been marked as failed and placed into the "Available" state.

Steps to clear:

1. Add the drives again.
2. After 3 to 4 minutes, check that the `sliceServiceUnhealthy` fault has cleared.

See cluster fault codes for more information.

External key management API commands

List of all of the APIs available for managing and configuring EKM.

Used for establishing a trust relationship between the cluster and external customer-owned servers:

- CreatePublicPrivateKeyPair
- GetClientCertificateSignRequest

Used for defining the specific details of external customer-owned servers:

- CreateKeyServerKmip
- ModifyKeyServerKmip
- DeleteKeyServerKmip
- GetKeyServerKmip
- ListKeyServersKmip
- TestKeyServerKmip

Used for creating and maintaining key providers which manage external key servers:

- CreateKeyProviderKmip
- DeleteKeyProviderKmip

- AddKeyServerToProviderKmip
- RemoveKeyServerFromProviderKmip
- GetKeyProviderKmip
- ListKeyProvidersKmip
- RekeySoftwareEncryptionAtRestMasterKey
- TestKeyProviderKmip

For information about the API methods, see API reference information.

Manage volumes and virtual volumes

You can manage the data in a cluster running Element software from the Management tab in the Element UI. Available cluster management functions include creating and managing data volumes, volume access groups, initiators, and Quality of Service (QoS) policies.

- Work with volumes
- Work with virtual volumes
- Work with volume access groups and initiators

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Work with volumes

The SolidFire system provisions storage using volumes. Volumes are block devices accessed over the network by iSCSI or Fibre Channel clients. From the Volumes page on the Management tab, you can create, modify, clone, and delete volumes on a node. You can also view statistics about volume bandwidth and I/O usage.

Find more information

- Manage Quality of Service policies
- Create a volume
- View individual volume performance details
- Edit active volumes
- Delete a volume
- Restore a deleted volume
- Purge a volume
- Clone a volume
- Assign LUNs to Fibre Channel volumes

- Apply a QoS policy to volumes
- Remove the QoS policy association of a volume

Manage Quality of Service policies

A Quality of Service (QoS) policy enables you to create and save a standardized quality of service setting that can be applied to many volumes. You can create, edit, and delete QoS policies from the QoS Policies page on the Management tab.

If you are using QoS policies, do not use custom QoS on a volume. Custom QoS will override and adjust QoS policy values for volume QoS settings.

NetApp video: SolidFire Quality of Service Policies

See Performance and quality of service.

- Create a QoS policy
- Edit a QoS policy
- Delete a QoS policy

Create a QoS policy

You can create QoS policies and apply them when creating volumes.

1. Select **Management** > **QoS Policies**.
2. Click **Create QoS Policy**.
3. Enter the **Policy Name**.
4. Enter the**Min IOPS**, **Max IOPS**, and **Burst IOPS** values.
5. Click **Create QoS Policy**.

Edit a QoS policy

You can change the name of an existing QoS policy or edit the values associated with the policy. Changing a QoS policy affects all volumes associated with the policy.

1. Select **Management** > **QoS Policies**.
2. Click the Actions icon for the QoS policy you want to edit.
3. In the resulting menu, select**Edit**.
4. In the **Edit QoS Policy** dialog box, modify the following properties as required:
   - Policy Name
   - Min IOPS
   - Max IOPS
   - Burst IOPS
5. Click **Save Changes**.

Delete a QoS policy

You can delete a QoS policy if it is no longer needed. When you delete a QoS policy, all volumes associated with the policy maintain the QoS settings but become unassociated with a policy.

If you are trying instead to disassociate a volume from a QoS policy, you can change the QoS settings for that volume to custom.

1. Select **Management** > **QoS Policies**.
2. Click the Actions icon for the QoS policy you want to delete.
3. In the resulting menu, select **Delete**.
4. Confirm the action.

Find more information

- Remove the QoS policy association of a volume
- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Manage volumes

The SolidFire system provisions storage using volumes. Volumes are block devices accessed over the network by iSCSI or Fibre Channel clients.

From the Volumes page on the Management tab, you can create, modify, clone, and delete volumes on a node.

Create a volume

You can create a volume and associate the volume with a given account. Every volume must be associated with an account. This association gives the account access to the volume through the iSCSI initiators using the CHAP credentials.

You can specify QoS settings for a volume during creation.

1. Select **Management** > **Volumes**.
2. Click **Create Volume**.
3. In the **Create a New Volume** dialog box, enter the **Volume Name**.
4. Enter the total size of the volume.

   The default volume size selection is in GB. You can create volumes using sizes measured in GB or GiB:

   - 1GB = 1 000 000 000 bytes
   - 1GiB = 1 073 741 824 bytes
5. Select a **Block Size** for the volume.
6. Click the **Account** drop-down list and select the account that should have access to the volume.

   If an account does not exist, click the **Create Account** link, enter a new account name, and click **Create**.

The account is created and associated with the new volume.

> If there are more than 50 accounts, the list does not appear. Begin typing and the auto-complete function displays possible values for you to choose.

7. To set the **Quality of Service**, do one of the following:
   a. Under **Policy**, you can select an existing QoS policy, if available.
   b. Under **Custom Settings**, set customized minimum, maximum, and burst values for IOPS or use the default QoS values.

      Volumes that have a Max or Burst IOPS value greater than 20,000 IOPS might require high queue depth or multiple sessions to achieve this level of IOPS on a single volume.

8. Click **Create Volume**.

View volume details

1. Select **Management** > **Volumes**.
2. Review the details.
   - **ID**: The system-generated ID for the volume.
   - **Name**: The name given to the volume when it was created.
   - **Account**: The name of the account assigned to the volume.
   - **Access Groups**: The name of the volume access group or groups to which the volume belongs.
   - **Access**: The type of access assigned to the volume when it was created. Possible values:
     - Read / Write: All reads and writes are accepted.
     - Read Only: All read activity allowed; no writes allowed.
     - Locked: Only Administrator access allowed.
     - ReplicationTarget: Designated as a target volume in a replicated volume pair.
   - **Used**: The percentage of used space in the volume.
   - **Size**: The total size (in GB) of the volume.
   - **Primary Node ID**: The primary node for this volume.
   - **Secondary Node ID**: The list of secondary nodes for this volume. Can be multiple values during transitory states, like change of secondary nodes, but will usually have a single value.
   - **QoS Throttle**: Identifies if the volume is being throttled due to high load on the primary storage node.
   - **QoS Policy**: The name and link to the user-defined QoS policy.
   - **Min IOPS**: The minimum number of IOPS guaranteed for the volume.
   - **Max IOPS**: The maximum number of IOPS allowed for the volume.
   - **Burst IOPS**: The maximum number of IOPS allowed over a short period of time for the volume. Default = 15,000.
   - **Snapshots**: The number of snapshots created for the volume.
   - **Attributes**: Attributes that have been assigned to the volume as a key/value pair through an API method.
   - **512e**: Indication of whether 512e is enabled on a volume. Possible values:

        - Yes
        - No
    - **Created On**: The date and time that the volume was created.

View individual volume details

You can view performance statistics for individual volumes.

1. Select **Reporting** > **Volume Performance**.
2. In the volume list, click the Actions icon for a volume.
3. Click **View Details**.

   A tray appears at the bottom of the page containing general information about the volume.

4. To see more detailed information about the volume, click **See More Details**.

   The system displays detailed information as well as performance graphs for the volume.

Edit active volumes

You can modify volume attributes such as QoS values, volume size, and the unit of measurement in which byte values are calculated. You can also modify account access for replication usage or to restrict access to the volume.

You can resize a volume when there is sufficient space on the cluster under the following conditions:

- Normal operating conditions.
- Volume errors or failures are being reported.
- The volume is being cloned.
- The volume is being resynced.

Steps

1. Select **Management** > **Volumes**.
2. In the **Active** window, click the Actions icon for the volume you want to edit.
3. Click **Edit**.
4. **Optional:** Change the total size of the volume.

    - You can increase, but not decrease, the size of the volume. You can only resize one volume in a single resizing operation. Garbage collection operations and software upgrades do not interrupt the resizing operation.
    - If you are adjusting volume size for replication, you should first increase the size of the volume assigned as the replication target. Then you can resize the source volume. The target volume can be greater or equal in size to the source volume, but it cannot be smaller.

   The default volume size selection is in GB. You can create volumes using sizes measured in GB or GiB:

    - 1GB = 1 000 000 000 bytes
    - 1GiB = 1 073 741 824 bytes
5. **Optional:** Select a different account access level of one of the following:

- Read Only
- Read/Write
- Locked
- Replication Target

6. **Optional:** Select the account that should have access to the volume.

   If the account does not exist, click the **Create Account** link, enter a new account name, and click **Create**. The account is created and associated with the volume.

   If there are more than 50 accounts, the list does not appear. Begin typing and the auto-complete function displays possible values for you to choose.

7. **Optional:** To change the selection in **Quality of Service**, do one of the following:
   a. Under **Policy**, you can select an existing QoS policy, if available.
   b. Under **Custom Settings**, set customized minimum, maximum, and burst values for IOPS or use the default QoS values.

   If you are using QoS policies on a volume, you can set custom QoS to remove the QoS policy affiliation with the volume. Custom QoS will override and adjust QoS policy values for volume QoS settings.

   When you change IOPS values, you should increment in tens or hundreds. Input values require valid whole numbers.

   Configure volumes with an extremely high burst value. This allows the system to process occasional large block sequential workloads more quickly, while still constraining the sustained IOPS for a volume.

8. Click **Save Changes**.

Delete a volume

You can delete one or more volumes from an Element storage cluster.

The system does not immediately purge a deleted volume; the volume remains available for approximately eight hours. If you restore a volume before the system purges it, the volume comes back online and iSCSI connections are restored.

If a volume used to create a snapshot is deleted, its associated snapshots become inactive. When the deleted source volumes are purged, the associated inactive snapshots are also removed from the system.

Persistent volumes that are associated with management services are created and assigned to a new account during installation or upgrade. If you are using persistent volumes, do not modify or delete the volumes or their associated account.

Steps

1. Select **Management** > **Volumes**.
2. To delete a single volume, perform the following steps:

a. Click the Actions icon for the volume you want to delete.
b. In the resulting menu, click **Delete**.
c. Confirm the action.

The system moves the volume to the **Deleted** area on the **Volumes** page.

3. To delete multiple volumes, perform the following steps:
   a. In the list of volumes, check the box next to any volumes you want to delete.
   b. Click **Bulk Actions**.
   c. In the resulting menu, click **Delete**.
   d. Confirm the action.

   The system moves the volumes to the **Deleted** area on the **Volumes** page.

Restore a deleted volume

You can restore a volume in the system if it has been deleted but not yet purged. The system automatically purges a volume approximately eight hours after it has been deleted. If the system has purged the volume, you cannot restore it.

1. Select **Management** > **Volumes**.
2. Click the **Deleted** tab to view the list of deleted volumes.
3. Click the Actions icon for the volume you want to restore.
4. In the resulting menu, click **Restore**.
5. Confirm the action.

   The volume is placed in the **Active** volumes list and iSCSI connections to the volume are restored.

Purge a volume

When a volume is purged, it is permanently removed from the system. All data in the volume is lost.

The system automatically purges deleted volumes eight hours after deletion. However, if you want to purge a volume before the scheduled time, you can do so.

1. Select **Management** > **Volumes**.
2. Click the **Deleted** button.
3. Perform the steps to purge a single volume or multiple volumes.

Option	Steps
Purge a single volume	a. Click the Actions icon for the volume you want to purge. b. Click **Purge**. c. Confirm the action.

Option	Steps
Purge multiple volumes	a. Select the volumes you want to purge. b. Click **Bulk Actions**. c. In the resulting menu, select **Purge**. d. Confirm the action.

Clone a volume

You can create a clone of a single volume or multiple volumes to make a point-in-time copy of the data. When you clone a volume, the system creates a snapshot of the volume and then creates a copy of the data referenced by the snapshot. This is an asynchronous process, and the amount of time the process requires depends on the size of the volume you are cloning and the current cluster load.

The cluster supports up to two running clone requests per volume at a time and up to eight active volume clone operations at a time. Requests beyond these limits are queued for later processing.

Operating systems differ in how they treat cloned volumes. VMware ESXi will treat a cloned volume as a volume copy or snapshot volume. The volume will be an available device to use to create a new datastore. For more information on mounting clone volumes and handling snapshot LUNs, see VMware documentation on mounting a VMFS datastore copy and managing duplicate VMFS datastores.

Before you truncate a cloned volume by cloning to a smaller size, ensure that you prepare the partitions so that they fit into the smaller volume.

Steps

1. Select **Management** > **Volumes**.
2. To clone a single volume, perform the following steps:
   a. In the list of volumes on the **Active** page, click the Actions icon for the volume you want to clone.
   b. In the resulting menu, click **Clone**.
   c. In the **Clone Volume** window, enter a volume name for the newly cloned volume.
   d. Select a size and measurement for the volume using the **Volume Size** spin box and list.

      The default volume size selection is in GB. You can create volumes using sizes measured in GB or GiB:

      - 1GB = 1 000 000 000 bytes
      - 1GiB = 1 073 741 824 bytes

   e. Select the type of access for the newly cloned volume.
   f. Select an account to associate with the newly cloned volume from the **Account** list.

      You can create an account during this step if you click the **Create Account** link, enter an account name, and click **Create**. The system automatically adds the account to the **Account** list after you create it.

3. To clone multiple volumes, perform the following steps:
   a. In the list of volumes on the **Active** page, check the box next to any volumes you want to clone.
   b. Click **Bulk Actions**.
   c. In the resulting menu, select **Clone**.
   d. In the **Clone Multiple Volumes** dialog box, enter a prefix for the cloned volumes in the **New Volume Name Prefix** field.
   e. Select an account to associate with the cloned volumes from the **Account** list.
   f. Select the type of access for the cloned volumes.
4. Click **Start Cloning**.

> Increasing the volume size of a clone results in a new volume with additional free space at the end of the volume. Depending on how you use the volume, you might need to extend partitions or create new partitions in the free space to make use of it.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Assign LUNs to Fibre Channel volumes

You can change the LUN assignment for a Fibre Channel volume in a volume access group. You can also make Fibre Channel volume LUN assignments when you create a volume access group.

Assigning new Fibre Channel LUNs is an advanced function and could have unknown consequences on the connecting host. For example, the new LUN ID might not be automatically discovered on the host, and the host might require a rescan to discover the new LUN ID.

1. Select **Management** > **Access Groups**.
2. Click the Actions icon for the access group you want to edit.
3. In the resulting menu, select**Edit**.
4. Under **Assign LUN IDs** in the **Edit Volume Access Group** dialog box, click the arrow on the **LUN Assignments** list.
5. For each volume in the list that you want to assign a LUN to, enter a new value in the corresponding **LUN** field.
6. Click **Save Changes**.

Apply a QoS policy to volumes

You can bulk apply an existing QoS policy to one or more volumes.

The QoS policy you want to bulk apply must exist.

1. Select **Management** > **Volumes**.
2. In the list of volumes, check the box next to any volumes you want to apply the QoS policy to.

3. Click **Bulk Actions**.
4. In the resulting menu, click **Apply QoS Policy**.
5. Select the QoS policy from the drop-down list.
6. Click **Apply**.

Find more information

Quality of Service policies

Remove the QoS policy association of a volume

You can remove a QoS policy association from a volume by selecting custom QoS settings.

The volume you want to modify should be associated with a QoS policy.

1. Select **Management** > **Volumes**.
2. Click the Actions icon for a volume that contains a QoS policy you want to modify.
3. Click **Edit**.
4. In the resulting menu under **Quality of Service**, click **Custom Settings**.
5. Modify **Min IOPS**, **Max IOPS**, and **Burst IOPS**, or keep the default settings.
6. Click **Save Changes**.

Find more information

Delete a QoS policy

Work with virtual volumes

You can view information and perform tasks for virtual volumes and their associated storage containers, protocol endpoints, bindings, and hosts using the Element UI.

The NetApp Element software storage system ships with the Virtual Volumes (VVols) feature disabled. You must perform a one-time task of manually enabling vSphere VVol functionality through the Element UI.

After you enable the VVol functionality, a VVols tab appears in the user interface that offers VVols-related monitoring and limited management options. Additionally, a storage-side software component known as the VASA Provider acts as a storage awareness service for vSphere. Most VVols commands, such as VVol creation, cloning, and editing, are initiated by a vCenter Server or ESXi host and translated by the VASA Provider to Element APIs for the Element software storage system. Commands to create, delete, and manage storage containers and delete virtual volumes can be initiated using the Element UI.

The majority of configurations necessary for using Virtual Volumes functionality with Element software storage systems are made in vSphere. See the *VMware vSphere Virtual Volumes for SolidFire Storage Configuration Guide* to register the VASA Provider in vCenter, create and manage VVol datastores, and manage storage based on policies.

For Element 12.5 and earlier, do not register more than one NetApp Element VASA provider to a single vCenter instance. Where a second NetApp Element VASA provider is added, this renders all VVOL datastores inaccessible.

VASA support for multiple vCenters is available as an upgrade patch if you have already registered a VASA provider with your vCenter. To install, download the VASA39 .tar.gz file from the NetApp Software Downloads site and follow the directions in the manifest. The NetApp Element VASA provider uses a NetApp certificate. With this patch, the certificate is used unmodified by vCenter to support multiple vCenters for VASA and VVols use. Do not modify the certificate. Custom SSL certificates are not supported by VASA.

Find more information

- Enable virtual volumes
- View virtual volume details
- Delete a virtual volume
- Create a storage container
- Edit a storage container
- Delete a storage container
- Protocol endpoints
- Bindings
- Host details

Enable virtual volumes

You must manually enable vSphere Virtual Volumes (VVols) functionality through the NetApp Element software. The Element software system comes with VVols functionality disabled by default, and it is not automatically enabled as part of a new installation or upgrade. Enabling the VVols feature is a one-time configuration task.

What you'll need

- The cluster must be running Element 9.0 or later.
- The cluster must be connected to an ESXi 6.0 or later environment that is compatible with VVols.
- If you are using Element 11.3 or later, the cluster must be connected to an ESXi 6.0 update 3 or later environment.

Enabling vSphere Virtual Volumes functionality permanently changes the Element software configuration. You should only enable VVols functionality if your cluster is connected to a VMware ESXi VVols-compatible environment. You can disable the VVols feature and restore the default settings only by returning the cluster to the factory image, which deletes all data on the system.

Steps

1. Select **Clusters** > **Settings**.
2. Find the cluster-specific settings for Virtual Volumes.

3. Click **Enable Virtual Volumes**.
4. Click **Yes** to confirm the Virtual Volumes configuration change.

   The **VVols** tab appears in the Element UI.

   When VVols functionality is enabled, the SolidFire cluster starts the VASA Provider, opens port 8444 for VASA traffic, and creates protocol endpoints that can be discovered by vCenter and all ESXi hosts.

5. Copy the VASA Provider URL from the Virtual Volumes (VVols) settings in **Clusters** > **Settings**. You will use this URL to register the VASA Provider in vCenter.
6. Create a storage container in **VVols** > **Storage Containers**.

   You must create at least one storage container so that VMs can be provisioned to a VVol datastore.

7. Select **VVols** > **Protocol Endpoints**.
8. Verify that a protocol endpoint has been created for each node in the cluster.

   Additional configuration tasks are required in vSphere. See the *VMware vSphere Virtual Volumes for SolidFire Storage Configuration Guide* to register the VASA Provider in vCenter, create and manage VVol datastores, and manage storage based on policies.

Find more information

VMware vSphere Virtual Volumes for SolidFire Storage Configuration Guide

View virtual volume details

You can review virtual volume information for all active virtual volumes on the cluster in the Element UI. You can also view performance activity for each virtual volume, including input, output, throughput, latency, queue depth, and volume information.

What you'll need

- You should have enabled VVols functionality in the Element UI for the cluster.
- You should have created an associated storage container.
- You should have configured your vSphere cluster to use Element software VVols functionality.
- You should have created at least one VM in vSphere.

Steps

1. Click **VVols** > **Virtual Volumes**.

   The information for all active virtual volumes is displayed.

2. Click the **Actions** icon for the virtual volume you want to review.
3. In the resulting menu, select **View Details**.

Details

The Virtual Volumes page of the VVols tab provides information about each active virtual volume on the cluster, such as volume ID, snapshot ID, parent virtual volume ID, and virtual volume ID.

- **Volume ID**: The ID of the underlying volume.
- **Snapshot ID**: The ID of the underlying volume snapshot. The value is 0 if the virtual volume does not represent a SolidFire snapshot.
- **Parent Virtual Volume ID**: The virtual volume ID of the parent virtual volume. If the ID is all zeros, the virtual volume is independent with no link to a parent.
- **Virtual Volume ID**: The UUID of the virtual volume.
- **Name**: The name assigned to the virtual volume.
- **Storage Container**: The storage container that owns the virtual volume.
- **Guest OS Type**: Operating system associated with the virtual volume.
- **Virtual Volume Type**: The virtual volume type: Config, Data, Memory, Swap, or Other.
- **Access**: The read-write permissions assigned to the virtual volume.
- **Size**: The size of the virtual volume in GB or GiB.
- **Snapshots**: The number of associated snapshots. Click the number to link to snapshot details.
- **Min IOPS**: The minimum IOPS QoS setting of the virtual volume.
- **Max IOPS**: The maximum IOPS QoS setting of the virtual volume.
- **Burst IOPS**: The maximum burst QoS setting of the virtual volume.
- **VMW_VmID**: Information in fields prefaced with "VMW_" are defined by VMware.
- **Create Time**: The time the virtual volume creation task was completed.

Individual virtual volume details

The Virtual Volumes page on the VVols tab provides the following virtual volume information when you select an individual virtual volume and view its details.

- **VMW_XXX**: Information in fields prefaced with "VMW_" are defined by VMware.
- **Parent Virtual Volume ID**: The virtual volume ID of the parent virtual volume. If the ID is all zeros, the virtual volume is independent with no link to a parent.
- **Virtual Volume ID**: The UUID of the virtual volume.
- **Virtual Volume Type**: The virtual volume type: Config, Data, Memory, Swap, or Other.
- **Volume ID**: The ID of the underlying volume.
- **Access**: The read-write permissions assigned to the virtual volume.
- **Account Name**: Name of the account containing the volume.
- **Access Groups**: Associated volume access groups.
- **Total Volume Size**: Total provisioned capacity in bytes.
- **Non-Zero Blocks**: Total number of 4KiB blocks with data after the last garbage collection operation has completed.
- **Zero Blocks**: Total number of 4KiB blocks without data after the last round of garbage collection operation has completed.

- **Snapshots**: The number of associated snapshots. Click the number to link to snapshot details.
- **Min IOPS**: The minimum IOPS QoS setting of the virtual volume.
- **Max IOPS**: The maximum IOPS QoS setting of the virtual volume.
- **Burst IOPS**: The maximum burst QoS setting of the virtual volume.
- **Enable 512**: Because virtual volumes always use 512-byte block size emulation, the value is always yes.
- **Volumes Paired**: Indicates if a volume is paired.
- **Create Time**: The time the virtual volume creation task was completed.
- **Blocks Size**: Size of the blocks on the volume.
- **Unaligned Writes**: For 512e volumes, the number of write operations that were not on a 4k sector boundary. High numbers of unaligned writes might indicate improper partition alignment.
- **Unaligned Reads**: For 512e volumes, the number of read operations that were not on a 4k sector boundary. High numbers of unaligned reads might indicate improper partition alignment.
- **scsiEUIDeviceID**: Globally unique SCSI device identifier for the volume in EUI-64 based 16-byte format.
- **scsiNAADeviceID**: Globally unique SCSI device identifier for the volume in NAA IEEE Registered Extended format.
- **Attributes**: List of name-value pairs in JSON object format.

Delete a virtual volume

Although virtual volumes should always be deleted from the VMware Management Layer, the functionality for you to delete virtual volumes is enabled from the Element UI. You should only delete a virtual volume from the Element UI when absolutely necessary, such as when vSphere fails to clean up virtual volumes on SolidFire storage.

1. Select **VVols** > **Virtual Volumes**.
2. Click the Actions icon for the virtual volume you want to delete.
3. In the resulting menu, select **Delete**.

> You should delete a virtual volume from the VMware Management Layer to ensure that the virtual volume is properly unbound before deletion. You should only delete a virtual volume from the Element UI when absolutely necessary, such as when vSphere fails to clean up virtual volumes on SolidFire storage. If you delete a virtual volume from the Element UI, the volume will be purged immediately.

4. Confirm the action.
5. Refresh the list of virtual volumes to confirm that the virtual volume has been removed.
6. **Optional**: Select **Reporting** > **Event Log** to confirm that the purge has been successful.

Manage storage containers

A storage container is a vSphere datastore representation created on a cluster running Element software.

Storage containers are created and tied to NetApp Element accounts. A storage container created on Element storage appears as a vSphere datastore in vCenter and ESXi. Storage containers do not allocate any space on

Element storage. They are simply used to logically associate virtual volumes.

A maximum of four storage containers per cluster is supported. A minimum of one storage container is required to enable VVols functionality.

Create a storage container

You can create storage containers in the Element UI and discover them in vCenter. You must create at least one storage container to begin provisioning VVol-backed virtual machines.

Before you begin, enable VVols functionality in the Element UI for the cluster.

Steps

1. Select **VVols** > **Storage Containers**.
2. Click the **Create Storage Containers** button.
3. Enter storage container information in the **Create a New Storage Container** dialog box:
   a. Enter a name for the storage container.
   b. Configure initiator and target secrets for CHAP.

      Leave the CHAP Settings fields blank to automatically generate secrets.

   c. Click the **Create Storage Container** button.
4. Verify that the new storage container appears in the list in the **Storage Containers** sub-tab.

   Because a NetApp Element account ID is created automatically and assigned to the storage container, it is not necessary to manually create an account.

View storage container details

On the Storage Containers page of the VVols tab, you can view information for all active storage containers on the cluster.

- **Account ID**: The ID of the NetApp Element account associated with the storage container.
- **Name**: The name of the storage container.
- **Status**: The status of the storage container. Possible values:
  - Active: The storage container is in use.
  - Locked: The storage container is locked.
- **PE Type**: The protocol endpoint type (SCSI is the only available protocol for Element software).
- **Storage Container ID**: The UUID of the virtual volume storage container.
- **Active Virtual Volumes**: The number of active virtual volumes associated with the storage container.

View individual storage container details

You can view the storage container information for an individual storage container by selecting it from the Storage Containers page on the VVols tab.

- **Account ID**: The ID of the NetApp Element account associated with the storage container.

- **Name**: The name of the storage container.
- **Status**: The status of the storage container. Possible values:
  - Active: The storage container is in use.
  - Locked: The storage container is locked.
- **Chap Initiator Secret**: The unique CHAP secret for the initiator.
- **Chap Target Secret**: The unique CHAP secret for the target.
- **Storage Container ID**: The UUID of the virtual volume storage container.
- **Protocol Endpoint Type**: Indicates the protocol endpoint type (SCSI is the only available protocol).

Edit a storage container

You can modify storage container CHAP authentication in the Element UI.

1. Select **VVols** > **Storage Containers**.
2. Click the **Actions** icon for the storage container you want to edit.
3. In the resulting menu, select **Edit**.
4. Under CHAP Settings, edit the Initiator Secret and Target Secret credentials used for authentication.

   If you do not change the CHAP Settings credentials, they remain the same. If you make the credentials fields blank, the system automatically generates new secrets.

5. Click **Save Changes**.

Delete a storage container

You can delete storage containers from the Element UI.

What you'll need

Ensure that all virtual machines have been removed from the VVol datastore.

Steps

1. Select **VVols** > **Storage Containers**.
2. Click the **Actions** icon for the storage container you want to delete.
3. In the resulting menu, select **Delete**.
4. Confirm the action.
5. Refresh the list of storage containers in the **Storage Containers** sub-tab to confirm that the storage container has been removed.

Protocol endpoints

Protocol endpoints are access points used by a host to address storage on a cluster running NetApp Element software. Protocol endpoints cannot be deleted or modified by a user, are not associated with an account, and cannot be added to a volume access group.

A cluster running Element software automatically creates one protocol endpoint per storage node in the cluster.

For example, a six-node storage cluster has six protocol endpoints that are mapped to each ESXi host. Protocol endpoints are dynamically managed by Element software and are created, moved, or removed as needed without any intervention. Protocol endpoints are the target for multi-pathing and act as an I/O proxy for subsidiary LUNs. Each protocol endpoint consumes an available SCSI address, just like a standard iSCSI target. Protocol endpoints appear as a single-block (512-byte) storage device in the vSphere client, but this storage device is not available to be formatted or used as storage.

iSCSI is the only supported protocol. Fibre Channel protocol is not supported.

Protocol endpoints details

The Protocol Endpoints page on the VVols tab provides protocol endpoint information.

- **Primary Provider ID**

  The ID of the primary protocol endpoint provider.

- **Secondary Provider ID**

  The ID of the secondary protocol endpoint provider.

- **Protocol Endpoint ID**

  The UUID of the protocol endpoint.

- **Protocol Endpoint State**

  The status of the protocol endpoint. Possible values are as follows:

  - Active: The protocol endpoint is in use.
  - Start: The protocol endpoint is starting.
  - Failover: The protocol endpoint has failed over.
  - Reserved: The protocol endpoint is reserved.

- **Provider Type**

  The type of the protocol endpoint's provider. Possible values are as follows:

  - Primary
  - Secondary

- **SCSI NAA Device ID**

  The globally unique SCSI device identifier for the protocol endpoint in NAA IEEE Registered Extended Format.

Bindings

To perform I/O operations with a virtual volume, an ESXi host must first bind the virtual volume.

The SolidFire cluster chooses an optimal protocol endpoint, creates a binding that associates the ESXi host and virtual volume with the protocol endpoint, and returns the binding to the ESXi host. After it is bound, the ESXi host can perform I/O operations with the bound virtual volume.

Bindings details

The Bindings page on the VVols tab provides binding information about each virtual volume.

The following information is displayed:

- **Host ID**

  The UUID for the ESXi host that hosts virtual volumes and is known to the cluster.

- **Protocol Endpoint ID**

  Protocol endpoint IDs that correspond to each node in the SolidFire cluster.

- **Protocol Endpoint in Band ID**

  The SCSI NAA device ID of the protocol endpoint.

- **Protocol Endpoint Type**

  The protocol endpoint type.

- **VVol Binding ID**

  The binding UUID of the virtual volume.

- **VVol ID**

  The universally unique identifier (UUID) of the virtual volume.

- **VVol Secondary ID**

  The secondary ID of the virtual volume that is a SCSI second level LUN ID.

Host details

The Hosts page on the VVols tab provides information about VMware ESXi hosts that host virtual volumes.

The following information is displayed:

- **Host ID**

  The UUID for the ESXi host that hosts virtual volumes and is known to the cluster.

- **Host Address**

  The IP address or DNS name for the ESXi host.

- **Bindings**

  Binding IDs for all virtual volumes bound by the ESXi host.

- **ESX Cluster ID**

The vSphere host cluster ID or vCenter GUID.

- **Initiator IQNs**

  Initiator IQNs for the virtual volume host.

- **SolidFire Protocol Endpoint IDs**

  The protocol endpoints that are currently visible to the ESXi host.

Work with volume access groups and initiators

You can use iSCSI initiators or Fibre Channel initiators to access the volumes defined within volume access groups.

You can create access groups by mapping iSCSI initiator IQNs or Fibre Channel WWPNs in a collection of volumes. Each IQN that you add to an access group can access each volume in the group without requiring CHAP authentication.

There are two types of CHAP authentication methods:

- Account-level CHAP authentication: You can assign CHAP authentication for the account.
- Initiator-level CHAP authentication: You can assign unique CHAP target and secrets for specific initiators without being bound to single CHAP across a single account. This Initiator-level CHAP authentication replaces account level credentials.

Optionally, with per-initiator CHAP, you can enforce initiator authorization and per-initiator CHAP authentication. These options can be defined on a per-initiator basis and an access group can contain a mix of initiators with different options.

Each WWPN that you add to an access group enables Fibre Channel network access to the volumes in the access group.

Volume access groups have the following limits:

- A maximum of 64 IQNs or WWPNs are allowed in an access group.
- An access group can be made up of a maximum of 2000 volumes.
- An IQN or WWPN can belong to only one access group.
- A single volume can belong to a maximum of four access groups.

Find more information

- Create a volume access group
- Add volumes to an access group
- Remove volumes from an access group
- Create an initiator
- Edit an initiator
- Add a single initiator to a volume access group

- Add multiple initiators to a volume access group
- Remove initiators from an access group
- Delete an access group
- Delete an initiator

Create a volume access group

You can create volume access groups by mapping initiators to a collection of volumes for secured access. You can then grant access to the volumes in the group with an account CHAP initiator secret and target secret.

If you use initiator-based CHAP, you can add CHAP credentials for a single initiator in a volume access group, providing more security. This enables you to apply this option for volume access groups that already exist.

Steps

1. Click **Management** > **Access Groups**.
2. Click **Create Access Group**.
3. Enter a name for the volume access group in the **Name** field.
4. Add an initiator to the volume access group in one of the following ways:

Option	Description
Adding a Fibre Channel initiator	a. Under Add Initiators, select an existing Fibre Channel initiator from the Unbound Fibre Channel Initiators list. b. Click **Add FC Initiator**. You can create an initiator during this step if you click the **Create Initiator** link, enter an initiator name, and click **Create**. The system automatically adds the initiator to the Initiators list after you create it. A sample of the format is as follows: `5f:47:ac:c0:5c:74:d4:02`

Option	Description
Adding an iSCSI initiator	Under Add Initiators, select an existing initiator from the Initiators list. **Note:** You can create an initiator during this step if you click the **Create Initiator** link, enter an initiator name, and click **Create**. The system automatically adds the initiator to the Initiators list after you create it. A sample of the format is as follows: `iqn.2010-01.com.solidfire:c2r9.fc0.2100000e1e09bb8b` You can find the initiator IQN for each volume by selecting **View Details** in the Actions menu for the volume on the **Management** > **Volumes** > **Active** list. When you modify an initiator, you can toggle the requiredCHAP attribute to True, which enables you to set the target initiator secret. For details, see API information about the ModifyInitiator API method. Manage storage with the Element API

5. **Optional:** Add more initiators as needed.
6. Under Add Volumes, select a volume from the **Volumes** list.

   The volume appears in the **Attached Volumes** list.
7. **Optional:** Add more volumes as needed.
8. Click **Create Access Group**.

Find more information

Add volumes to an access group

View individual access group details

You can view details for an individual access group, such as attached volumes and initiators, in a graphical format.

1. Click **Management** > **Access Groups**.
2. Click the Actions icon for an access group.
3. Click **View Details**.

Volume access group details

The Access Groups page on the Management tab provides information about volume access groups.

The following information is displayed:

- **ID**: The system-generated ID for the access group.

- **Name**: The name given to the access group when it was created.
- **Active Volumes**: The number of active volumes in the access group.
- **Compression**: The compression efficiency score for the access group.
- **Deduplication**: The deduplication efficiency score for the access group.
- **Thin Provisioning**: The thin provisioning efficiency score for the access group.
- **Overall Efficiency**: The overall efficiency score for the access group.
- **Initiators**: The number of initiators connected to the access group.

Add volumes to an access group

You can add volumes to a volume access group. Each volume can belong to more than one volume access group; you can see the groups that each volume belongs to on the **Active** volumes page.

You can also use this procedure to add volumes to a Fibre Channel volume access group.

1. Click **Management** > **Access Groups**.
2. Click the Actions icon for the access group you want to add volumes to.
3. Click the **Edit** button.
4. Under Add Volumes, select a volume from the **Volumes** list.

   You can add more volumes by repeating this step.

5. Click **Save Changes**.

Remove volumes from an access group

When you remove a volume from an access group, the group no longer has access to that volume.

Modifying CHAP settings in an account or removing initiators or volumes from an access group can cause initiators to lose access to volumes unexpectedly. To verify that volume access will not be lost unexpectedly, always logout iSCSI sessions that will be affected by an account or access group change, and verify that initiators can reconnect to volumes after any changes to initiator settings and cluster settings have been completed.

1. Click **Management** > **Access Groups**.
2. Click the Actions icon for the access group you want to remove volumes from.
3. Click **Edit**.
4. Under Add Volumes in the **Edit Volume Access Group** dialog box, click the arrow on the **Attached Volumes** list.
5. Select the volume you want to remove from the list and click the **x** icon to remove the volume from the list.

   You can remove more volumes by repeating this step.

6. Click **Save Changes**.

Create an initiator

You can create iSCSI or Fibre Channel initiators and optionally assign them aliases.

You can also assign initator-based CHAP attributes by using an API call. To add a CHAP account name and credentials per initiator, you must use the `CreateInitiator` API call to remove and add CHAP access and attributes. Initiator access can be restricted to one or more VLANs by specifying one or more virtualNetworkIDs via the `CreateInitiators` and `ModifyInitiators` API calls. If no virtual networks are specified, the initiator can access all networks.

For details, see the API reference information.
Manage storage with the Element API

Steps

1. Click **Management** > **Initiators**.
2. Click **Create Initiator**.
3. Perform the steps to create a single initiator or multiple initiators:

Option	Steps
Create a single initiator	a. Click **Create a Single Initiator**. b. Enter the IQN or WWPN for the initiator in the **IQN/WWPN** field. c. Enter a friendly name for the initiator in the **Alias** field. d. Click **Create Initiator**.
Create multiple initiators	a. Click **Bulk Create Initiators**. b. Enter a list of IQNs or WWPNs in the text box. c. Click **Add Initiators**. d. Choose an initiator from the resulting list and click the corresponding Add icon in the **Alias** column to add an alias for the initiator. e. Click the check mark to confirm the new alias. f. Click **Create Initiators**.

Edit an initiator

You can change the alias of an existing initiator or add an alias if one does not already exist.

To add a CHAP account name and credentials per initiator, you must use the `ModifyInitiator` API call to remove and add CHAP access and attributes.

See Manage storage with the Element API.

Steps

1. Click **Management** > **Initiators**.
2. Click the Actions icon for the initiator you want to edit.

3. Click **Edit**.
4. Enter a new alias for the initiator in the **Alias** field.
5. Click **Save Changes**.

Add a single initiator to a volume access group

You can add an initiator to an existing volume access group.

When you add an initiator to a volume access group, the initiator has access to all volumes in that volume access group.

You can find the initiator for each volume by clicking the Actions icon and then selecting **View Details** for the volume in the active volumes list.

If you use initiator-based CHAP, you can add CHAP credentials for a single initiator in a volume access group, providing more security. This enables you to apply this option for volume access groups that already exist.

Steps

1. Click **Management** > **Access Groups**.
2. Click the **Actions** icon for the access group you want to edit.
3. Click **Edit**.
4. To add a Fibre Channel initiator to the volume access group, perform the following steps:

   a. Under Add Initiators, select an existing Fibre Channel initiator from the **Unbound Fibre Channel Initiators** list.

   b. Click **Add FC Initiator**.

   You can create an initiator during this step if you click the **Create Initiator** link, enter an initiator name, and click **Create**. The system automatically adds the initiator to the **Initiators** list after you create it.

   A sample of the format is as follows:

```
5f:47:ac:c0:5c:74:d4:02
```

5. To add an iSCSI initiator to the volume access group, under Add Initiators, select an existing initiator from the **Initiators** list.

   You can create an initiator during this step if you click the **Create Initiator** link, enter an initiator name, and click **Create**. The system automatically adds the initiator to the **Initiators** list after you create it.

   The accepted format of an initiator IQN is as follows: iqn.yyyy-mm, in which y and m are digits, followed by text which must only contain digits, lower-case alphabetic characters, a period (.), colon (:) or dash (-).

   A sample of the format is as follows:

```
iqn.2010-01.com.solidfire:c2r9.fc0.2100000e1e09bb8b
```

You can find the initiator IQN for each volume from the **Management** > **Volumes** Active Volumes page by clicking the Actions icon and then selecting **View Details** for the volume.

6. Click **Save Changes**.

Add multiple initiators to a volume access group

You can add multiple initiators to an existing volume access group to allow access to volumes in the volume access group with or without requiring CHAP authentication..

When you add initiators to a volume access group, the initiators have access to all volumes in that volume access group.

You can find the initiator for each volume by clicking the Actions icon and then **View Details** for the volume in the active volumes list.

You can add multiple initiators to an existing volume access group to enable access to volumes and assign unique CHAP credentials for each initiator within that volume access group. This enables you to apply this option for volume access groups that already exist.

You can assign initator-based CHAP attributes by using an API call. To add a CHAP account name and credentials per initiator, you must use the ModifyInitiator API call to remove and add CHAP access and attributes.

For details, see Manage storage with the Element API.

Steps

1. Click **Management** > **Initiators**.
2. Select the initiators you want to add to an access group.
3. Click the **Bulk Actions** button.
4. Click **Add to Volume Access Group**.
5. In the Add to Volume Access Group dialog box, select an access group from the **Volume Access Group** list.
6. Click **Add**.

Remove initiators from an access group

When you remove an initiator from an access group, it can no longer access the volumes in that volume access group. Normal account access to the volume is not disrupted.

Modifying CHAP settings in an account or removing initiators or volumes from an access group can cause initiators to lose access to volumes unexpectedly. To verify that volume access will not be lost unexpectedly, always logout iSCSI sessions that will be affected by an account or access group change, and verify that initiators can reconnect to volumes after any changes to initiator settings and cluster settings have been completed.

Steps

1. Click **Management** > **Access Groups**.
2. Click the **Actions** icon for the access group you want to remove.
3. In the resulting menu, select **Edit**.
4. Under Add Initiators in the **Edit Volume Access Group** dialog box, click the arrow on the **Initiators** list.
5. Select the x icon for each initiator you want to remove from the access group.
6. Click **Save Changes**.

Delete an access group

You can delete an access group when it is no longer needed. You do not need to delete Initiator IDs and Volume IDs from the volume access group before deleting the group. After you delete the access group, group access to the volumes is discontinued.

1. Click **Management** > **Access Groups**.
2. Click the **Actions** icon for the access group you want to delete.
3. In the resulting menu, click **Delete**.
4. To also delete the initiators associated with this access group, select the **Delete initiators in this access group** check box.
5. Confirm the action.

Delete an initiator

You can delete an initiator after it is no longer needed. When you delete an initiator, the system removes it from any associated volume access group. Any connections using the initiator remain valid until the connection is reset.

Steps

1. Click **Management** > **Initiators**.
2. Perform the steps to delete a single initiator or multiple initiators:

Option	Steps
Delete single initiator	a. Click the **Actions** icon for the initiator you want to delete. b. Click **Delete**. c. Confirm the action.
Delete multiple initiators	a. Select the check boxes next to the initiators you want to delete. b. Click the **Bulk Actions** button. c. In the resulting menu, select **Delete**. d. Confirm the action.

Protect your data

NetApp Element software enables you to protect your data in a variety of ways with capabilities such as snapshots for individual volumes or groups of volumes, replication between clusters and volumes running on Element, and replication to ONTAP systems.

- **Snapshots**

  Snapshot-only data protection replicates changed data at specific points of time to a remote cluster. Only those snapshots that are created on the source cluster are replicated. Active writes from the source volume are not.

  Use volume snapshots for data protection

- **Remote replication between clusters and volumes running on Element**

  You can replicate volume data synchronously or asynchronously from either cluster in a cluster pair both running on running on Element for failover and failback scenarios.

  Perform remote replication between clusters running NetApp Element software

- **Replication between Element and ONTAP clusters using SnapMirror technology**

  With NetApp SnapMirror technology, you can replicate snapshots that were taken using Element to ONTAP for disaster recovery purposes. In a SnapMirror relationship, Element is one endpoint and ONTAP is the other.

  Use SnapMirror replication between Element and ONTAP clusters

- **Back up to and restore volumes from SolidFire, S3 or Swift object stores**

  You can back up and restore volumes to other SolidFire storage, as well as secondary object stores that are compatible with Amazon S3 or OpenStack Swift.

  Back up and restore volumes to SolidFire, S3, or Swift object stores

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Use volume snapshots for data protection

A volume snapshot is a point-in-time copy of a volume. You can take a snapshot of a volume and use the snapshot later if you need to roll a volume back to the state it was in at the time the snapshot was created.

Snapshots are similar to volume clones. However, snapshots are simply replicas of volume metadata, so you cannot mount or write to them. Creating a volume snapshot also takes only a small amount of system resources and space, which makes snapshot creation faster than cloning.

You can take a snapshot of an individual volume or a set of volumes.

Optionally, replicate snapshots to a remote cluster and use them as a backup copy of the volume. This enables you to roll back a volume to a specific point in time by using the replicated snapshot. Alternatively, you can create a clone of a volume from a replicated snapshot.

Find more information

- Use individual volume snapshots for data protection
- Using group snapshots for data protection task
- Scheduling a snapshot

Use individual volume snapshots for data protection

A volume snapshot is a point-in-time copy of a volume. You can use an individual volume rather than a group of volumes for the snapshot.

Find more information

- Create a volume snapshot
- Edit snapshot retention
- Deleting a snapshot
- Cloning a volume from a snapshot
- Rolling back a volume to a snapshot
- Backing up a volume snapshot to an Amazon S3 object store
- Backing up a volume snapshot to an OpenStack Swift object store
- Backing up a volume snapshot to a SolidFire cluster

Create a volume snapshot

You can create a snapshot of an active volume to preserve the volume image at any point in time. You can create up to 32 snapshots for a single volume.

1. Click **Management** > **Volumes**.
2. Click the **Actions** icon for the volume you want to use for the snapshot.
3. In the resulting menu, select **Snapshot**.
4. In the **Create Snapshot of Volume** dialog box, enter the new snapshot name.
5. **Optional:** Select the **Include Snapshot in Replication When Paired** check box to ensure that the snapshot is captured in replication when the parent volume is paired.
6. To set the retention for the snapshot, select from one of the following options:
   - Click **Keep Forever** to retain the snapshot on the system indefinitely.
   - Click **Set Retention Period** and use the date spin boxes to choose a length of time for the system to retain the snapshot.
7. To take a single, immediate snapshot, perform the following steps:
   a. Click **Take Snapshot Now**.
   b. Click**Create Snapshot**.

8. To schedule the snapshot to run at a future time, perform the following steps:
    a. Click **Create Snapshot Schedule**.
    b. Enter a **New Schedule Name**.
    c. Choose a **Schedule Type** from the list.
    d. **Optional:** Select the **Recurring Schedule** check box to repeat the scheduled snapshot periodically.
    e. Click **Create Schedule**.

Find more information

Schedule a snapshot

Edit snapshot retention

You can change the retention period for a snapshot to control when or if the system deletes snapshots. The retention period you specify begins when you enter the new interval. When you set a retention period, you can select a period that begins at the current time (retention is not calculated from the snapshot creation time). You can specify intervals in minutes, hours, and days.

Steps

1. Click **Data Protection** > **Snapshots**.
2. Click the **Actions** icon for the snapshot you want to edit.
3. In the resulting menu, click **Edit**.
4. **Optional:** Select the**Include Snapshot in Replication When Paired** check box to ensure that the snapshot is captured in replication when the parent volume is paired.
5. **Optional:** Select a retention option for the snapshot:
    - Click **Keep Forever** to retain the snapshot on the system indefinitely.
    - Click **Set Retention Period** and use the date spin boxes to select a length of time for the system to retain the snapshot.
6. Click **Save Changes**.

Delete a snapshot

You can delete a volume snapshot from a storage cluster running Element software. When you delete a snapshot, the system immediately removes it.

You can delete snapshots that are being replicated from the source cluster. If a snapshot is syncing to the target cluster when you delete it, the sync replication completes and the snapshot is deleted from the source cluster. The snapshot is not deleted from the target cluster.

You can also delete snapshots that have been replicated to the target from the target cluster. The deleted snapshot is kept in a list of deleted snapshots on the target until the system detects that you have deleted the snapshot on the source cluster. When the target detects that you have deleted the source snapshot, the target stops replication of the snapshot.

When you delete a snapshot from the source cluster, the target cluster snapshot is not affected (the reverse is also true).

1. Click **Data Protection** > **Snapshots**.
2. Click the **Actions** icon for the snapshot you want to delete.
3. In the resulting menu, select **Delete**.
4. Confirm the action.

Clone a volume from a snapshot

You can create a new volume from a snapshot of a volume. When you do this, the system uses the snapshot information to clone a new volume using the data contained on the volume at the time the snapshot was created. This process stores information about other snapshots of the volume in the newly created volume.

1. Click **Data Protection** > **Snapshots**.
2. Click the **Actions** icon for the snapshot you want to use for the volume clone.
3. In the resulting menu, click **Clone Volume From Snapshot**.
4. Enter a **Volume Name** in the **Clone Volume From Snapshot** dialog box.
5. Select a **Total Size** and size units for the new volume.
6. Select an **Access** type for the volume.
7. Select an **Account** from the list to associate with the new volume.
8. Click **Start Cloning**.

Roll back a volume to a snapshot

You can roll back a volume to a previous snapshot at any time. This reverts any changes made to the volume since the snapshot was created.

Steps

1. Click **Data Protection** > **Snapshots**.
2. Click the **Actions** icon for the snapshot you want to use for the volume rollback.
3. In the resulting menu, select **Rollback Volume To Snapshot**.
4. **Optional:** To save the current state of the volume before rolling back to the snapshot:
   a. In the **Rollback To Snapshot** dialog box, select **Save volume's current state as a snapshot**.
   b. Enter a name for the new snapshot.
5. Click **Rollback Snapshot**.

Back up a volume snapshot

You can use the integrated backup feature to back up a volume snapshot. You can back up snapshots from a SolidFire cluster to an external object store, or to another SolidFire cluster. When you back up a snapshot to an external object store, you must have a connection to the object store that allows read/write operations.

- Back up a volume snapshot to an Amazon S3 object store
- Back up a volume snapshot to an OpenStack Swift object store

- Back up a volume snapshot to a SolidFire cluster

Back up a volume snapshot to an Amazon S3 object store

You can back up SolidFire snapshots to external object stores that are compatible with Amazon S3.

1. Click**Data Protection** > **Snapshots**.
2. Click the **Actions** icon for the snapshot you want to back up.
3. In the resulting menu, click **Backup to**.
4. In the **Integrated Backup** dialog box under **Backup to**, select **S3**.
5. Select an option under **Data Format**:
   - **Native**: A compressed format readable only by SolidFire storage systems.
   - **Uncompressed**: An uncompressed format compatible with other systems.
6. Enter a hostname to use to access the object store in the **Hostname** field.
7. Enter an access key ID for the account in the **Access Key ID** field.
8. Enter the secret access key for the account in the **Secret Access Key** field.
9. Enter the S3 bucket in which to store the backup in the **S3 Bucket** field.
10. **Optional**: Enter a nametag to append to the prefix in the **Nametag** field.
11. Click **Start Read**.

Back up a volume snapshot to an OpenStack Swift object store

You can back up SolidFire snapshots to secondary object stores that are compatible with OpenStack Swift.

1. Click **Data Protection** > **Snapshots**.
2. Click the **Actions** icon for the snapshot you want to back up.
3. In the resulting menu, click **Backup to**.
4. In the **Integrated Backup** dialog box, under **Backup to**, select **Swift**.
5. Select an option under **Data Format**:
   - **Native**: A compressed format readable only by SolidFire storage systems.
   - **Uncompressed**: An uncompressed format compatible with other systems.
6. Enter a **URL** to use to access the object store.
7. Enter a **Username** for the account.
8. Enter the **Authentication Key** for the account.
9. Enter the **Container** in which to store the backup.
10. **Optional**: Enter a **Nametag**.
11. Click **Start Read**.

Back up a volume snapshot to a SolidFire cluster

You can back up volume snapshots residing on a SolidFire cluster to a remote SolidFire cluster.

Ensure that the source and target clusters are paired.

When backing up or restoring from one cluster to another, the system generates a key to be used as authentication between the clusters. This bulk volume write key allows the source cluster to authenticate with the destination cluster, providing a level of security when writing to the destination volume. As part of the backup or restore process, you need to generate a bulk volume write key from the destination volume before starting the operation.

1. On the destination cluster, click **Management** > **Volumes**.
2. Click the **Actions** icon for the destination volume.
3. In the resulting menu, click **Restore from**.
4. In the **Integrated Restore** dialog box under **Restore from**, select **SolidFire**.
5. Select a data format under **Data Format**:
   - **Native**: A compressed format readable only by SolidFire storage systems.
   - **Uncompressed**: An uncompressed format compatible with other systems.
6. Click **Generate Key**.
7. Copy the key from the **Bulk Volume Write Key** box to your clipboard.
8. On the source cluster, click **Data Protection** > **Snapshots**.
9. Click the Actions icon for the snapshot you want to use for the backup.
10. In the resulting menu, click **Backup to**.
11. In the**Integrated Backup** dialog box under **Backup to**, select **SolidFire**.
12. Select the same data format you selected earlier in the **Data Format** field.
13. Enter the management virtual IP address of the destination volume's cluster in the **Remote Cluster MVIP** field.
14. Enter the remote cluster user name in the **Remote Cluster Username** field.
15. Enter the remote cluster password in the **Remote Cluster Password** field.
16. In the **Bulk Volume Write Key** field, paste the key you generated on the destination cluster earlier.
17. Click **Start Read**.

Using group snapshots for data protection task

You can create a group snapshot of a related set of volumes to preserve a point-in-time copy of the metadata for each volume. You can use the group snapshot in the future as a backup or rollback to restore the state of the group of volumes to a previous state.

Find more information

- Create a group snapshot
- Edit group snapshots

- Edit members of group snapshot
- Delete a group snapshot
- Roll back volumes to a group snapshot
- Clone multiple volumes
- Clone multiple volumes from a group snapshot

Group snapshot details

The Group Snapshots page on the Data Protection tab provides information about the group snapshots.

- **ID**

  The system-generated ID for the group snapshot.

- **UUID**

  The unique ID of the group snapshot.

- **Name**

  User-defined name for the group snapshot.

- **Create Time**

  The time at which the group snapshot was created.

- **Status**

  The current status of the snapshot. Possible values:

  - Preparing: The snapshot is being prepared for use and is not yet writable.
  - Done: This snapshot has finished preparation and is now usable.
  - Active: The snapshot is the active branch.

- **# Volumes**

  The number of volumes in the group.

- **Retain Until**

  The day and time the snapshot will be deleted.

- **Remote Replication**

  Indication of whether or not the snapshot is enabled for replication to a remote SolidFire cluster. Possible values:

  - Enabled: The snapshot is enabled for remote replication.
  - Disabled: The snapshot is not enabled for remote replication.

Creating a group snapshot

You can create a snapshot of a group of volumes, and you can also create a group snapshot schedule to automate group snapshots. A single group snapshot can consistently snapshot up to 32 volumes at one time.

Steps

1. Click **Management** > **Volumes**.
2. Use the check boxes to select multiple volumes for a group of volumes.
3. Click **Bulk Actions**.
4. Click **Group Snapshot**.
5. Enter a new group snapshot name in the Create Group Snapshot of Volumes dialog box.
6. **Optional:** Select the **Include Each Group Snapshot Member in Replication When Paired** check box to ensure that each snapshot is captured in replication when the parent volume is paired.
7. Select a retention option for the group snapshot:
   - Click **Keep Forever** to retain the snapshot on the system indefinitely.
   - Click **Set Retention Period** and use the date spin boxes to choose a length of time for the system to retain the snapshot.
8. To take a single, immediate snapshot, perform the following steps:
   a. Click **Take Group Snapshot Now**.
   b. Click **Create Group Snapshot**.
9. To schedule the snapshot to run at a future time, perform the following steps:
   a. Click **Create Group Snapshot Schedule**.
   b. Enter a **New Schedule Name**.
   c. Select a **Schedule Type** from the list.
   d. **Optional:** Select the **Recurring Schedule** check box to repeat the scheduled snapshot periodically.
   e. Click **Create Schedule**.

Editing group snapshots

You can edit the replication and retention settings for existing group snapshots.

1. Click **Data Protection** > **Group Snapshots**.
2. Click the Actions icon for the group snapshot you want to edit.
3. In the resulting menu, select **Edit**.
4. **Optional:** To change the replication setting for the group snapshot:
   a. Click **Edit** next to **Current Replication**.
   b. Select the **Include Each Group Snapshot Member in Replication When Paired** check box to ensure that each snapshot is captured in replication when the parent volume is paired.
5. **Optional:** To change the retention setting for the group snapshot, select from the following options:
   a. Click **Edit** next to **Current Retention**.
   b. Select a retention option for the group snapshot:

- Click **Keep Forever** to retain the snapshot on the system indefinitely.
- Click **Set Retention Period** and use the date spin boxes to choose a length of time for the system to retain the snapshot.

6. Click **Save Changes**.

Deleting a group snapshot

You can delete a group snapshot from the system. When you delete the group snapshot, you can choose whether all snapshots associated with the group are deleted or retained as individual snapshots.

If you delete a volume or snapshot that is a member of a group snapshot, you can no longer roll back to the group snapshot. However, you can roll back each volume individually.

1. Click **Data Protection** > **Group Snapshots**.
2. Click the Actions icon for the snapshot you want to delete.
3. In the resulting menu, click **Delete**.
4. Select from one of the following options in the confirmation dialog box:
   - Click **Delete group snapshot AND all group snapshot members** to delete the group snapshot and all member snapshots.
   - Click **Retain group snapshot members as individual snapshots** to delete the group snapshot but keep all member snapshots.
5. Confirm the action.

Roll back volumes to a group snapshot

You can roll back a group of volumes at any time to a group snapshot.

When you roll back a group of volumes, all volumes in the group are restored to the state they were in at the time the group snapshot was created. Rolling back also restores volume sizes to the size recorded in the original snapshot. If the system has purged a volume, all snapshots of that volume were also deleted at the time of the purge; the system does not restore any deleted volume snapshots.

1. Click **Data Protection** > **Group Snapshots**.
2. Click the Actions icon for the group snapshot you want to use for the volume rollback.
3. In the resulting menu, select **Rollback Volumes To Group Snapshot**.
4. **Optional**: To save the current state of the volumes before rolling back to the snapshot:
   a. In the **Rollback To Snapshot** dialog box, select **Save volumes' current state as a group snapshot**.
   b. Enter a name for the new snapshot.
5. Click **Rollback Group Snapshot**.

Editing members of group snapshot

You can edit the retention settings for members of an existing group snapshot.

1. Click **Data Protection** > **Snapshots**.
2. Click the **Members** tab.

3. Click the Actions icon for the group snapshot member you want to edit.
4. In the resulting menu, select **Edit**.
5. To change the replication setting for the snapshot, select from the following options:
   - Click **Keep Forever** to retain the snapshot on the system indefinitely.
   - Click **Set Retention Period** and use the date spin boxes to choose a length of time for the system to retain the snapshot.
6. Click **Save Changes**.

Clone multiple volumes

You can create multiple volume clones in a single operation to create a point-in-time copy of the data on a group of volumes.

When you clone a volume, the system creates a snapshot of the volume and then creates a new volume from the data in the snapshot. You can mount and write to the new volume clone. Cloning multiple volumes is an asynchronous process and takes a variable amount of time depending on the size and number of the volumes being cloned.

Volume size and current cluster load affect the time needed to complete a cloning operation.

Steps

1. Click **Management** > **Volumes**.
2. Click the **Active** tab.
3. Use the check boxes to select multiple volumes, creating a group of volumes.
4. Click **Bulk Actions**.
5. Click **Clone** in the resulting menu.
6. Enter a **New Volume Name Prefix** in the **Clone Multiple Volumes** dialog box.

   The prefix is applied to all volumes in the group.

7. **Optional:** Select a different account to which the clone will belong.

   If you do not select an account, the system assigns the new volumes to the current volume account.

8. **Optional:** Select a different access method for the volumes in the clone.

   If you do not select an access method, the system uses the current volume access.

9. Click **Start Cloning**.

Cloning multiple volumes from a group snapshot

You can clone a group of volumes from a point-in-time group snapshot. This operation requires that a group snapshot of the volumes already exist, because the group snapshot is used as the basis to create the volumes. After you create the volumes, you can use them like any other volume in the system.

Volume size and current cluster load affect the time needed to complete a cloning operation.

1. Click **Data Protection** > **Group Snapshots**.
2. Click the Actions icon for the group snapshot you want to use for the volume clones.
3. In the resulting menu, select **Clone Volumes From Group Snapshot**.
4. Enter a **New Volume Name Prefix** in the **Clone Volumes From Group Snapshot** dialog box.

   The prefix is applied to all volumes created from the group snapshot.

5. **Optional:** Select a different account to which the clone will belong.

   If you do not select an account, the system assigns the new volumes to the current volume account.

6. **Optional:** Select a different access method for the volumes in the clone.

   If you do not select an access method, the system uses the current volume access.

7. Click **Start Cloning**.

Schedule a snapshot

You can protect data on a volume or a group of volumes by scheduling volume snapshots to occur at specified intervals. You can schedule either single volume snapshots or group snapshots to run automatically.

When you configure a snapshot schedule, you can choose from time intervals based on days of the week or days of the month. You can also specify the days, hours, and minutes before the next snapshot occurs. You can store the resulting snapshots on a remote storage system if the volume is being replicated.

Find more information

- Create a snapshot schedule
- Edit a snapshot schedule
- Delete a snapshot schedule
- Copy a snapshot schedule

Snapshot schedule details

On the Data Protection > Schedules page, you can view the following information in the list of snapshot schedules.

- **ID**

  The system-generated ID for the snapshot.

- **Type**

  The type of schedule. Snapshot is currently the only type supported.

- **Name**

  The name given to the schedule when it was created. Snapshot schedule names can be up to 223 characters in length and contain a-z, 0-9, and dash (-) characters.

- **Frequency**

  The frequency at which the schedule is run. The frequency can be set in hours and minutes, weeks, or months.

- **Recurring**

  Indication of whether the schedule is to run only once or at regular intervals.

- **Manually Paused**

  Indication of whether or not the schedule has been manually paused.

- **Volume IDs**

  The ID of the volume the schedule will use when the schedule is run.

- **Last Run**

  The last time the schedule was run.

- **Last Run Status**

  The outcome of the last schedule execution. Possible values:

  - Success
  - Failure

Create a snapshot schedule

You can schedule a snapshot of a volume or volumes to automatically occur at specified intervals.

When you configure a snapshot schedule, you can choose from time intervals based on days of the week or days of the month. You can also create a recurring schedule and specify the days, hours, and minutes before the next snapshot occurs.

If you schedule a snapshot to run at a time period that is not divisible by 5 minutes, the snapshot will run at the next time period that is divisible by 5 minutes. For example, if you schedule a snapshot to run at 12:42:00 UTC, it will run at 12:45:00 UTC. You cannot schedule a snapshot to run at intervals of less than 5 minutes.

Beginning with Element 12.5, you can enable serial creation and select to retain the snapshots on a First-In-First-Out (FIFO) basis from the UI.

- The **Enable Serial Creation** option specifies that only one snapshot is replicated at a time. The creation of a new snapshot fails when a previous snapshot replication is still in progress. If the checkbox is not selected, a snapshot creation is allowed when another snapshot replication is still in progress.
- The **FIFO** option adds the capability to retain a consistent number of the latest snapshots. When the checkbox is selected, snapshots are retained on a FIFO basis. After the queue of FIFO snapshots reaches its maximum depth, the oldest FIFO snapshot is discarded when a new FIFO snapshot is inserted.

Steps

1. Select **Data Protection** > **Schedules**.

2. Select **Create Schedule**.
3. In the **Volume IDs CSV** field, enter a single volume ID or a comma-separated list of volume IDs to include in the snapshot operation.
4. Enter a new schedule name.
5. Select a schedule type and set the schedule from the options provided.
6. **Optional:** Select **Recurring Schedule** to repeat the snapshot schedule indefinitely.
7. **Optional:** Enter a name for the new snapshot in the **New Snapshot Name** field.

   If you leave the field blank, the system uses the time and date of the snapshot's creation as the name.
8. **Optional:** Select the **Include Snapshots in Replication When Paired** check box to ensure that the snapshots are captured in replication when the parent volume is paired.
9. **Optional:** Select the **Enable Serial Creation** check box to ensure that only one snapshot is replicated at a time.
10. To set the retention for the snapshot, select from the following options:
    - **Optional:** Select the **FIFO (First In First out)** check box to retain a consistent number of the latest snapshots.
    - Select **Keep Forever** to retain the snapshot on the system indefinitely.
    - Select **Set Retention Period** and use the date spin boxes to choose a length of time for the system to retain the snapshot.
11. Select **Create Schedule**.

Edit a snapshot schedule

You can modify existing snapshot schedules. After modification, the next time the schedule runs it uses the updated attributes. Any snapshots created by the original schedule remain on the storage system.

Steps

1. Click **Data Protection** > **Schedules**.
2. Click the **Actions** icon for the schedule you want to change.
3. In the resulting menu, click **Edit**.
4. In the **Volume IDs CSV** field, modify the single volume ID or comma-separated list of volume IDs currently included in the snapshot operation.
5. To pause or resume the schedule, select from the following options:
    - To pause an active schedule, select **Yes** from the **Manually Pause Schedule** list.
    - To resume a paused schedule, select **No** from the **Manually Pause Schedule** list.
6. Enter a different name for the schedule in the **New Schedule Name** field if desired.
7. To change the schedule to run on different days of the week or month, select **Schedule Type** and change the schedule from the options provided.
8. **Optional:** Select **Recurring Schedule** to repeat the snapshot schedule indefinitely.
9. **Optional:** Enter or modify the name for the new snapshot in the **New Snapshot Name** field.

   If you leave the field blank, the system uses the time and date of the snapshot's creation as the name.

10. **Optional:** Select the **Include Snapshots in Replication When Paired** check box to ensure that the snapshots are captured in replication when the parent volume is paired.
11. To change the retention setting, select from the following options:
    - Click **Keep Forever** to retain the snapshot on the system indefinitely.
    - Click **Set Retention Period** and use the date spin boxes to select a length of time for the system to retain the snapshot.
12. Click **Save Changes**.

Copy a snapshot schedule

You can copy a schedule and maintain its current attributes.

1. Click **Data Protection** > **Schedules**.
2. Click the Actions icon for the schedule you want to copy.
3. In the resulting menu, click **Make a Copy**.

   The **Create Schedule** dialog box appears, populated with the current attributes of the schedule.

4. **Optional:** Enter a name and updated attributes for the new schedule.
5. Click **Create Schedule**.

Delete a snapshot schedule

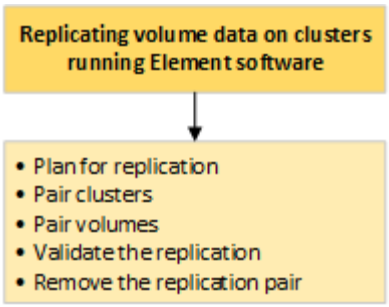
You can delete a snapshot schedule. After you delete the schedule, it does not run any future scheduled snapshots. Any snapshots that were created by the schedule remain on the storage system.

1. Click **Data Protection** > **Schedules**.
2. Click the **Actions** icon for the schedule you want to delete.
3. In the resulting menu, click **Delete**.
4. Confirm the action.

Perform remote replication between clusters running NetApp Element software

For clusters running Element software, real-time replication enables the quick creation of remote copies of volume data. You can pair a storage cluster with up to four other storage clusters. You can replicate volume data synchronously or asynchronously from either cluster in a cluster pair for failover and failback scenarios.

The replication process includes these steps:



- Plan cluster and volume pairing for real-time replication
- Pair clusters for replication
- Pair volumes
- Validate volume replication
- Delete a volume relationship after replication
- Manage volume relationships

Plan cluster and volume pairing for real-time replication

Real-time remote replication requires that you pair two storage clusters running Element software, pair volumes on each cluster, and validate replication. After replication completes, you should delete the volume relationship.

What you'll need

- You must have cluster administrator privileges to one or both clusters being paired.
- All node IP addresses on both management and storage networks for paired clusters are routed to each other.
- MTU of all paired nodes must be the same and be supported end-to-end between clusters.
- Both storage clusters should have unique cluster names, MVIPs, SVIPs., and all node IP addresses.
- The difference between Element software versions on the clusters is no greater than one major version. If the difference is greater, one of the clusters must be upgraded to perform data replication.

> WAN Accelerator appliances have not been qualified by NetApp for use when replicating data. These appliances can interfere with compression and deduplication if deployed between two clusters that are replicating data. Be sure to fully qualify the effects of any WAN Accelerator appliance before you deploy it in a production environment.

Find more information

- Pair clusters for replication
- Pair volumes
- Assign a replication source and target to paired volumes

Pair clusters for replication

You must pair two clusters as a first step to using real-time replication functionality. After

you pair and connect two clusters, you can configure active volumes on one cluster to be continuously replicated to a second cluster, providing continuous data protection (CDP).

What you'll need

- You must have cluster administrator privileges to one or both clusters being paired.
- All node MIPs and SIPs are routed to each other.
- Less than 2000 ms of round-trip latency between clusters.
- Both storage clusters should have unique cluster names, MVIPs, SVIPs, and all node IP addresses.
- The difference between Element software versions on the clusters is no greater than one major version. If the difference is greater, one of the clusters must be upgraded to perform data replication.

Cluster pairing requires full connectivity between nodes on the management network. Replication requires connectivity between the individual nodes on the storage cluster network.

You can pair one cluster with up to four other clusters for replicating volumes. You can also pair clusters within the cluster group with each other.

Find more information

Network port requirements

Pair clusters using MVIP or a pairing key

You can pair a source and target cluster using the MVIP of the target cluster if there is cluster administrator access to both clusters. If cluster administrator access is only available on one cluster in a cluster pair, a pairing key can be used on the target cluster to complete the cluster pairing.

1. Select one of the following methods to pair clusters:
   - Pair clusters using MVIP: Use this method if there is cluster administrator access to both clusters. This method uses the MVIP of the remote cluster to pair two clusters.
   - Pair clusters using a pairing key: Use this method if there is cluster administrator access to only one of the clusters. This method generates a pairing key that can be used on the target cluster to complete the cluster pairing.

Find more information

- Pair clusters using MVIP
- Pair clusters using a pairing key

Pair clusters using MVIP

You can pair two clusters for real-time replication by using the MVIP of one cluster to establish a connection with the other cluster. Cluster administrator access on both of clusters is required to use this method. The cluster administrator user name and password is used to authenticate cluster access before the clusters can be paired.

1. On the local cluster, select **Data Protection** > **Cluster Pairs**.

2. Click **Pair Cluster**.
3. Click **Start Pairing** and click **Yes** to indicate that you have access to the remote cluster.
4. Enter the remote cluster MVIP address.
5. Click **Complete pairing on remote cluster**.

   In the **Authentication Required** window, enter the cluster administrator user name and password of the remote cluster.

6. On the remote cluster, select **Data Protection** > **Cluster Pairs**.
7. Click **Pair Cluster**.
8. Click **Complete Pairing**.
9. Click the **Complete Pairing** button.

Find more information

- Pair clusters using a pairing key
- Pairing clusters using MVIP (video)

Pair clusters using a pairing key

If you have cluster administrator access to a local cluster but not the remote cluster, you can pair the clusters using a pairing key. A pairing key is generated on a local cluster and then sent securely to a cluster administrator at a remote site to establish a connection and complete the cluster pairing for real-time replication.

1. On the local cluster, select **Data Protection** > **Cluster Pairs**.
2. Click **Pair Cluster**.
3. Click **Start Pairing** and click **No** to indicate that you do not have access to the remote cluster.
4. Click **Generate Key**.

   This action generates a text key for pairing and creates an unconfigured cluster pair on the local cluster. If you do not complete the procedure, you will need to manually delete the cluster pair.

5. Copy the cluster pairing key to your clipboard.
6. Make the pairing key accessible to the cluster administrator at the remote cluster site.

   The cluster pairing key contains a version of the MVIP, user name, password, and database information to permit volume connections for remote replication. This key should be treated in a secure manner and not stored in a way that would allow accidental or unsecured access to the user name or password.

   Do not modify any of the characters in the pairing key. The key becomes invalid if it is modified.

7. On the remote cluster, select **Data Protection** > **Cluster Pairs**.

8. Click **Pair Cluster**.
9. Click **Complete Pairing** and enter the pairing key in the **Pairing Key** field (paste is the recommended method).
10. Click **Complete Pairing**.

Find more information

- Pair clusters using MVIP
- Pairing clusters using a cluster pairing key (video)

Validate the cluster pair connection

After the cluster pairing has completed, you might want to verify the cluster pair connection to ensure replication success.

1. On the local cluster, select **Data Protection** > **Cluster Pairs**.
2. In the **Cluster Pairs** window, verify that the cluster pair is connected.
3. **Optional:** Navigate back to the local cluster and the **Cluster Pairs** window and verify that the cluster pair is connected.

Pair volumes

After you have established a connection between clusters in a cluster pair, you can pair a volume on one cluster with a volume on the other cluster in the pair. When a volume pairing relationship is established, you must identify which volume is the replication target.

You can pair two volumes for real-time replication that are stored on different storage clusters in a connected cluster pair. After you pair two clusters, you can configure active volumes on one cluster to be continuously replicated to a second cluster, providing continuous data protection (CDP). You can also assign either volume to be the source or target of the replication.

Volume pairings are always one-to-one. After a volume is part of a pairing with a volume on another cluster, you cannot pair it again with any other volume.

What you'll need

- You have established a connection between clusters in a cluster pair.
- You have cluster administrator privileges to one or both clusters being paired.

Steps

1. Create a target volume with read or write access
2. Pair volumes using a volume ID or pairing key
3. Assign a replication source and target to paired volumes

Create a target volume with read or write access

The replication process involves two endpoints: the source and the target volume. When you create the target volume, the volume is automatically set to read/write mode to accept the data during the replication.

1. Select **Management** > **Volumes**.
2. Click **Create Volume**.
3. In the Create a New Volume dialog box, enter the Volume Name.
4. Enter the total size of the volume, select a block size for the volume, and select the account that should have access to the volume.
5. Click **Create Volume**.
6. In the Active window, click the Actions icon for the volume.
7. Click **Edit**.
8. Change the account access level to Replication Target.
9. Click **Save Changes**.

Pair volumes using a volume ID or pairing key

The pairing process involves pairing two volumes by using either a volume ID or a pairing key.

1. Pair volumes by selecting one of the following methods:
   - Using a volume ID: Use this method if you have cluster administrator access to both clusters on which volumes are to be paired. This method uses the volume ID of the volume on the remote cluster to initiate a connection.
   - Using a pairing Key: Use this method if you have cluster administrator access to only the source cluster. This method generates a pairing key that can be used on the remote cluster to complete the volume pair.

   > The volume pairing key contains an encrypted version of the volume information and might contain sensitive information. Only share this key in a secure manner.

Find more information

- Pair volumes using a volume ID
- Pair volumes using a pairing key

Pair volumes using a volume ID

You can pair a volume with another volume on a remote cluster if you have cluster administrator credentials for the remote cluster.

What you'll need

- Ensure that the clusters containing the volumes are paired.
- Create a new volume on the remote cluster.

You can assign a replication source and target after the pairing process. A replication source or target can be either volume in a volume pair. You should create a target volume that contains no data and has the exact characteristics of the source volume, such as size, block size setting for the volumes (either 512e or 4k), and QoS configuration. If you assign an existing volume as the replication target, the data on that volume will be overwritten. The target volume can be greater or equal in size to the source volume, but it cannot be smaller.

- Know the target Volume ID.

Steps

1. Select **Management** > **Volumes**.
2. Click the **Actions** icon for the volume you want to pair.
3. Click **Pair**.
4. In the **Pair Volume** dialog box, select **Start Pairing**.
5. Select **I Do** to indicate that you have access to the remote cluster.
6. Select a **Replication Mode** from the list:
   - **Real-time (Asynchronous)**: Writes are acknowledged to the client after they are committed on the source cluster.
   - **Real-time (Synchronous)**: Writes are acknowledged to the client after they are committed on both the source and target clusters.
   - **Snapshots Only**: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.
7. Select a remote cluster from the list.
8. Choose a remote volume ID.
9. Click **Start Pairing**.

   The system opens a web browser tab that connects to the Element UI of the remote cluster. You might be required to log on to the remote cluster with cluster administrator credentials.

10. In the Element UI of the remote cluster, select **Complete Pairing**.
11. Confirm the details in **Confirm Volume Pairing**.
12. Click **Complete Pairing**.

    After you confirm the pairing, the two clusters begin the process of connecting the volumes for pairing. During the pairing process, you can see messages in the **Volume Status** column of the **Volume Pairs** window. The volume pair displays `PausedMisconfigured` until the volume pair source and target are assigned.

    After you successfully complete the pairing, it is recommended that you refresh the Volumes table to remove the **Pair** option from the **Actions** list for the paired volume. If you do not refresh the table, the **Pair** option remains available for selection. If you select the **Pair** option again, a new tab opens and because the volume is already paired, the system reports a `StartVolumePairing Failed: xVolumeAlreadyPaired` error message in the **Pair Volume** window of the Element UI page.

Find more information

- Volume pairing messages

- Volume pairing warnings
- Assign a replication source and target to paired volumes

Pair volumes using a pairing key

If you do not have cluster admin credentials for a remote cluster, you can pair a volume with another volume on a remote cluster using a pairing key.

What you'll need

- Ensure that the clusters containing the volumes are paired.
- Ensure that there is a volume on the remote cluster to use for the pairing.

> You can assign a replication source and target after the pairing process. A replication source or target can be either volume in a volume pair. You should create a target volume that contains no data and has the exact characteristics of the source volume, such as size, block size setting for the volumes (either 512e or 4k), and QoS configuration. If you assign an existing volume as the replication target, the data on that volume will be overwritten. The target volume can be greater or equal in size to the source volume, but it cannot be smaller.

Steps

1. Select **Management** > **Volumes**.
2. Click **Actions** icon for the volume you want to pair.
3. Click **Pair**.
4. In the **Pair Volume** dialog box, select **Start Pairing**.
5. Select **I Do Not** to indicate that you do not have access to the remote cluster.
6. Select a **Replication Mode** from the list:
   - **Real-time (Asynchronous)**: Writes are acknowledged to the client after they are committed on the source cluster.
   - **Real-time (Synchronous)**: Writes are acknowledged to the client after they are committed on both the source and target clusters.
   - **Snapshots Only**: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.
7. Click **Generate Key**.

> This action generates a text key for pairing and creates an unconfigured volume pair on the local cluster. If you do not complete the procedure, you will need to manually delete the volume pair.

8. Copy the pairing key to your computer's clipboard.
9. Make the pairing key accessible to the cluster admin at the remote cluster site.

> The volume pairing key should be treated in a secure manner and not used in a way that would allow accidental or unsecured access.

Do not modify any of the characters in the pairing key. The key becomes invalid if it is modified.

10. In the remote cluster Element UI, select **Management** > **Volumes**.
11. Click the Actions icon for the volume you want to pair.
12. Click **Pair**.
13. In the **Pair Volume** dialog box, select **Complete Pairing**.
14. Paste the pairing key from the other cluster into the **Pairing Key** box.
15. Click **Complete Pairing**.

    After you confirm the pairing, the two clusters begin the process of connecting the volumes for pairing. During the pairing process, you can see messages in the **Volume Status** column of the **Volume Pairs** window. The volume pair displays `PausedMisconfigured` until the volume pair source and target are assigned.

    After you successfully complete the pairing, it is recommended that you refresh the Volumes table to remove the **Pair** option from the **Actions** list for the paired volume. If you do not refresh the table, the **Pair** option remains available for selection. If you select the **Pair** option again, a new tab opens and because the volume is already paired, the system reports a `StartVolumePairing Failed: xVolumeAlreadyPaired` error message in the **Pair Volume** window of the Element UI page.

Find more information

- Volume pairing messages
- Volume pairing warnings
- Assign a replication source and target to paired volumes

Assign a replication source and target to paired volumes

After volumes are paired, you must assign a source volume and its replication target volume. A replication source or target can be either volume in a volume pair. You can also use this procedure to redirect data sent to a source volume to a remote target volume should the source volume become unavailable.

What you'll need

You have access to the clusters containing the source and target volumes.

Steps

1. Prepare the source volume:
   a. From the cluster that contains the volume you want to assign as source, select **Management** > **Volumes**.
   b. Click the **Actions** icon for the volume you want to assign as source and click **Edit**.
   c. In the **Access** drop-down list, select **Read/Write**.

   If you are reversing source and target assignment, this action will cause the volume pair to display the following message until a new replication target is assigned: `PausedMisconfigured`

Changing access pauses volume replication and causes the transmission of data to cease. Be sure that you have coordinated these changes at both sites.

d. Click **Save Changes**.

2. Prepare the target volume:

a. From the cluster that contains the volume you want to assign as target, select **Management** > **Volumes**.

b. Click the Actions icon for the volume you want to assign as target and click **Edit**.

c. In the **Access** drop-down list, select **Replication Target**.

If you assign an existing volume as the replication target, the data on that volume will be overwritten. You should use a new target volume that contains no data and has the exact characteristics of the source volume, such as size, 512e setting, and QoS configuration. The target volume can be greater or equal in size to the source volume, but it cannot be smaller.

d. Click **Save Changes**.

Find more information

- Pair volumes using a volume ID
- Pair volumes using a pairing key

Validate volume replication

After a volume is replicated, you should ensure that the source and target volumes are active. When in an active state, volumes are paired, data is being sent from the source to the target volume, and the data is in sync.

1. From both clusters, select **Data Protection** > **Volume Pairs**.
2. Verify that the volume status is Active.

Find more information

Volume pairing warnings

Delete a volume relationship after replication

After replication completes and you no longer need the volume pair relationship, you can delete the volume relationship.

1. Select **Data Protection** > **Volume Pairs**.
2. Click the **Actions** icon for the volume pair you want to delete.
3. Click **Delete**.
4. Confirm the message.

Manage volume relationships

You can manage volume relationships in many ways, such as pausing replication, reversing volume pairing, changing the mode of replication, deleting a volume pair, or deleting a cluster pair.

Find more information

- Pause replication
- Change the mode of replication
- Delete volume pairs

Pause replication

You can manually pause replication if you need to stop I/O processing for a short time. You might want to pause replication if there is a surge in I/O processing and you want to reduce the processing load.

1. Select **Data Protection** > **Volume Pairs**.
2. Click the Actions icon for the volume pair.
3. Click **Edit**.
4. In the **Edit Volume Pair** pane, manually pause the replication process.

   Pausing or resuming volume replication manually causes the transmission of data to cease or resume. Be sure that you have coordinated these changes at both sites.

5. Click **Save Changes**.

Change the mode of replication

You can edit volume pair properties to change the replication mode of the volume pair relationship.

1. Select **Data Protection** > **Volume Pairs**.
2. Click the Actions icon for the volume pair.
3. Click **Edit**.
4. In the **Edit Volume Pair** pane, select a new replication mode:
   - **Real-time (Asynchronous)**: Writes are acknowledged to the client after they are committed on the source cluster.
   - **Real-time (Synchronous)**: Writes are acknowledged to the client after they are committed on both the source and target clusters.
   - **Snapshots Only**: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.
     Attention: Changing the mode of replication changes the mode immediately. Be sure that you have coordinated these changes at both sites.
5. Click **Save Changes**.

Delete volume pairs

You can delete a volume pair if want to remove a pair association between two volumes.

1. Select **Data Protection** > **Volume Pairs**.
2. Click the Actions icon for the volume pair you want to delete.
3. Click **Delete**.
4. Confirm the message.

Delete a cluster pair

You can delete a cluster pair from the Element UI of either of the clusters in the pair.

1. Click **Data Protection** > **Cluster Pairs**.
2. Click the Actions icon for a cluster pair.
3. In the resulting menu, click **Delete**.
4. Confirm the action.
5. Perform the steps again from the second cluster in the cluster pairing.

Cluster pair details

The Cluster Pairs page on the Data Protection tab provides information about clusters that have been paired or are in the process of being paired. The system displays pairing and progress messages in the Status column.

- **ID**

  A system-generated ID given to each cluster pair.

- **Remote Cluster Name**

  The name of the other cluster in the pair.

- **Remote MVIP**

  The management virtual IP address of the other cluster in the pair.

- **Status**

  Replication status of the remote cluster

- **Replicating Volumes**

  The number of volumes contained by the cluster that are paired for replication.

- **UUID**

  A unique ID given to each cluster in the pair.

Volume pair details

The Volume Pairs page on the Data Protection tab provides information about volumes that have been paired or are in the process of being paired. The system displays pairing and progress messages in the Volume Status column.

- **ID**

  System-generated ID for the volume.

- **Name**

  The name given to the volume when it was created. Volume names can be up to 223 characters and contain a-z, 0-9, and dash (-).

- **Account**

  Name of the account assigned to the volume.

- **Volume Status**

  Replication status of the volume

- **Snapshot Status**

  Status of the snapshot volume.

- **Mode**

  The client write replication method. Possible values are as follows:

  - Async
  - Snapshot-Only
  - Sync

- **Direction**

  The direction of the volume data:

  - Source volume icon (→) indicates data is being written to a target outside the cluster.
  - Target volume icon (←) indicates data is being written to the local volume from an outside source.

- **Async Delay**

  Length of time since the volume was last synced with the remote cluster. If the volume is not paired, the value is null.

- **Remote Cluster**

  Name of the remote cluster on which the volume resides.

- **Remote Volume ID**

  Volume ID of the volume on the remote cluster.

- **Remote Volume Name**

  Name given to the remote volume when it was created.

Volume pairing messages

You can view volume pairing messages during the initial pairing process from the Volume Pairs page under the Data Protection tab. These messages can display on both source and target ends of the pair in the Replicating Volumes list view.

- **PausedDisconnected**

  Source replication or sync RPCs timed out. Connection to the remote cluster has been lost. Check network connections to the cluster.

- **ResumingConnected**

  The remote replication sync is now active. Beginning the sync process and waiting for data.

- **ResumingRRSync**

  A single helix copy of the volume metadata is being made to the paired cluster.

- **ResumingLocalSync**

  A double helix copy of the volume metadata is being made to the paired cluster.

- **ResumingDataTransfer**

  Data transfer has resumed.

- **Active**

  Volumes are paired and data is being sent from the source to the target volume and the data is in sync.

- **Idle**

  No replication activity is occurring.

Volume pairing warnings

TheVolume Pairs page on the Data Protection tab provides these messages after you pair volumes. These messages can display on both source and target ends of the pair (unless otherwise indicated) in the Replicating Volumes list view.

- **PausedClusterFull**

  Because the target cluster is full, source replication and bulk data transfer cannot proceed. The message displays on the source end of the pair only.

- **PausedExceededMaxSnapshotCount**

  The target volume already has the maximum number of snapshots and cannot replicate additional

snapshots.

- **PausedManual**

  Local volume has been manually paused. It must be unpaused before replication resumes.

- **PausedManualRemote**

  Remote volume is in manual paused mode. Manual intervention required to unpause the remote volume before replication resumes.

- **PausedMisconfigured**

  Waiting for an active source and target. Manual intervention required to resume replication.

- **PausedQoS**

  Target QoS could not sustain incoming IO. Replication auto-resumes. The message displays on the source end of the pair only.

- **PausedSlowLink**

  Slow link detected and stopped replication. Replication auto-resumes. The message displays on the source end of the pair only.

- **PausedVolumeSizeMismatch**

  Target volume is not the same size as the source volume.

- **PausedXCopy**

  A SCSI XCOPY command is being issued to a source volume. The command must complete before replication can resume. The message displays on the source end of the pair only.

- **StoppedMisconfigured**

  A permanent configuration error has been detected. The remote volume has been purged or unpaired. No corrective action is possible; a new pairing must be established.

Use SnapMirror replication between Element and ONTAP clusters (Element UI)

You can create SnapMirror relationships from the Data Protection tab in the NetApp Element UI. SnapMirror functionality must be enabled to see this in the user interface.

IPv6 is not supported for SnapMirror replication between NetApp Element software and ONTAP clusters.

NetApp video: SnapMirror for NetApp HCI and Element Software

Systems running NetApp Element software support SnapMirror functionality to copy and restore Snapshot copies with NetApp ONTAP systems. The primary reason for using this technology is disaster recovery of NetApp HCI to ONTAP. Endpoints include ONTAP, ONTAP Select, and Cloud Volumes ONTAP. See TR-4641 NetApp HCI Data Protection.

NetApp Technical Report 4641: NetApp HCI Data Protection

Find more information

- Building your Data Fabric with NetApp HCI, ONTAP, and Converged Infrastructure
- Perform replication between NetApp Element software and ONTAP (ONTAP CLI)

SnapMirror overview

Systems running NetApp Element software support SnapMirror functionality to copy and restore snapshots with NetApp ONTAP systems.

Systems running Element can communicate directly with SnapMirror on ONTAP systems 9.3 or higher. The NetApp Element API provides methods to enable SnapMirror functionality on clusters, volumes, and snapshots. Additionally, the Element UI includes all necessary functionality to manage SnapMirror relationships between Element software and ONTAP systems.

You can replicate ONTAP originated volumes to Element volumes in specific use cases with limited functionality. For more information, see Replication between Element software and ONTAP (ONTAP CLI).

Enable SnapMirror on the cluster

You must manually enable SnapMirror functionality at the cluster level through the NetApp Element UI. The system comes with SnapMirror functionality disabled by default, and it is not automatically enabled as part of a new installation or upgrade. Enabling the SnapMirror feature is a one-time configuration task.

SnapMirror can only be enabled for clusters running Element software used in conjunction with volumes on a NetApp ONTAP system. You should enable SnapMirror functionality only if your cluster is connected for use with NetApp ONTAP volumes.

What you'll need

The storage cluster must be running NetApp Element software.

Steps

1. Click **Clusters** > **Settings**.
2. Find the cluster-specific settings for SnapMirror.
3. Click **Enable SnapMirror**.

> Enabling SnapMirror functionality permanently changes the Element software configuration. You can disable the SnapMirror feature and restore the default settings only by returning the cluster to the factory image.

4. Click **Yes** to confirm the SnapMirror configuration change.

Enable SnapMirror on the volume

You must enable SnapMirror on the volume in the Element UI. This allows replication of data to specified ONTAP volumes. This is permission from the administrator of the cluster running NetApp Element software for SnapMirror to control a volume.

What you'll need

- You have enabled SnapMirror in the Element UI for the cluster.
- A SnapMirror endpoint is available.
- The volume must be 512e block size.
- The volume is not participating in remote replication.
- The volume access type is not Replication Target.

You can also set this property when creating or cloning a volume.

Steps

1. Click **Management** > **Volumes**.
2. Click the **Actions** icon for the volume you want to enable SnapMirror for.
3. In the resulting menu, select **Edit**.
4. In the **Edit Volume** dialog box, select the check box **Enable SnapMirror**.
5. Click **Save Changes**.

Create a SnapMirror endpoint

You must create a SnapMirror endpoint in the NetApp Element UI before you can create a relationship.

A SnapMirror endpoint is an ONTAP cluster that serves as a replication target for a cluster running Element software. Before you create a SnapMirror relationship, you first create a SnapMirror endpoint.

You can create and manage up to four SnapMirror endpoints on a storage cluster running Element software.

If an existing endpoint was originally created using the API and credentials were not saved, you can see the endpoint in the Element UI and verify its existence, but it cannot be managed using the Element UI. This endpoint can then only be managed using the Element API.

For details about API methods, see Manage storage with the Element API.

What you'll need

- You should have enabled SnapMirror in the Element UI for the storage cluster.
- You know the ONTAP credentials for the endpoint.

Steps

1. Click **Data Protection** > **SnapMirror Endpoints**.
2. Click **Create Endpoint**.
3. In the **Create a New Endpoint** dialog box, enter the cluster management IP address of the ONTAP system.
4. Enter the ONTAP administrator credentials associated with the endpoint.
5. Review additional details:
   - LIFs: Lists the ONTAP intercluster logical interfaces used to communicate with Element.
   - Status: Shows the current status of the SnapMirror endpoint. Possible values are: connected, disconnected, and unmanaged.

6. Click **Create Endpoint**.

Create a SnapMirror relationship

You must create a SnapMirror relationship in the NetApp Element UI.

When a volume is not yet enabled for SnapMirror and you select to create a relationship from the Element UI, SnapMirror is automatically enabled on that volume.

What you'll need

SnapMirror is enabled on the volume.

Steps

1. Click **Management** > **Volumes**.
2. Click the **Actions** icon for the volume that is to be a part of the relationship.
3. Click **Create a SnapMirror Relationship**.
4. In the **Create a SnapMirror Relationship** dialog box, select an endpoint from the **Endpoint** list.
5. Select if the relationship will be created using a new ONTAP volume or an existing ONTAP volume.
6. To create a new ONTAP volume in the Element UI, click **Create new volume**.
   a. Select the **Storage Virtual Machine** for this relationship.
   b. Select the **Aggregate** from the drop-down list.
   c. In the **Volume Name Suffix** field, enter a suffix.

      The system detects the source volume name and copies it to the **Volume Name** field. The suffix you enter appends the name.

   d. Click **Create Destination Volume**.
7. To use an existing ONTAP volume, click **Use existing volume**.
   a. Select the **Storage Virtual Machine** for this relationship.
   b. Select the volume that is the destination for this new relationship.
8. In the **Relationship Details** section, select a policy. If the selected policy has keep rules, the Rules table displays the rules and associated labels.
9. **Optional**: Select a schedule.

   This determines how often the relationship creates copies.

10. **Optional**: In the **Limit Bandwidth to** field, enter the maximum amount of bandwidth that can be consumed by data transfers associated with this relationship.
11. Review additional details:
    - **State**: Current relationship state of the destination volume. Possible values are:
      - uninitialized: The destination volume has not been initialized.
      - snapmirrored: The destination volume has been initialized and is ready to receive SnapMirror updates.
      - broken-off: The destination volume is read/write and snapshots are present.

- **Status**: Current status of the relationship. Possible values are idle, transferring, checking, quiescing, quiesced, queued, preparing, finalizing, aborting, and breaking.
- **Lag Time**: The amount of time in seconds that the destination system lags behind the source system. The lag time must be no more than the transfer schedule interval.
- **Bandwidth Limit**: The maximum amount of bandwidth that can be consumed by data transfers associated with this relationship.
- **Last Transferred**: Timestamp of the last transferred snapshot. Click for further information.
- **Policy Name**: The name of the ONTAP SnapMirror policy for the relationship.
- **Policy Type**: Type of ONTAP SnapMirror policy selected for the relationship. Possible values are:
  - async_mirror
  - mirror_vault
- **Schedule Name**: Name of the pre-existing schedule on the ONTAP system selected for this relationship.

12. To not initialize at this time, ensure that the **Initialize** check box is not selected.

> Initialization can be time-consuming. You might want to run this during off-peak hours. Initialization performs a baseline transfer; it makes a snapshot copy of the source volume, then transfers that copy and all the data blocks it references to the destination volume. You can initialize manually or use a schedule to start the initialization process (and subsequent updates) according to the schedule.

13. Click **Create Relationship**.
14. Click **Data Protection** > **SnapMirror Relationships** to view this new SnapMirror relationship.

SnapMirror relationship actions

You can configure a relationship from the SnapMirror Relationships page of the Data Protection tab. The options from the Actions icon are described here.

- **Edit**: Edits the policy used or schedule for the relationship.
- **Delete**: Deletes the SnapMirror relationship. This function does not delete the destination volume.
- **Initialize**: Performs the first initial baseline transfer of data to establish a new relationship.
- **Update**: Performs an on-demand update of the relationship, replicating any new data and Snapshot copies included since the last update to the destination.
- **Quiesce**: Prevents any further updates for a relationship.
- **Resume**: Resumes a relationship that is quiesced.
- **Break**: Makes the destination volume read-write and stops all current and future transfers. Determine that clients are not using the original source volume, because the reverse resync operation makes the original source volume read-only.
- **Resync**: Reestablishes a broken relationship in the same direction before the break occurred.
- **Reverse Resync**: Automates the necessary steps to create and initialize a new relationship in the opposite direction. This can be done only if the existing relationship is in a broken state. This operation will not delete the current relationship. The original source volume reverts to the most recent common Snapshot copy and resynchronizes with the destination. Any changes that are made to the original source volume since the last successful SnapMirror update are lost. Any changes that were made to, or new data written

into the current destination volume is sent back to the original source volume.

- **Abort**: Cancels a current transfer in progress. If a SnapMirror update is issued for an aborted relationship, the relationship continues with the last transfer from the last restart checkpoint that was created before the abort occurred.

SnapMirror labels

A SnapMirror label serves as a marker for transferring a specified snapshot according to the retention rules of the relationship.

Applying a label to a snapshot marks it as a target for SnapMirror replication. The role of the relationship is to enforce the rules upon data transfer by selecting the matching labeled snapshot, copying it to the destination volume, and ensuring the correct number of copies are kept. It refers to the policy to determine the keep count and the retention period. The policy can have any number of rules and each rule has a unique label. This label serves as the link between the snapshot and the retention rule.

It is the SnapMirror label that indicates which rule is applied for the selected snapshot, group snapshot, or schedule.

Add SnapMirror labels to snapshots

SnapMirror labels specify the snapshot retention policy on the SnapMirror endpoint. You can add labels to snapshots and group snapshots.

You can view available labels from an existing SnapMirror relationship dialog box or the NetApp ONTAP System Manager.

> When you add a label to a group snapshot, any existing labels to individual snapshots are overwritten.

What you'll need

- SnapMirror is enabled on the cluster.
- The label you want to add already exists in ONTAP.

Steps

1. Click **Data Protection** > **Snapshots** or **Group Snapshots** page.
2. Click the **Actions** icon for the snapshot or group snapshot you want to add a SnapMirror label to.
3. In the **Edit Snapshot** dialog box, enter text in the **SnapMirror Label** field. The label must match a rule label in the policy applied to the SnapMirror relationship.
4. Click **Save Changes**.

Add SnapMirror labels to snapshot schedules

You can add SnapMirror labels to snapshot schedules to ensure that a SnapMirror policy is applied. You can view available labels from an existing SnapMirror relationship dialog box or the NetAppONTAP System Manager.

What you'll need

- SnapMirror must be enabled at the cluster level.

- The label you want to add already exists in ONTAP.

Steps

1. Click **Data Protection** > **Schedules**.
2. Add a SnapMirror label to a schedule in one of the following ways:

Option	Steps
Creating a new schedule	a. Select **Create Schedule**. b. Enter all other relevant details. c. Select **Create Schedule**.
Modifying existing schedule	a. Click the **Actions** icon for the schedule you want to add a label to and select **Edit**. b. In the resulting dialog box, enter text in the **SnapMirror Label** field. c. Select **Save Changes**.

Find more information

Create a snapshot schedule

Disaster recovery using SnapMirror

In the event of a problem with a volume or cluster running NetApp Element software, use the SnapMirror functionality to break the relationship and failover to the destination volume.

If the original cluster has completely failed or is non-existent, contact NetApp Support for further assistance.

Perform a failover from an Element cluster

You can perform a failover from the Element cluster to make the destination volume read/write and accessible to hosts on the destination side. Before you perform a failover from the Element cluster, you must break the SnapMirror relationship.

Use the NetApp Element UI to perform the failover. If the Element UI is not available, you can also use ONTAP System Manager or ONTAP CLI to issue the break relationship command.

What you'll need

- A SnapMirror relationship exists and has at least one valid snapshot on the destination volume.
- You have a need to failover to the destination volume due to unplanned outage or planned event at the primary site.

Steps

1. In the Element UI, click **Data Protection** > **SnapMirror Relationships**.
2. Find the relationship with the source volume that you want to failover.

3. Click the **Actions** icon.
4. Click **Break**.
5. Confirm the action.

   The volume on the destination cluster now has read-write access and can be mounted to the application hosts to resume production workloads. All SnapMirror replication is halted as a result of this action. The relationship shows a state of broken-off.

Perform a failback to Element

When the issue on the primary side has been mitigated, you must resynchronize the original source volume and fail back to NetApp Element software. The steps you perform vary depending on whether the original source volume still exists or whether you need to failback to a newly created volume.

Find more information

- Perform a failback when source volume still exists
- Perform a failback when source volume no longer exists
- SnapMirror failback scenarios

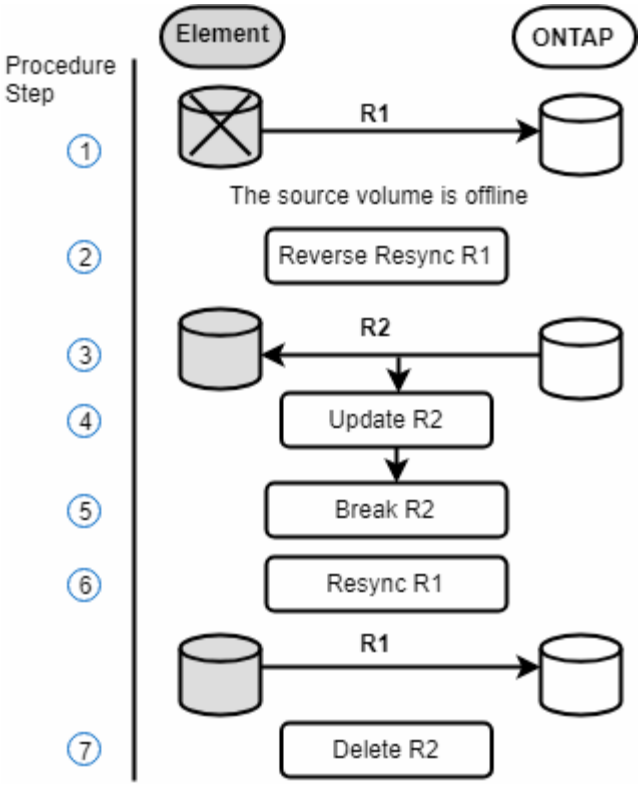
SnapMirror failback scenarios

The SnapMirror disaster recovery functionality is illustrated in two failback scenarios. These assume the original relationship has been failed over (broken).

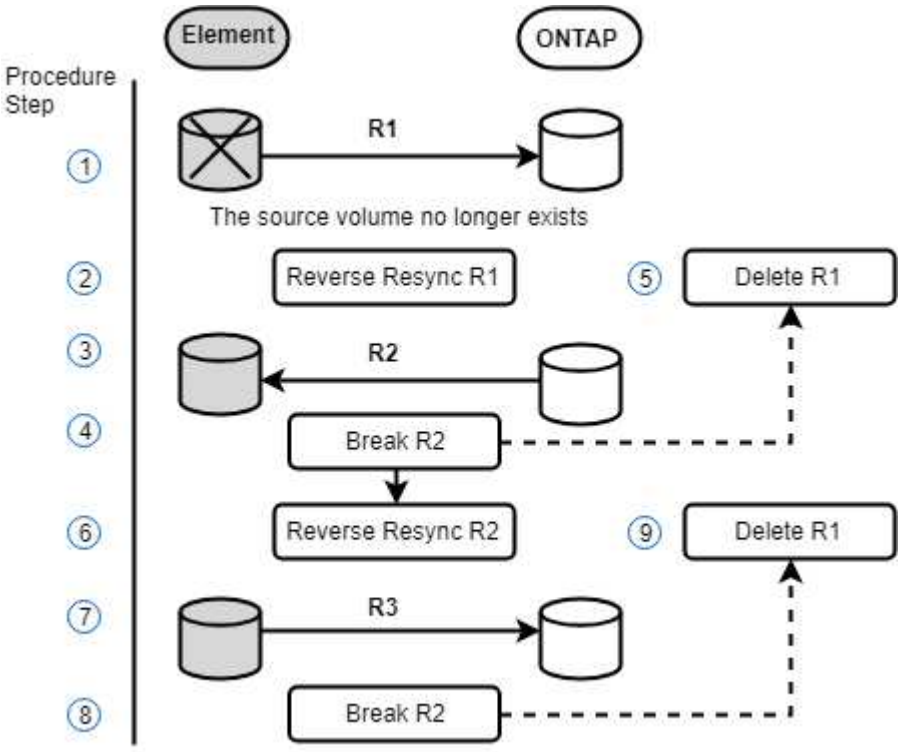
The steps from the corresponding procedures are added for reference.

> In the examples here, R1 = the original relationship in which the cluster running NetApp Element software is the original source volume (Element) and ONTAP is the original destination volume (ONTAP). R2 and R3 represent the inverse relationships created through the reverse resync operation.

The following image shows the failback scenario when the source volume still exists:



The following image shows the failback scenario when the source volume no longer exists:



Find more information

- Perform a failback when source volume still exists
- Perform a failback when source volume no longer exists

Perform a failback when source volume still exists

You can resynchronize the original source volume and fail back using the NetApp Element UI. This procedure applies to scenarios where the original source volume still exists.

1. In the Element UI, find the relationship that you broke to perform the failover.
2. Click the Actions icon and click **Reverse Resync**.
3. Confirm the action.

> The Reverse Resync operation creates a new relationship in which the roles of the original source and destination volumes are reversed (this results in two relationships as the original relationship persists). Any new data from the original destination volume is transferred to the original source volume as part of the reverse resync operation. You can continue to access and write data to the active volume on the destination side, but you will need to disconnect all hosts to the source volume and perform a SnapMirror update before redirecting back to the original primary.

4. Click the Actions icon of the inverse relationship that you just created and click **Update**.

   Now that you have completed the reverse resync and ensured that there are no active sessions connected to the volume on the destination side and that the latest data is on the original primary volume, you can perform the following steps to complete the failback and reactivate the original primary volume:

5. Click the Actions icon of the inverse relationship and click **Break**.
6. Click the Actions icon of the original relationship and click **Resync**.

> The original primary volume can now be mounted to resume production workloads on the original primary volume. The original SnapMirror replication resumes based on the policy and schedule configured for the relationship.

7. After you confirm that the original relationship status is “snapmirrored”, click the Actions icon of the inverse relationship and click **Delete**.

Find more information

SnapMirror failback scenarios

Perform a failback when source volume no longer exists

You can resynchronize the original source volume and fail back using the NetApp Element UI. This section applies to scenarios in which the original source volume has been lost but the original cluster is still intact. For instructions about how to restore to a new cluster, see the documentation on the NetApp Support Site.

What you’ll need

- You have a broken-off replication relationship between Element and ONTAP volumes.
- The Element volume is irretrievably lost.
- The original volume name shows as NOT FOUND.

Steps

1. In the Element UI, find the relationship that you broke to perform the failover.

   Best Practice: Make note of the SnapMirror policy and schedule details of the original broken-off relationship. This information will be required when recreating the relationship.

2. Click the **Actions** icon and click **Reverse Resync**.
3. Confirm the action.

   The Reverse Resync operation creates a new relationship in which the roles of the original source volume and the destination volume are reversed (this results in two relationships as the original relationship persists). Because the original volume no longer exists, the system creates a new Element volume with the same volume name and volume size as the original source volume. The new volume is assigned a default QoS policy called sm-recovery and is associated with a default account called sm-recovery. You will want to manually edit the account and QoS policy for all volumes that are created by SnapMirror to replace the original source volumes that were destroyed.

   Data from the latest snapshot is transferred to the new volume as part of the reverse resync operation. You can continue to access and write data to the active volume on the destination side, but you will need to disconnect all hosts to the active volume and perform a SnapMirror update before reinstating the original primary relationship in a later step. After you complete the reverse resync and ensure that there are no active sessions connected to the volume on the destination side and that the latest data is on the original primary volume, continue with the following steps to complete the failback and reactivate the original primary volume:

4. Click the **Actions** icon of the inverse relationship that was created during the Reverse Resync operation and click **Break**.
5. Click the **Actions** icon of the original relationship, in which the source volume does not exist, and click **Delete**.
6. Click the **Actions** icon of the inverse relationship, which you broke in step 4, and click **Reverse Resync**.
7. This reverses the source and destination and results in a relationship with the same volume source and volume destination as the original relationship.
8. Click the **Actions** icon and **Edit** to update this relationship with the original QoS policy and schedule settings you took note of.
9. Now it is safe to delete the inverse relationship that you reverse resynced in step 6.

Find more information

SnapMirror failback scenarios

Perform a transfer or one-time migration from ONTAP to Element

Typically, when you use SnapMirror for disaster recovery from a SolidFire storage cluster running NetApp Element software to ONTAP software, Element is the source and ONTAP the destination. However, in some cases the ONTAP storage system can serve as the source and Element as the destination.

- Two scenarios exist:
  - No previous disaster recovery relationship exists. Follow all the steps in this procedure.

- Previous disaster recovery relationship does exist, but not between the volumes being used for this mitigation. In this case, follow only steps 3 and 4 below.

What you'll need

- The Element destination node must have been made accessible to ONTAP.
- The Element volume must have been enabled for SnapMirror replication.

You must specify the Element destination path in the form hostip:/lun/<id_number>, where lun is the actual string "lun" and id_number is the ID of the Element volume.

Steps

1. Using ONTAP, create the relationship with the Element cluster:

```
snapmirror create -source-path SVM:volume|cluster://SVM/volume
-destination-path hostip:/lun/name -type XDP -schedule schedule -policy
    policy
```

```
cluster_dst::> snapmirror create -source-path svm_1:volA_dst
-destination-path 10.0.0.11:/lun/0005 -type XDP -schedule my_daily
-policy MirrorLatest
```

2. Verify that the SnapMirror relationship was created by using the ONTAP snapmirror show command.

   See information about creating a replication relationship in the ONTAP documentation and for complete command syntax, see the ONTAP man page.

3. Using the `ElementCreateVolume` API, create the target volume and set the target volume access mode to SnapMirror:

   Create an Element volume using the Element API

```
{
   "method": "CreateVolume",
   "params": {
        "name": "SMTargetVolumeTest2",
        "accountID": 1,
        "totalSize": 100000000000,
        "enable512e": true,
        "attributes": {},
        "qosPolicyID": 1,
        "enableSnapMirrorReplication": true,
        "access": "snapMirrorTarget"
    },
    "id": 1
}
```

4. Initialize the replication relationship using the ONTAP `snapmirror initialize` command:

```
snapmirror initialize -source-path hostip:/lun/name
-destination-path SVM:volume|cluster://SVM/volume
```

Perform replication between NetApp Element software and ONTAP (ONTAP CLI)

Perform replication between NetApp Element software and ONTAP overview (ONTAP CLI)

You can ensure business continuity on an Element system by using SnapMirror to replicate snapshot copies of an Element volume to an ONTAP destination. In the event of a disaster at the Element site, you can serve data to clients from the ONTAP system, and then reactivate the Element system when service is restored.

Beginning with ONTAP 9.4, you can replicate snapshot copies of a LUN created on an ONTAP node back to an Element system. You might have created a LUN during an outage at the Element site, or you might be using a LUN to migrate data from ONTAP to Element software.

You should work with Element to ONTAP backup if the following apply:

- You want to use best practices, not explore every available option.
- You want to use the ONTAP command-line interface (CLI), not System Manager or an automated scripting tool.
- You are using iSCSI to serve data to clients.

If you require additional SnapMirror configuration or conceptual information, see Data protection overview.

About replication between Element and ONTAP

Beginning with ONTAP 9.3, you can use SnapMirror to replicate snapshot copies of an Element volume to an ONTAP destination. In the event of a disaster at the Element site, you can serve data to clients from the ONTAP system, then reactivate the Element source volume when service is restored.

Beginning with ONTAP 9.4, you can replicate snapshot copies of a LUN created on an ONTAP node back to an Element system. You might have created a LUN during an outage at the Element site, or you might be using a LUN to migrate data from ONTAP to Element software.

Types of data protection relationship

SnapMirror offers two types of data protection relationship. For each type, SnapMirror creates a snapshot copy of the Element source volume before initializing or updating the relationship:

- In a *disaster recovery (DR)* data protection relationship, the destination volume contains only the snapshot copy created by SnapMirror, from which you can continue to serve data in the event of a catastrophe at the primary site.
- In a *long-term retention* data protection relationship, the destination volume contains point-in-time snapshot copies created by Element software, as well as the snapshot copy created by SnapMirror. You might want to retain monthly snapshot copies created over a 20-year span, for example.

Default policies

The first time you invoke SnapMirror, it performs a *baseline transfer* from the source volume to the destination volume. The *SnapMirror policy* defines the contents of the baseline and any updates.

You can use a default or custom policy when you create a data protection relationship. The *policy type* determines which snapshot copies to include and how many copies to retain.

The table below shows the default policies. Use the `MirrorLatest` policy to create a traditional DR relationship. Use the `MirrorAndVault` or `Unified7year` policy to create a unified replication relationship, in which DR and long-term retention are configured on the same destination volume.

Policy	Policy Type	Update behavior
MirrorLatest	async-mirror	Transfer the snapshot copy created by SnapMirror.
MirrorAndVault	mirror-vault	Transfer the snapshot copy created by SnapMirror and any less recent snapshot copies made since the last update, provided they have SnapMirror labels “daily” or “weekly”.
Unified7year	mirror-vault	Transfer the snapshot copy created by SnapMirror and any less recent snapshot copies made since the last update, provided they have SnapMirror labels “daily”, “weekly”, or “monthly”.

For complete background information on SnapMirror policies, including guidance on which policy to use, see Data protection overview.

Understanding SnapMirror labels

Every policy with the “mirror-vault” policy type must have a rule that specifies which snapshot copies to replicate. The rule “daily”, for example, indicates that only snapshot copies assigned the SnapMirror label “daily” should be replicated. You assign the SnapMirror label when you configure Element snapshot copies.

Replication from an Element source cluster to an ONTAP destination cluster

You can use SnapMirror to replicate snapshot copies of an Element volume to an ONTAP destination system. In the event of a disaster at the Element site, you can serve data to clients from the ONTAP system, then reactivate the Element source volume when service is restored.

An Element volume is roughly equivalent to an ONTAP LUN. SnapMirror creates a LUN with the name of the Element volume when a data protection relationship between Element software and ONTAP is initialized. SnapMirror replicates data to an existing LUN if the LUN meets the requirements for Element to ONTAP replication.

Replication rules are as follows:

- An ONTAP volume can contain data from one Element volume only.
- You cannot replicate data from an ONTAP volume to multiple Element volumes.

Replication from an ONTAP source cluster to an Element destination cluster

Beginning with ONTAP 9.4, you can replicate snapshot copies of a LUN created on an ONTAP system back to an Element volume:

- If a SnapMirror relationship already exists between an Element source and an ONTAP destination, a LUN created while you are serving data from the destination is automatically replicated when the source is reactivated.
- Otherwise, you must create and initialize a SnapMirror relationship between the ONTAP source cluster and the Element destination cluster.

Replication rules are as follows:

- The replication relationship must have a policy of type “async-mirror”.

  Policies of type “mirror-vault” are not supported.

- Only iSCSI LUNs are supported.
- You cannot replicate more than one LUN from an ONTAP volume to an Element volume.
- You cannot replicate a LUN from an ONTAP volume to multiple Element volumes.

Prerequisites

You must have completed the following tasks before configuring a data protection relationship between Element and ONTAP:

- The Element cluster must be running NetApp Element software version 10.1 or later.
- The ONTAP cluster must be running ONTAP 9.3 or later.
- SnapMirror must have been licensed on the ONTAP cluster.
- You must have configured volumes on the Element and ONTAP clusters that are large enough to handle anticipated data transfers.
- If you are using the “mirror-vault” policy type, a SnapMirror label must have been configured for the Element snapshot copies to be replicated.

  You can only perform this task in the Element software web UI or using the API methods.

- You must have ensured that port 5010 is available.
- If you foresee that you might need to move a destination volume, you must have ensured that full-mesh connectivity exists between the source and destination. Every node on the Element source cluster must be able to communicate with every node on the ONTAP destination cluster.

Support details

The following table shows support details for Element to ONTAP backup.

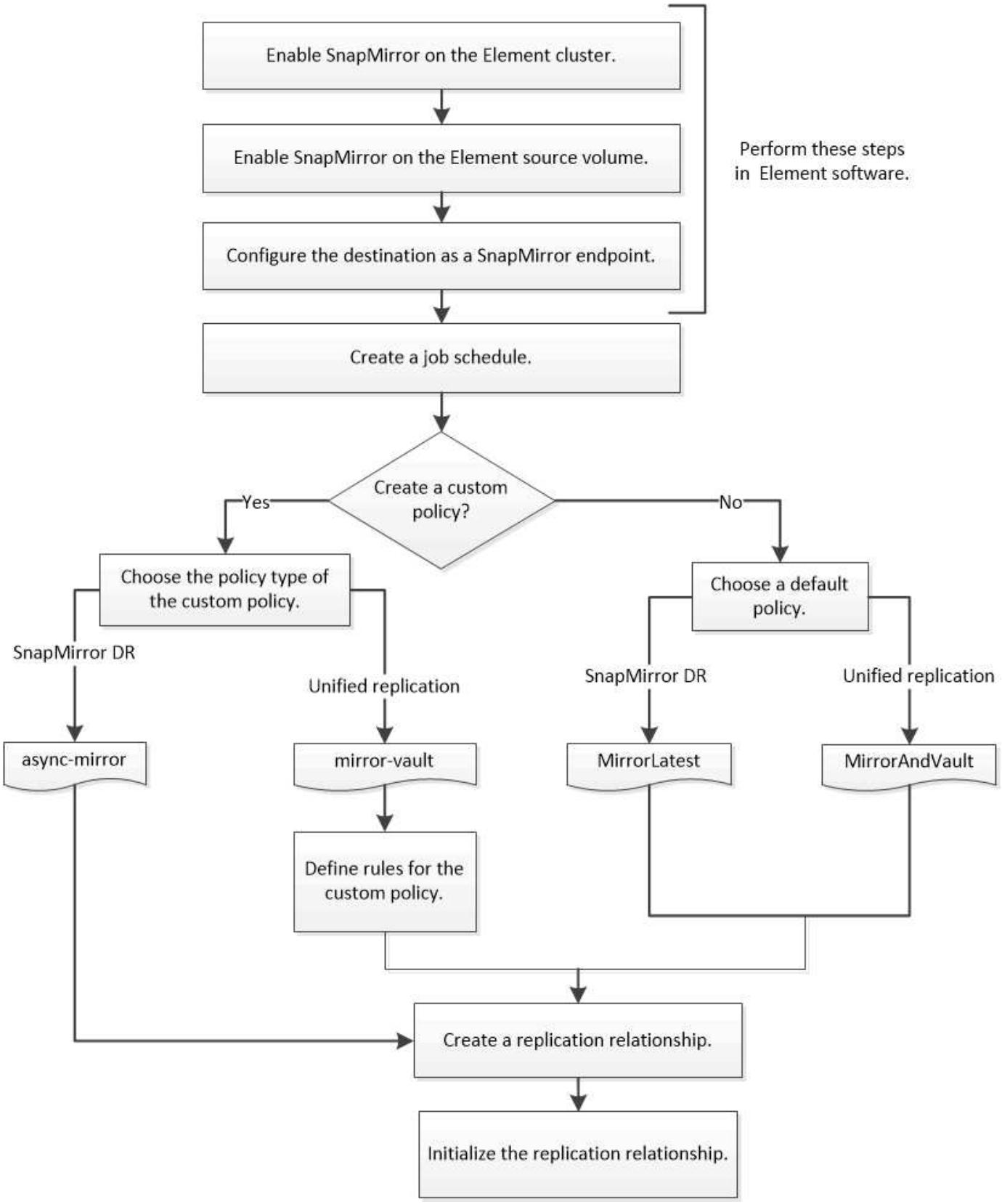
Resource or feature	Support details

SnapMirror	• The SnapMirror restore feature is not supported. • The `MirrorAllSnapshots` and `XDPDefault` policies are not supported. • The “vault” policy type is not supported. • The system-defined rule “all_source_snapshots” is not supported. • The “mirror-vault” policy type is supported only for replication from Element software to ONTAP. Use “async-mirror” for replication from ONTAP to Element software. • The `-schedule` and `-prefix` options for `snapmirror policy add-rule` are not supported. • The `-preserve` and `-quick-resync` options for `snapmirror resync` are not supported. • Storage efficiency is not preserved. • Fan-out and cascade data protection deployments are not supported.
ONTAP	• ONTAP Select is supported beginning with ONTAP 9.4 and Element 10.3. • Cloud Volumes ONTAP is supported beginning with ONTAP 9.5 and Element 11.0.
Element	• Volume size limit is 8 TiB. • Volume block size must be 512 bytes. A 4K byte block size is not supported. • Volume size must be a multiple of 1 MiB. • Volume attributes are not preserved. • Maximum number of snapshot copies to be replicated is 30.
Network	• A single TCP connection is allowed per transfer. • The Element node must be specified as an IP address. DNS hostname lookup is not supported. • IPspaces are not supported.
SnapLock	SnapLock volumes are not supported.
FlexGroup	FlexGroup volumes are not supported.
SVM DR	ONTAP volumes in an SVM DR configuration are not supported.
MetroCluster	ONTAP volumes in a MetroCluster configuration are not supported.

Workflow for replication between Element and ONTAP

Whether you are replicating data from Element to ONTAP or from ONTAP to Element, you need to configure a job schedule, specify a policy, and create and initialize the relationship. You can use a default or custom policy.

The workflow assumes that you have completed the prerequisite tasks listed in Prerequisites. For complete background information on SnapMirror policies, including guidance on which policy to use, see Data protection overview.



Enable SnapMirror in Element software

Enable SnapMirror on the Element cluster

You must enable SnapMirror on the Element cluster before you can create a replication relationship. You can only perform this task in the Element software web UI or using the API method.

Before you begin

- The Element cluster must be running NetApp Element software version 10.1 or later.
- SnapMirror can only be enabled for Element clusters used with NetApp ONTAP volumes.

About this task

The Element system comes with SnapMirror disabled by default. SnapMirror is not automatically enabled as part of a new installation or upgrade.

> Once enabled, SnapMirror cannot be disabled. You can only disable the SnapMirror feature and restore the default settings by returning the cluster to the factory image.

Steps

1. Click **Clusters** > **Settings**.
2. Find the cluster-specific settings for SnapMirror.
3. Click **Enable SnapMirror**.

Enable SnapMirror on the Element source volume

You must enable SnapMirror on the Element source volume before you can create a replication relationship. You can only perform this task in the Element software web UI or using the ModifyVolume and ModifyVolumes API methods.

Before you begin

- You must have enabled SnapMirror on the Element cluster.
- The volume block size must be 512 bytes.
- The volume must not be participating in Element remote replication.
- The volume access type must not be “Replication Target”.

About this task

The procedure below assumes the volume already exists. You can also enable SnapMirror when you create or clone a volume.

Steps

1. Select **Management** > **Volumes**.
2. Select the ⚙ button for the volume.
3. In the drop-down menu, select **Edit**.
4. In the **Edit Volume** dialog, select **Enable SnapMirror**.
5. Select **Save Changes**.

Create a SnapMirror endpoint

You must create a SnapMirror endpoint before you can create a replication relationship. You can only perform this task in the Element software web UI or using the SnapMirror API methods.

Before you begin

You must have enabled SnapMirror on the Element cluster.

Steps

1. Click **Data Protection** > **SnapMirror Endpoints**.
2. Click **Create Endpoint**.
3. In the **Create a New Endpoint** dialog, enter the ONTAP cluster management IP address.
4. Enter the user ID and password of the ONTAP cluster administrator.
5. Click **Create Endpoint**.

Configure a replication relationship

Create a replication job schedule

Whether you are replicating data from Element to ONTAP or from ONTAP to Element, you need to configure a job schedule, specify a policy, and create and initialize the relationship. You can use a default or custom policy.

You can use the `job schedule cron create` command to create a replication job schedule. The job schedule determines when SnapMirror automatically updates the data protection relationship to which the schedule is assigned.

About this task

You assign a job schedule when you create a data protection relationship. If you do not assign a job schedule, you must update the relationship manually.

Step

1. Create a job schedule:

   ```
   job schedule cron create -name job_name -month month -dayofweek day_of_week
   -day day_of_month -hour hour -minute minute
   ```

   For `-month`, `-dayofweek`, and `-hour`, you can specify `all` to run the job every month, day of the week, and hour, respectively.

   Beginning with ONTAP 9.10.1, you can include the Vserver for your job schedule:

   ```
   job schedule cron create -name job_name -vserver Vserver_name -month month
   -dayofweek day_of_week -day day_of_month -hour hour -minute minute
   ```

   The following example creates a job schedule named `my_weekly` that runs on Saturdays at 3:00 a.m.:

```
cluster_dst::> job schedule cron create -name my_weekly -dayofweek
"Saturday" -hour 3 -minute 0
```

Customize a replication policy

Create a custom replication policy

You can use a default or custom policy when you create a replication relationship. For a custom unified replication policy, you must define one or more *rules* that determine which snapshot copies are transferred during initialization and update.

You can create a custom replication policy if the default policy for a relationship is not suitable. You might want to compress data in a network transfer, for example, or modify the number of attempts SnapMirror makes to transfer snapshot copies.

About this task

The *policy type* of the replication policy determines the type of relationship it supports. The table below shows the available policy types.

Policy type	Relationship type
async-mirror	SnapMirror DR
mirror-vault	Unified replication

Step

1. Create a custom replication policy:

   ```
   snapmirror policy create -vserver SVM -policy policy -type async-
   mirror|mirror-vault -comment comment -tries transfer_tries -transfer-priority
   low|normal -is-network-compression-enabled true|false
   ```

   For complete command syntax, see the man page.

   Beginning with ONTAP 9.5, you can specify the schedule for creating a common snapshot copy schedule for SnapMirror Synchronous relationships by using the `-common-snapshot-schedule` parameter. By default, the common snapshot copy schedule for SnapMirror Synchronous relationships is one hour. You can specify a value from 30 minutes to two hours for the snapshot copy schedule for SnapMirror Synchronous relationships.

   The following example creates a custom replication policy for SnapMirror DR that enables network compression for data transfers:

   ```
   cluster_dst::> snapmirror policy create -vserver svm1 -policy
   DR_compressed -type async-mirror -comment “DR with network compression
   enabled” -is-network-compression-enabled true
   ```

The following example creates a custom replication policy for unified replication:

```
cluster_dst::> snapmirror policy create -vserver svm1 -policy my_unified
-type mirror-vault
```

After you finish

For “mirror-vault” policy types, you must define rules that determine which snapshot copies are transferred during initialization and update.

Use the `snapmirror policy show` command to verify that the SnapMirror policy was created. For complete command syntax, see the man page.

Define a rule for a policy

For custom policies with the “mirror-vault” policy type, you must define at least one rule that determines which snapshot copies are transferred during initialization and update. You can also define rules for default policies with the “mirror-vault” policy type.

About this task

Every policy with the “mirror-vault” policy type must have a rule that specifies which snapshot copies to replicate. The rule “bi-monthly”, for example, indicates that only snapshot copies assigned the SnapMirror label “bi-monthly” should be replicated. You assign the SnapMirror label when you configure Element snapshot copies.

Each policy type is associated with one or more system-defined rules. These rules are automatically assigned to a policy when you specify its policy type. The table below shows the system-defined rules.

System-defined rule	Used in policy types	Result
sm_created	async-mirror, mirror-vault	A snapshot copy created by SnapMirror is transferred on initialization and update.
daily	mirror-vault	New snapshot copies on the source with the SnapMirror label “daily” are transferred on initialization and update.
weekly	mirror-vault	New snapshot copies on the source with the SnapMirror label “weekly” are transferred on initialization and update.
monthly	mirror-vault	New snapshot copies on the source with the SnapMirror label “monthly” are transferred on initialization and update.

You can specify additional rules as needed, for default or custom policies. For example:

- For the default `MirrorAndVault` policy, you might create a rule called “bi-monthly” to match snapshot copies on the source with the “bi-monthly” SnapMirror label.
- For a custom policy with the “mirror-vault” policy type, you might create a rule called “bi-weekly” to match snapshot copies on the source with the “bi-weekly” SnapMirror label.

Step

1. Define a rule for a policy:

   `snapmirror policy add-rule -vserver` *`SVM`* `-policy` *`policy_for_rule`* `-snapmirror -label` *`snapmirror-label`* `-keep` *`retention_count`*

   For complete command syntax, see the man page.

   The following example adds a rule with the SnapMirror label `bi-monthly` to the default `MirrorAndVault` policy:

   ```
   cluster_dst::> snapmirror policy add-rule -vserver svm1 -policy
   MirrorAndVault -snapmirror-label bi-monthly -keep 6
   ```

   The following example adds a rule with the SnapMirror label `bi-weekly` to the custom `my_snapvault` policy:

   ```
   cluster_dst::> snapmirror policy add-rule -vserver svm1 -policy
   my_snapvault -snapmirror-label bi-weekly -keep 26
   ```

   The following example adds a rule with the SnapMirror label `app_consistent` to the custom `Sync` policy:

   ```
   cluster_dst::> snapmirror policy add-rule -vserver svm1 -policy Sync
   -snapmirror-label app_consistent -keep 1
   ```

   You can then replicate snapshot copies from the source cluster that match this SnapMirror label:

   ```
   cluster_src::> snapshot create -vserver vs1 -volume vol1 -snapshot
   snapshot1 -snapmirror-label app_consistent
   ```

Create a replication relationship

Create a relationship from an Element source to an ONTAP destination

The relationship between the source volume in primary storage and the destination volume in secondary storage is called a *data protection relationship*. You can use the `snapmirror create` command to create a data protection relationship from an Element source to an ONTAP destination, or from an ONTAP source to an Element destination.

You can use SnapMirror to replicate snapshot copies of an Element volume to an ONTAP destination system. In the event of a disaster at the Element site, you can serve data to clients from the ONTAP system, then reactivate the Element source volume when service is restored.

Before you begin

- The Element node containing the volume to be replicated must have been made accessible to ONTAP.
- The Element volume must have been enabled for SnapMirror replication.
- If you are using the “mirror-vault” policy type, a SnapMirror label must have been configured for the Element snapshot copies to be replicated.

You can only perform this task in the Element software web UI or using the API methods.

About this task

You must specify the Element source path in the form `<hostip:>/lun/<name>`, where “lun” is the actual string “lun” and `name` is the name of the Element volume.

An Element volume is roughly equivalent to an ONTAP LUN. SnapMirror creates a LUN with the name of the Element volume when a data protection relationship between Element software and ONTAP is initialized. SnapMirror replicates data to an existing LUN if the LUN meets the requirements for replicating from Element software to ONTAP.

Replication rules are as follows:

- An ONTAP volume can contain data from one Element volume only.
- You cannot replicate data from an ONTAP volume to multiple Element volumes.

In ONTAP 9.3 and earlier, a destination volume can contain up to 251 snapshot copies. In ONTAP 9.4 and later, a destination volume can contain up to 1019 snapshot copies.

Step

1. From the destination cluster, create a replication relationship from an Element source to an ONTAP destination:

   ```
   snapmirror create -source-path <hostip:>/lun/<name> -destination-path
   <SVM:volume>|<cluster://SVM/volume> -type XDP -schedule schedule -policy
   <policy>
   ```

   For complete command syntax, see the man page.

   The following example creates a SnapMirror DR relationship using the default `MirrorLatest` policy:

   ```
   cluster_dst::> snapmirror create -source-path 10.0.0.11:/lun/0005
   -destination-path svm_backup:volA_dst -type XDP -schedule my_daily
   -policy MirrorLatest
   ```

   The following example creates a unified replication relationship using the default `MirrorAndVault` policy:

```
cluster_dst:> snapmirror create -source-path 10.0.0.11:/lun/0005
-destination-path svm_backup:volA_dst -type XDP -schedule my_daily
-policy MirrorAndVault
```

The following example creates a unified replication relationship using the `Unified7year` policy:

```
cluster_dst::> snapmirror create -source-path 10.0.0.11:/lun/0005
-destination-path svm_backup:volA_dst -type XDP -schedule my_daily
-policy Unified7year
```

The following example creates a unified replication relationship using the custom `my_unified` policy:

```
cluster_dst::> snapmirror create -source-path 10.0.0.11:/lun/0005
-destination-path svm_backup:volA_dst -type XDP -schedule my_daily
-policy my_unified
```

After you finish

Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.

Create a relationship from an ONTAP source to an Element destination

Beginning with ONTAP 9.4, you can use SnapMirror to replicate snapshot copies of a LUN created on an ONTAP source back to an Element destination. You might be using the LUN to migrate data from ONTAP to Element software.

Before you begin

- The Element destination node must have been made accessible to ONTAP.
- The Element volume must have been enabled for SnapMirror replication.

About this task

You must specify the Element destination path in the form `<hostip:>/lun/<name>`, where “lun” is the actual string “lun” and `name` is the name of the Element volume.

Replication rules are as follows:

- The replication relationship must have a policy of type “async-mirror”.

  You can use a default or custom policy.

- Only iSCSI LUNs are supported.
- You cannot replicate more than one LUN from an ONTAP volume to an Element volume.
- You cannot replicate a LUN from an ONTAP volume to multiple Element volumes.

Step

1. Create a replication relationship from an ONTAP source to an Element destination:

   ```
   snapmirror create -source-path <SVM:volume>|<cluster://SVM/volume>
   -destination-path <hostip:>/lun/<name> -type XDP -schedule schedule -policy
   <policy>
   ```

   For complete command syntax, see the man page.

   The following example creates a SnapMirror DR relationship using the default `MirrorLatest` policy:

   ```
   cluster_dst::> snapmirror create -source-path svm_1:volA_dst
   -destination-path 10.0.0.11:/lun/0005 -type XDP -schedule my_daily
   -policy MirrorLatest
   ```

   The following example creates a SnapMirror DR relationship using the custom `my_mirror` policy:

   ```
   cluster_dst::> snapmirror create -source-path svm_1:volA_dst
   -destination-path 10.0.0.11:/lun/0005 -type XDP -schedule my_daily
   -policy my_mirror
   ```

After you finish

Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.

Initialize a replication relationship

For all relationship types, initialization performs a *baseline transfer*: it makes a snapshot copy of the source volume, then transfers that copy and all the data blocks it references to the destination volume.

Before you begin

- The Element node containing the volume to be replicated must have been made accessible to ONTAP.
- The Element volume must have been enabled for SnapMirror replication.
- If you are using the “mirror-vault” policy type, a SnapMirror label must have been configured for the Element snapshot copies to be replicated.

> You can only perform this task in the Element software web UI or using the API methods.

About this task

You must specify the Element source path in the form `<hostip:>/lun/<name>`, where “lun” is the actual string “lun” and *`name`* is the name of the Element volume.

Initialization can be time-consuming. You might want to run the baseline transfer in off-peak hours.

If initialization of a relationship from an ONTAP source to an Element destination fails for any reason, it will continue to fail even after you have corrected the problem (an invalid LUN name, for example). The workaround is as follows:

1. Delete the relationship.
2. Delete the Element destination volume.
3. Create a new Element destination volume.
4. Create and initialize a new relationship from the ONTAP source to the Element destination volume.

Step

1. Initialize a replication relationship:

   `snapmirror initialize -source-path <hostip:>/lun/<name> -destination-path <SVM:volume|cluster://SVM/volume>`

   For complete command syntax, see the man page.

   The following example initializes the relationship between the source volume `0005` at IP address 10.0.0.11 and the destination volume `volA_dst` on `svm_backup`:

   ```
   cluster_dst::> snapmirror initialize -source-path 10.0.0.11:/lun/0005
   -destination-path svm_backup:volA_dst
   ```

Serve data from a SnapMirror DR destination volume

Make the destination volume writeable

When disaster disables the primary site for a SnapMirror DR relationship, you can serve data from the destination volume with minimal disruption. You can reactivate the source volume when service is restored at the primary site.

You need to make the destination volume writeable before you can serve data from the volume to clients. You can use the `snapmirror quiesce` command to stop scheduled transfers to the destination, the `snapmirror abort` command to stop ongoing transfers, and the `snapmirror break` command to make the destination writeable.

About this task

You must specify the Element source path in the form `<hostip:>/lun/<name>`, where “lun” is the actual string “lun” and `name` is the name of the Element volume.

Steps

1. Stop scheduled transfers to the destination:

   `snapmirror quiesce -source-path <hostip:>/lun/<name> -destination-path <SVM:volume>|<cluster://SVM/volume>`

   For complete command syntax, see the man page.

The following example stops scheduled transfers between the source volume `0005` at IP address 10.0.0.11 and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst::> snapmirror quiesce -source-path 10.0.0.11:/lun/0005
-destination-path svm_backup:volA_dst
```

2. Stop ongoing transfers to the destination:

   `snapmirror abort -source-path <hostip:>/lun/<name> -destination-path <SVM:volume>|<cluster://SVM/volume>`

   For complete command syntax, see the man page.

   The following example stops ongoing transfers between the source volume `0005` at IP address 10.0.0.11 and the destination volume `volA_dst` on `svm_backup`:

   ```
   cluster_dst::> snapmirror abort -source-path 10.0.0.11:/lun/0005
   -destination-path svm_backup:volA_dst
   ```

3. Break the SnapMirror DR relationship:

   `snapmirror break -source-path <hostip:>/lun/<name> -destination-path <SVM:volume>|<cluster://SVM/volume>`

   For complete command syntax, see the man page.

   The following example breaks the relationship between the source volume `0005` at IP address 10.0.0.11 and the destination volume `volA_dst` on `svm_backup` and the destination volume `volA_dst` on `svm_backup`:

   ```
   cluster_dst::> snapmirror break -source-path 10.0.0.11:/lun/0005
   -destination-path svm_backup:volA_dst
   ```

Configure the destination volume for data access

After making the destination volume writeable, you must configure the volume for data access. SAN hosts can access the data from the destination volume until the source volume is reactivated.

1. Map the Element LUN to the appropriate initiator group.
2. Create iSCSI sessions from the SAN host initiators to the SAN LIFs.
3. On the SAN client, perform a storage re-scan to detect the connected LUN.

Reactivate the original source volume

You can reestablish the original data protection relationship between the source and

destination volumes when you no longer need to serve data from the destination.

About this task

The procedure below assumes that the baseline in the original source volume is intact. If the baseline is not intact, you must create and initialize the relationship between the volume you are serving data from and the original source volume before performing the procedure.

You must specify the Element source path in the form `<hostip:>/lun/<name>`, where “lun” is the actual string “lun” and `name` is the name of the Element volume.

Beginning with ONTAP 9.4, snapshot copies of a LUN created while you are serving data from the ONTAP destination are automatically replicated when the Element source is reactivated.

Replication rules are as follows:

- Only iSCSI LUNs are supported.
- You cannot replicate more than one LUN from an ONTAP volume to an Element volume.
- You cannot replicate a LUN from an ONTAP volume to multiple Element volumes.

Steps

1. Delete the original data protection relationship:

   ```
   snapmirror delete -source-path <SVM:volume>|<cluster://SVM/volume>
   -destination-path <hostip:>/lun/<name> -policy <policy>
   ```

   For complete command syntax, see the man page.

   The following example deletes the relationship between the original source volume, `0005` at IP address 10.0.0.11, and the volume you are serving data from, `volA_dst` on `svm_backup`:

   ```
   cluster_dst::> snapmirror delete -source-path 10.0.0.11:/lun/0005
   -policy MirrorLatest -destination-path svm_backup:volA_dst
   ```

2. Reverse the original data protection relationship:

   ```
   snapmirror resync -source-path <SVM:volume>|<cluster://SVM/volume>
   -destination-path <hostip:>/lun/<name> -policy <policy>
   ```

   For complete command syntax, see the man page.

   Although resync does not require a baseline transfer, it can be time-consuming. You might want to run the resync in off-peak hours.

   The following example reverses the relationship between the original source volume, `0005` at IP address 10.0.0.11, and the volume you are serving data from, `volA_dst` on `svm_backup`:

   ```
   cluster_dst::> snapmirror resync -source-path svm_backup:volA_dst
   -destination-path 10.0.0.11:/lun/0005 -policy MirrorLatest
   ```

3. Update the reversed relationship:

   ```
   snapmirror update -source-path <SVM:volume>|<cluster://SVM/volume>
   -destination-path <hostip:>/lun/<name>
   ```

   For complete command syntax, see the man page.

   > The command fails if a common snapshot copy does not exist on the source and destination. Use `snapmirror initialize` to re-initialize the relationship.

   The following example updates the relationship between the volume you are serving data from, `volA_dst` on `svm_backup`, and the original source volume, `0005` at IP address 10.0.0.11:

   ```
   cluster_dst::> snapmirror update -source-path svm_backup:volA_dst
   -destination-path 10.0.0.11:/lun/0005
   ```

4. Stop scheduled transfers for the reversed relationship:

   ```
   snapmirror quiesce -source-path <SVM:volume>|<cluster://SVM/volume>
   -destination-path <hostip:>/lun/<name>
   ```

   For complete command syntax, see the man page.

   The following example stops scheduled transfers between the volume you are serving data from, `volA_dst` on `svm_backup`, and the original source volume, `0005` at IP address 10.0.0.11:

   ```
   cluster_dst::> snapmirror quiesce -source-path svm_backup:volA_dst
   -destination-path 10.0.0.11:/lun/0005
   ```

5. Stop ongoing transfers for the reversed relationship:

   ```
   snapmirror abort -source-path <SVM:volume>|<cluster://SVM/volume> -destination
   -path <hostip:>/lun/<name>
   ```

   For complete command syntax, see the man page.

   The following example stops ongoing transfers between the volume you are serving data from, `volA_dst` on `svm_backup`, and the original source volume, `0005` at IP address 10.0.0.11:

   ```
   cluster_dst::> snapmirror abort -source-path svm_backup:volA_dst
   -destination-path 10.0.0.11:/lun/0005
   ```

6. Break the reversed relationship:

   ```
   snapmirror break -source-path <SVM:volume>|<cluster://SVM/volume> -destination
   -path <hostip:>/lun/<name>
   ```

For complete command syntax, see the man page.

The following example breaks the relationship between the volume you are serving data from, `volA_dst` on `svm_backup`, and the original source volume, `0005` at IP address 10.0.0.11:

```
cluster_dst::> snapmirror break -source-path svm_backup:volA_dst
-destination-path 10.0.0.11:/lun/0005
```

7. Delete the reversed data protection relationship:

   ```
   snapmirror delete -source-path <SVM:volume>|<cluster://SVM/volume>
   -destination-path <hostip:>/lun/<name> -policy <policy>
   ```

   For complete command syntax, see the man page.

   The following example deletes the reversed relationship between the original source volume, `0005` at IP address 10.0.0.11, and the volume you are serving data from, `volA_dst` on `svm_backup`:

   ```
   cluster_src::> snapmirror delete -source-path svm_backup:volA_dst
   -destination-path 10.0.0.11:/lun/0005 -policy MirrorLatest
   ```

8. Reestablish the original data protection relationship:

   ```
   snapmirror resync -source-path <hostip:>/lun/<name> -destination-path
   <SVM:volume>|<cluster://SVM/volume>
   ```

   For complete command syntax, see the man page.

   The following example reestablishes the relationship between the original source volume, `0005` at IP address 10.0.0.11, and the original destination volume, `volA_dst` on `svm_backup`:

   ```
   cluster_dst::> snapmirror resync -source-path 10.0.0.11:/lun/0005
   -destination-path svm_backup:volA_dst
   ```

After you finish

Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.

Update a replication relationship manually

You might need to update a replication relationship manually if an update fails because of a network error.

About this task

You must specify the Element source path in the form `<hostip:>/lun/<name>`, where “lun” is the actual string “lun” and `name` is the name of the Element volume.

Steps

1. Update a replication relationship manually:

   ```
   snapmirror update -source-path <hostip:>/lun/<name> -destination-path
   <SVM:volume>|<cluster://SVM/volume>
   ```

   For complete command syntax, see the man page.

   > The command fails if a common snapshot copy does not exist on the source and destination. Use `snapmirror initialize` to re-initialize the relationship.

   The following example updates the relationship between the source volume `0005` at IP address 10.0.0.11 and the destination volume `volA_dst` on `svm_backup`:

   ```
   cluster_src::> snapmirror update -source-path 10.0.0.11:/lun/0005
   -destination-path svm_backup:volA_dst
   ```

Resynchronize a replication relationship

You need to resynchronize a replication relationship after you make a destination volume writeable, after an update fails because a common Snapshot copy does not exist on the source and destination volumes, or if you want to change the replication policy for the relationship.

About this task

Although resync does not require a baseline transfer, it can be time-consuming. You might want to run the resync in off-peak hours.

You must specify the Element source path in the form `<hostip:>/lun/<name>`, where “lun” is the actual string “lun” and `name` is the name of the Element volume.

Step

1. Resync the source and destination volumes:

   ```
   snapmirror resync -source-path <hostip:>/lun/<name> -destination-path
   <SVM:volume>|<cluster://SVM/volume> -type XDP -policy <policy>
   ```

   For complete command syntax, see the man page.

   The following example resyncs the relationship between the source volume `0005` at IP address 10.0.0.11 and the destination volume `volA_dst` on `svm_backup`:

   ```
   cluster_dst::> snapmirror resync -source-path 10.0.0.11:/lun/0005
   -policy MirrorLatest -destination-path svm_backup:volA_dst
   ```

Back up and restore volumes

You can back up and restore volumes to other SolidFire storage, as well as secondary object stores that are compatible with Amazon S3 or OpenStack Swift.

When you restore volumes from OpenStack Swift or Amazon S3, you need manifest information from the original backup process. If you are restoring a volume that was backed up on a SolidFire storage system, no manifest information is required.

Find more information

- Back up a volume to an Amazon S3 object store
- Back up a volume to an OpenStack Swift object store
- Back up a volume to a SolidFire storage cluster
- Restore a volume from backup on an Amazon S3 object store
- Restore a volume from backup on an OpenStack Swift object store
- Restore a volume from backup on a SolidFire storage cluster

Back up a volume to an Amazon S3 object store

You can back up volumes to external object stores that are compatible with Amazon S3.

1. Click **Management** > **Volumes**.
2. Click the Actions icon for the volume you want to back up.
3. In the resulting menu, click **Backup to**.
4. In the **Integrated Backup** dialog box under **Backup to**, select **S3**.
5. Select an option under **Data Format**:
   - **Native**: A compressed format readable only by SolidFire storage systems.
   - **Uncompressed**: An uncompressed format compatible with other systems.
6. Enter a hostname to use to access the object store in the **Hostname** field.
7. Enter an access key ID for the account in the **Access Key ID** field.
8. Enter the secret access key for the account in the **Secret Access Key** field.
9. Enter the S3 bucket in which to store the backup in the **S3 Bucket** field.
10. Enter a nametag to append to the prefix in the **Nametag** field.
11. Click **Start Read**.

Back up a volume to an OpenStack Swift object store

You can back up volumes to external object stores that are compatible with OpenStack Swift.

1. Click **Management** > **Volumes**.
2. Click the Actions icon for the volume to back up.
3. In the resulting menu, click **Backup to**.

4. In the **Integrated Backup** dialog box under **Backup to**, select **Swift**.
5. Select a data format under **Data Format**:
   - **Native**: A compressed format readable only by SolidFire storage systems.
   - **Uncompressed**: An uncompressed format compatible with other systems.
6. Enter a URL to use to access the object store in the **URL** field.
7. Enter a user name for the account in the **Username** field.
8. Enter the authentication key for the account in the **Authentication Key** field.
9. Enter the container in which to store the backup in the **Container** field.
10. **Optional**: Enter a name tag to append to the prefix in the **Nametag** field.
11. Click **Start Read**.

Back up a volume to a SolidFire storage cluster

You can back up volumes residing on a cluster to a remote cluster for storage clusters running Element software.

Ensure that the source and target clusters are paired.

See Pair clusters for replication.

When backing up or restoring from one cluster to another, the system generates a key to be used as authentication between the clusters. This bulk volume write key allows the source cluster to authenticate with the destination cluster, providing a level of security when writing to the destination volume. As part of the backup or restore process, you need to generate a bulk volume write key from the destination volume before starting the operation.

1. On the destination cluster, **Management** > **Volumes**.
2. Click the Actions icon for the destination volume.
3. In the resulting menu, click **Restore from**.
4. In the **Integrated Restore** dialog box, under **Restore from**, select **SolidFire**.
5. Select an option under **Data Format**:
   - **Native**: A compressed format readable only by SolidFire storage systems.
   - **Uncompressed**: An uncompressed format compatible with other systems.
6. Click **Generate Key**.
7. Copy the key from the **Bulk Volume Write Key** box to your clipboard.
8. On the source cluster, go to **Management** > **Volumes**.
9. Click the Actions icon for the volume to back up.
10. In the resulting menu, click **Backup to**.
11. In the **Integrated Backup** dialog box under **Backup to**, select **SolidFire**.
12. Select the same option you selected earlier in the **Data Format** field.
13. Enter the management virtual IP address of the destination volume's cluster in the **Remote Cluster MVIP** field.
14. Enter the remote cluster user name in the **Remote Cluster Username** field.

15. Enter the remote cluster password in the **Remote Cluster Password** field.
16. In the **Bulk Volume Write Key** field, paste the key you generated on the destination cluster earlier.
17. Click **Start Read**.

Restore a volume from backup on an Amazon S3 object store

You can restore a volume from a backup on an Amazon S3 object store.

1. Click **Reporting** > **Event Log**.
2. Locate the backup event that created the backup you need to restore.
3. In the **Details** column for the event, click **Show Details**.
4. Copy the manifest information to your clipboard.
5. Click **Management** > **Volumes**.
6. Click the Actions icon for the volume you want to restore.
7. In the resulting menu, click **Restore from**.
8. In the **Integrated Restore** dialog box under **Restore from**, select **S3**.
9. Select the option that matches the backup under **Data Format**:
    - **Native**: A compressed format readable only by SolidFire storage systems.
    - **Uncompressed**: An uncompressed format compatible with other systems.
10. Enter a hostname to use to access the object store in the **Hostname** field.
11. Enter an access key ID for the account in the **Access Key ID** field.
12. Enter the secret access key for the account in the **Secret Access Key** field.
13. Enter the S3 bucket in which to store the backup in the **S3 Bucket** field.
14. Paste the manifest information into the **Manifest** field.
15. Click **Start Write**.

Restore a volume from backup on an OpenStack Swift object store

You can restore a volume from a backup on an OpenStack Swift object store.

1. Click **Reporting** > **Event Log**.
2. Locate the backup event that created the backup you need to restore.
3. In the **Details** column for the event, click **Show Details**.
4. Copy the manifest information to your clipboard.
5. Click **Management** > **Volumes**.
6. Click the Actions icon for the volume you want to restore.
7. In the resulting menu, click **Restore from**.
8. In the **Integrated Restore** dialog box under **Restore from**, select **Swift**.
9. Select the option that matches the backup under **Data Format**:
    - **Native**: A compressed format readable only by SolidFire storage systems.
    - **Uncompressed**: An uncompressed format compatible with other systems.

10. Enter a URL to use to access the object store in the **URL** field.
11. Enter a user name for the account in the **Username** field.
12. Enter the authentication key for the account in the **Authentication Key** field.
13. Enter the name of the container in which the backup is stored in the **Container** field.
14. Paste the manifest information into the **Manifest** field.
15. Click **Start Write**.

Restore a volume from backup on a SolidFire storage cluster

You can restore a volume from a backup on a SolidFire storage cluster.

When backing up or restoring from one cluster to another, the system generates a key to be used as authentication between the clusters. This bulk volume write key allows the source cluster to authenticate with the destination cluster, providing a level of security when writing to the destination volume. As part of the backup or restore process, you need to generate a bulk volume write key from the destination volume before starting the operation.

1. On the destination cluster, click **Management** > **Volumes**.
2. Click the Actions icon for the volume you want to restore.
3. In the resulting menu, click **Restore from**.
4. In the **Integrated Restore** dialog box, under **Restore from**, select **SolidFire**.
5. Select the option that matches the backup under **Data Format**:
   - **Native**: A compressed format readable only by SolidFire storage systems.
   - **Uncompressed**: An uncompressed format compatible with other systems.
6. Click **Generate Key**.
7. Copy the **Bulk Volume Write Key** information to the clipboard.
8. On the source cluster, click **Management** > **Volumes**.
9. Click the Actions icon for the volume you want to use for the restore.
10. In the resulting menu, click **Backup to**.
11. In the **Integrated Backup** dialog box, select **SolidFire** under **Backup to**.
12. Select the option that matches the backup under **Data Format**.
13. Enter the management virtual IP address of the destination volume's cluster in the **Remote Cluster MVIP** field.
14. Enter the remote cluster user name in the **Remote Cluster Username** field.
15. Enter the remote cluster password in the **Remote Cluster Password** field.
16. Paste the key from your clipboard into the **Bulk Volume Write Key** field.
17. Click **Start Read**.

Configure custom Protection Domains

For Element clusters that contain more than two storage nodes, you can configure custom Protection Domains for each node. When you configure custom Protection Domains, you must assign all nodes in the cluster to a domain.

When you assign Protection Domains, a data sync between nodes begins, and some cluster operations are unavailable until the data sync completes. After a custom Protection Domain is configured for a cluster, when you add a new storage node, you cannot add drives for the new node until you assign a Protection Domain for the node and allow the data sync to complete. Visit the Protection Domains documentation to learn more about Protection Domains.

For a custom Protection Domain scheme to be useful for a cluster, all storage nodes within each chassis must be assigned to the same custom Protection Domain. You need to create as many custom Protection Domains as is needed for this to be the case (the smallest possible custom Protection Domain scheme is three domains). As a best practice, configure an equal number of nodes per domain and try to ensure that each node assigned to a particular domain is of the same type.

Steps

1. Click **Cluster** > **Nodes**.
2. Click **Configure Protection Domains**.

   In the **Configure Custom Protection Domains** window, you can see the currently configured Protection Domains (if any) as well as Protection Domain assignments for individual nodes.

3. Enter a name for the new custom Protection Domain, and click **Create**.

   Repeat this step for all new Protection Domains you need to create.

4. For each node in the **Assign Nodes** list, click the dropdown in the **Protection Domain** column and select a Protection Domain to assign to that node.

   Ensure that you understand your node and chassis layout, the custom Protection Domain scheme you have configured, and the scheme's effects on data protection before applying the changes. If you apply a Protection Domain scheme and immediately need to make changes, it could be some time before you can do so because of the data sync that happens once a configuration is applied.

5. Click **Configure Protection Domains**.

Result

Depending on the size of your cluster, the data sync data between domains could take some time. After the data sync is complete, you can view the custom Protection Domain assignments on the **Cluster** > **Nodes** page, and the Element web UI dashboard shows the protection status of the cluster in the **Custom Protection Domain Health** pane.

Possible errors

Here are some errors you might see after applying a custom Protection Domain configuration:

Error	Description	Resolution
SetProtectionDomainLayout Failed: ProtectionDomainLayout would leave NodeID {9} unusable. Default and non-default names cannot both be used together.	A node does not have a Protection Domain assigned.	Assign a Protection Domain to the node.

SetProtectionDomainLayout Failed: Protection domain type 'custom' splits Protection Domain type 'chassis'.	A node in a multi-node chassis is assigned a Protection Domain that is different from other nodes in the chassis.	Ensure that all nodes in the chassis are assigned the same Protection Domain.

Find more information

- Custom Protection Domains
- Manage storage with the Element API

Troubleshoot your system

You must monitor the system for diagnostic purposes and to get information about performance trends and statuses of various system operations. You might need to replace nodes or SSDs for maintenance purposes.

- View information about system events
- View status of running tasks
- View system alerts
- View node performance activity
- View volume performance
- View iSCSI sessions
- View Fibre Channel sessions
- Troubleshoot drives
- Troubleshoot nodes
- Work with per-node utilities for storage nodes
- Work with the management node
- Understand cluster fullness levels

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

View information about system events

You can view information about various events detected in the system. The system refreshes the event messages every 30 seconds. The event log displays key events for the cluster.

1. In the Element UI, select **Reporting** > **Event Log**.

   For every event, you see the following information:

Item	Description

ID	Unique ID associated with each event.
Event Type	The type of event being logged, for example, API events or clone events.
Message	Message associated with the event.
Details	Information that helps identify why the event occurred.
Service ID	The service that reported the event (if applicable).
Node	The node that reported the event (if applicable).
Drive ID	The drive that reported the event (if applicable).
Event Time	The time the event occurred.

Find more information

Event types

Event types

The system reports multiple types of events; each event is an operation that the system has completed. Events can be routine, normal events or events that require administrator attention. The Event Types column on the Event Log page indicates in which part of the system the event occurred.

The system does not log read-only API commands in the event log.

The following list describes the types of events that appear in the event log:

- **apiEvent**

  Events initiated by a user through an API or web UI that modify settings.

- **binAssignmentsEvent**

  Events related to the assignment of data bins. Bins are essentially containers that hold data and are mapped across the cluster.

- **binSyncEvent**

  System events related to a reassignment of data among block services.

- **bsCheckEvent**

  System events related to block service checks.

- **bsKillEvent**

  System events related to block service terminations.

- **bulkOpEvent**

  Events related to operations performed on an entire volume, such as a backup, restore, snapshot, or clone.

- **cloneEvent**

  Events related to volume cloning.

- **clusterMasterEvent**

  Events appearing upon cluster initialization or upon configuration changes to the cluster, such as adding or removing nodes.

- **cSumEvent**

  Events related to the detection of a checksum mismatch during end-to-end checksum validation.

  Services that detect a checksum mismatch are automatically stopped and not restarted after generating this event.

- **dataEvent**

  Events related to reading and writing data.

- **dbEvent**

  Events related to the global database maintained by ensemble nodes in the cluster.

- **driveEvent**

  Events related to drive operations.

- **encryptionAtRestEvent**

  Events related to the process of encryption on a cluster.

- **ensembleEvent**

  Events related to increasing or decreasing the number of nodes in an ensemble.

- **fibreChannelEvent**

  Events related to the configuration of and connections to the nodes.

- **gcEvent**

  Events related to processes run every 60 minutes to reclaim storage on block drives. This process is also known as garbage collection.

- **ieEvent**

  Internal system error.

- **installEvent**

  Automatic software installation events. Software is being automatically installed on a pending node.

- **iSCSIEvent**

  Events related to iSCSI issues in the system.

- **limitEvent**

  Events related to the number of volumes or virtual volumes in an account or in the cluster nearing the maximum allowed.

- **maintenanceModeEvent**

  Events related to the node maintenance mode, such as disabling the node.

- **networkEvent**

  Events related to the network error reporting for each physical network interface card (NIC) interface.

  These events are triggered when any error count for an interface exceeds a default threshold of 1000 during a 10-minute monitoring interval. These events apply to network errors such as received misses, cyclic redundancy check (CRC) errors, length errors, overrun errors, and frame errors.

- **platformHardwareEvent**

  Events related to issues detected on hardware devices.

- **remoteClusterEvent**

  Events related to remote cluster pairing.

- **schedulerEvent**

  Events related to scheduled snapshots.

- **serviceEvent**

  Events related to system service status.

- **sliceEvent**

  Events related to the Slice Server, such as removing a metadata drive or volume.

  There are three types of slice reassignment events, which include information about the service where a volume is assigned:

  - flipping: changing the primary service to a new primary service

    ```
    sliceID oldPrimaryServiceID->newPrimaryServiceID
    ```

  - moving: changing the secondary service to a new secondary service

```
sliceID {oldSecondaryServiceID(s)}->{newSecondaryServiceID(s)}
```

- pruning: removing a volume from a set of services

```
sliceID {oldSecondaryServiceID(s)}
```

- **snmpTrapEvent**

  Events related to SNMP traps.

- **statEvent**

  Events related to system statistics.

- **tsEvent**

  Events related to the system transport service.

- **unexpectedException**

  Events related to unexpected system exceptions.

- **ureEvent**

  Events related to Unrecoverable Read Errors that occur while reading from the storage device.

- **vasaProviderEvent**

  Events related to a VASA (vSphere APIs for Storage Awareness) Provider.

View status of running tasks

You can view the progress and completion status of running tasks in the web UI that are being reported by the ListSyncJobs and ListBulkVolumeJobs API methods. You can access the Running Tasks page from the Reporting tab of the Element UI.

If there are a large number of tasks, the system might queue them and run them in batches. The Running Tasks page displays the services currently being synchronized. When a task is complete, it is replaced by the next queued synchronizing task. Synchronizing tasks might continue to appear on the Running Tasks page until there are no more tasks to complete.

> You can see replication synchronizations data for volumes undergoing replication on the Running Tasks page of the cluster containing the target volume.

View system alerts

You can view alerts for information about cluster faults or errors in the system. Alerts can be information, warnings, or errors and are a good indicator of how well the cluster is running. Most errors resolve themselves automatically.

You can use the ListClusterFaults API method to automate alert monitoring. This enables you to be notified about all alerts that occur.

1. In the Element UI, select **Reporting** > **Alerts**.

   The system refreshes the alerts on the page every 30 seconds.

   For every event, you see the following information:

Item	Description
ID	Unique ID associated with a cluster alert.
Severity	The degree of importance of the alert. Possible values: • warning: A minor issue that might soon require attention. System upgrades are still allowed. • error: A failure that might cause performance degradation or loss of high availability (HA). Errors generally should not affect service otherwise. • critical: A serious failure that affects service. The system is unable to serve API or client I/O requests. Operating in this state could lead to potential loss of data. • bestPractice: A recommended system configuration best practice is not being used.
Type	The component that the fault affects. Can be node, drive, cluster, service, or volume.
Node	Node ID for the node that this fault refers to. Included for node and drive faults, otherwise set to - (dash).
Drive ID	Drive ID for the drive that this fault refers to. Included for drive faults, otherwise set to - (dash).
Error Code	A descriptive code that indicates what caused the fault.
Details	A description of the fault with additional details.
Date	The date and time the fault was logged.

2. Click **Show Details** for an individual alert to view information about the alert.
3. To view the details of all alerts on the page, click the Details column.

After the system resolves an alert, all information about the alert including the date it was resolved is moved to the Resolved area.

Find more information

- Cluster fault codes
- Manage storage with the Element API

Cluster fault codes

The system reports an error or a state that might be of interest by generating a fault code, which is listed on the Alerts page. These codes help you determine what component of the system experienced the alert and why the alert was generated.

The following list outlines the different types of codes:

- **authenticationServiceFault**

  The Authentication Service on one or more cluster nodes is not functioning as expected.

  Contact NetApp Support for assistance.

- **availableVirtualNetworkIPAddressesLow**

  The number of virtual network addresses in the block of IP addresses is low.

  To resolve this fault, add more IP addresses to the block of virtual network addresses.

- **blockClusterFull**

  There is not enough free block storage space to support a single node loss. See the GetClusterFullThreshold API method for details on cluster fullness levels. This cluster fault indicates one of the following conditions:

  - stage3Low (Warning): User-defined threshold was crossed. Adjust Cluster Full settings or add more nodes.
  - stage4Critical (Error): There is not enough space to recover from a 1-node failure. Creation of volumes, snapshots, and clones is not allowed.
  - stage5CompletelyConsumed (Critical)1; No writes or new iSCSI connections are allowed. Current iSCSI connections will be maintained. Writes will fail until more capacity is added to the cluster.

  To resolve this fault, purge or delete volumes or add another storage node to the storage cluster.

- **blocksDegraded**

  Block data is no longer fully replicated due to a failure.

Severity	Description
Warning	Only two complete copies of the block data are accessible.

Error	Only a single complete copy of the block data is accessible.
Critical	No complete copies of the block data are accessible.

Note: The warning status can only occur on a Triple Helix system.

To resolve this fault, restore any offline nodes or block services, or contact NetApp Support for assistance.

- **blockServiceTooFull**

  A block service is using too much space.

  To resolve this fault, add more provisioned capacity.

- **blockServiceUnhealthy**

  A block service has been detected as unhealthy:

  - Severity = Warning: No action is taken. This warning period will expire in cTimeUntilBSIsKilledMSec=330000 milliseconds.
  - Severity = Error: The system is automatically decommissioning data and re-replicating its data to other healthy drives.
  - Severity = Critical: There are failed block services on several nodes greater than or equal to the replication count (2 for double helix). Data is unavailable and bin syncing will not finish.

    Check for network connectivity issues and hardware errors. There will be other faults if specific hardware components have failed. The fault will clear when the block service is accessible or when the service has been decommissioned.

- **BmcSelfTestFailed**

  The Baseboard Management Controller (BMC) failed a self-test.

  Contact NetApp support for assistance.

  During an upgrade to Element 12.5 or later, the `BmcSelfTestFailed` fault is not generated for a node that has a preexisting failed BMC, or when a node's BMC fails during the upgrade. The BMCs that fail the self-tests during the upgrade will issue a `BmcSelfTestFailed` warning fault after the entire cluster completes the upgrade.

- **clockSkewExceedsFaultThreshold**

  Time skew between the Cluster master and the node which is presenting a token exceeds the recommended threshold. Storage cluster cannot correct the time skew between the nodes automatically.

  To resolve this fault, use NTP servers that are internal to your network, rather than the installation defaults. If you are using an internal NTP server, contact NetApp Support for assistance.

- **clusterCannotSync**

  There is an out-of-space condition and data on the offline block storage drives cannot be synced to drives

that are still active.

To resolve this fault, add more storage.

- **clusterFull**

  There is no more free storage space in the storage cluster.

  To resolve this fault, add more storage.

- **clusterIOPSAreOverProvisioned**

  Cluster IOPS are over provisioned. The sum of all minimum QoS IOPS is greater than the expected IOPS of the cluster. Minimum QoS cannot be maintained for all volumes simultaneously.

  To resolve this issue, lower the minimum QoS IOPS settings for volumes.

- **CpuThermalEventThreshold**

  The number of CPU thermal events on one or more CPUs exceeds the configured threshold.

  If no new CPU thermal events are detected within ten minutes, the warning will resolve itself.

- **disableDriveSecurityFailed**

  The cluster is not configured to enable drive security (Encryption at Rest), but at least one drive has drive security enabled, meaning that disabling drive security on those drives failed. This fault is logged with “Warning” severity.

  To resolve this fault, check the fault details for the reason why drive security could not be disabled. Possible reasons are:

  - The encryption key could not be acquired, investigate the problem with access to the key or the external key server.
  - The disable operation failed on the drive, determine whether the wrong key could possibly have been acquired.

  If neither of these are the reason for the fault, the drive might need to be replaced.

  You can attempt to recover a drive that does not successfully disable security even when the correct authentication key is provided. To perform this operation, remove the drive(s) from the system by moving it to Available, perform a secure erase on the drive and move it back to Active.

- **disconnectedClusterPair**

  A cluster pair is disconnected or configured incorrectly.

  Check network connectivity between the clusters.

- **disconnectedRemoteNode**

  A remote node is either disconnected or configured incorrectly.

  Check network connectivity between the nodes.

- **disconnectedSnapMirrorEndpoint**

A remote SnapMirror endpoint is disconnected or configured incorrectly.

Check network connectivity between the cluster and the remote SnapMirrorEndpoint.

- **driveAvailable**

  One or more drives are available in the cluster. In general, all clusters should have all drives added and none in the available state. If this fault appears unexpectedly, contact NetApp Support.

  To resolve this fault, add any available drives to the storage cluster.

- **driveFailed**

  The cluster returns this fault when one or more drives have failed, indicating one of the following conditions:

  - The drive manager cannot access the drive.
  - The slice or block service has failed too many times, presumably because of drive read or write failures, and cannot restart.
  - The drive is missing.
  - The master service for the node is inaccessible (all drives in the node are considered missing/failed).
  - The drive is locked and the authentication key for the drive cannot be acquired.
  - The drive is locked and the unlock operation fails.

    To resolve this issue:

  - Check network connectivity for the node.
  - Replace the drive.
  - Ensure that the authentication key is available.

- **driveHealthFault**

  A drive has failed the SMART health check and as a result, the drive's functions are diminished. There is a Critical severity level for this fault:

  - Drive with serial: <serial number> in slot: <node slot><drive slot> has failed the SMART overall health check.

  To resolve this fault, replace the drive.

- **driveWearFault**

  A drive's remaining life has dropped below thresholds, but it is still functioning.There are two possible severity levels for this fault: Critical and Warning:

  - Drive with serial: <serial number> in slot: <node slot><drive slot> has critical wear levels.
  - Drive with serial: <serial number> in slot: <node slot><drive slot> has low wear reserves.

    To resolve this fault, replace the drive soon.

- **duplicateClusterMasterCandidates**

  More than one storage cluster master candidate has been detected.

Contact NetApp Support for assistance.

- **enableDriveSecurityFailed**

  The cluster is configured to require drive security (Encryption at Rest), but drive security could not be enabled on at least one drive. This fault is logged with “Warning” severity.

  To resolve this fault, check the fault details for the reason why drive security could not be enabled. Possible reasons are:

  - The encryption key could not be acquired, investigate the problem with access to the key or the external key server.
  - The enable operation failed on the drive, determine whether the wrong key could possibly have been acquired.
    If neither of these are the reason for the fault, the drive might need to be replaced.

  You can attempt to recover a drive that does not successfully enable security even when the correct authentication key is provided. To perform this operation, remove the drive(s) from the system by moving it to Available, perform a secure erase on the drive and move it back to Active.

- **ensembleDegraded**

  Network connectivity or power has been lost to one or more of the ensemble nodes.

  To resolve this fault, restore network connectivity or power.

- **exception**

  A fault reported that is other than a routine fault. These faults are not automatically cleared from the fault queue.

  Contact NetApp Support for assistance.

- **failedSpaceTooFull**

  A block service is not responding to data write requests. This causes the slice service to run out of space to store failed writes.

  To resolve this fault, restore block services functionality to allow writes to continue normally and failed space to be flushed from the slice service.

- **fanSensor**

  A fan sensor has failed or is missing.

  To resolve this fault, replace any failed hardware.

- **fibreChannelAccessDegraded**

  A Fibre Channel node is not responding to other nodes in the storage cluster over its storage IP for a period of time. In this state, the node will then be considered unresponsive and generate a cluster fault.

  Check network connectivity.

- **fibreChannelAccessUnavailable**

All Fibre Channel nodes are unresponsive. The node IDs are displayed.

Check network connectivity.

- **fibreChannelActiveIxL**

  The IxL Nexus count is approaching the supported limit of 8000 active sessions per Fibre Channel node.

  - Best practice limit is 5500.
  - Warning limit is 7500.
  - Maximum limit (not enforced) is 8192.

  To resolve this fault, reduce the IxL Nexus count below the best practice limit of 5500.

- **fibreChannelConfig**

  This cluster fault indicates one of the following conditions:

  - There is an unexpected Fibre Channel port on a PCI slot.
  - There is an unexpected Fibre Channel HBA model.
  - There is a problem with the firmware of a Fibre Channel HBA.
  - A Fibre Channel port is not online.
  - There is a persistent issue configuring Fibre Channel passthrough.

  Contact NetApp Support for assistance.

- **fibreChannelIOPS**

  The total IOPS count is approaching the IOPS limit for Fibre Channel nodes in the cluster. The limits are:

  - FC0025: 450K IOPS limit at 4K block size per Fibre Channel node.
  - FCN001: 625K OPS limit at 4K block size per Fibre Channel node.

  To resolve this fault, balance the load across all available Fibre Channel nodes.

- **fibreChannelStaticIxL**

  The IxL Nexus count is approaching the supported limit of 16000 static sessions per Fibre Channel node.

  - Best practice limit is 11000.
  - Warning limit is 15000.
  - Maximum limit (enforced) is 16384.

  To resolve this fault, reduce the IxL Nexus count below the best practice limit of 11000.

- **fileSystemCapacityLow**

  There is insufficient space on one of the filesystems.

  To resolve this fault, add more capacity to the filesystem.

- **fileSystemIsReadOnly**

  A filesystem has moved into read-only mode.

Contact NetApp Support for assistance.

- **fipsDrivesMismatch**

  A non-FIPS drive has been physically inserted into a FIPS capable storage node or a FIPS drive has been physically inserted into a non-FIPS storage node. A single fault is generated per node and lists all drives affected.

  To resolve this fault, remove or replace the mismatched drive or drives in question.

- **fipsDrivesOutOfCompliance**

  The system has detected that Encryption at Rest was disabled after the FIPS Drives feature was enabled. This fault is also generated when the FIPS Drives feature is enabled and a non-FIPS drive or node is present in the storage cluster.

  To resolve this fault, enable Encryption at Rest or remove the non-FIPS hardware from the storage cluster.

- **fipsSelfTestFailure**

  The FIPS subsystem has detected a failure during the self test.

  Contact NetApp Support for assistance.

- **hardwareConfigMismatch**

  This cluster fault indicates one of the following conditions:

  - The configuration does not match the node definition.
  - There is an incorrect drive size for this type of node.
  - An unsupported drive has been detected. A possible reason is that the installed Element version does not recognize this drive. Recommend updating the Element software on this node.
  - There is a drive firmware mismatch.
  - The drive encryption capable state does not match the node.

  Contact NetApp Support for assistance.

- **idPCertificateExpiration**

  The cluster's service provider SSL certificate for use with a third-party identity provider (IdP) is nearing expiration or has already expired. This fault uses the following severities based on urgency:

Severity	Description
Warning	Certificate expires within 30 days.
Error	Certificate expires within 7 days.
Critical	Certificate expires within 3 days or has already expired.

  To resolve this fault, update the SSL certificate before it expires. Use the UpdateIdpConfiguration API

method with `refreshCertificateExpirationTime=true` to provide the updated SSL certificate.

- **inconsistentBondModes**

  The bond modes on the VLAN device are missing. This fault will display the expected bond mode and the bond mode currently in use.

- **inconsistentMtus**

  This cluster fault indicates one of the following conditions:

  - Bond1G mismatch: Inconsistent MTUs have been detected on Bond1G interfaces.
  - Bond10G mismatch: Inconsistent MTUs have been detected on Bond10G interfaces.

  This fault displays the node or nodes in question along with the associated MTU value.

- **inconsistentRoutingRules**

  The routing rules for this interface are inconsistent.

- **inconsistentSubnetMasks**

  The network mask on the VLAN device does not match the internally recorded network mask for the VLAN. This fault displays the expected network mask and the network mask currently in use.

- **incorrectBondPortCount**

  The number of bond ports is incorrect.

- **invalidConfiguredFibreChannelNodeCount**

  One of the two expected Fibre Channel node connections is degraded. This fault appears when only one Fibre Channel node is connected.

  To resolve this fault, check the cluster network connectivity and network cabling, and check for failed services. If there are no network or service problems, contact NetApp Support for a Fibre Channel node replacement.

- **irqBalanceFailed**

  An exception occurred while attempting to balance interrupts.

  Contact NetApp Support for assistance.

- **kmipCertificateFault**
  - Root Certification Authority (CA) certificate is nearing expiration.

    To resolve this fault, acquire a new certificate from the root CA with expiration date at least 30 days out and use ModifyKeyServerKmip to provide the updated root CA certificate.

  - Client certificate is nearing expiration.

    To resolve this fault, create a new CSR using GetClientCertificateSigningRequest, have it signed ensuring the new expiration date is at least 30 days out, and use ModifyKeyServerKmip to replace the expiring KMIP client certificate with the new certificate.

- Root Certification Authority (CA) certificate has expired.

  To resolve this fault, acquire a new certificate from the root CA with expiration date at least 30 days out and use ModifyKeyServerKmip to provide the updated root CA certificate.

- Client certificate has expired.

  To resolve this fault, create a new CSR using GetClientCertificateSigningRequest, have it signed ensuring the new expiration date is at least 30 days out, and use ModifyKeyServerKmip to replace the expired KMIP client certificate with the new certificate.

- Root Certification Authority (CA) certificate error.

  To resolve this fault, check that the correct certificate was provided, and, if needed, reacquire the certificate from the root CA. Use ModifyKeyServerKmip to install the correct KMIP client certificate.

- Client certificate error.

  To resolve this fault, check that the correct KMIP client certificate is installed. The root CA of the client certificate should be installed on the EKS. Use ModifyKeyServerKmip to install the correct KMIP client certificate.

- **kmipServerFault**
  - Connection failure

    To resolve this fault, check that the External Key Server is alive and reachable via the network. Use TestKeyServerKimp and TestKeyProviderKmip to test your connection.

  - Authentication failure

    To resolve this fault, check that the correct root CA and KMIP client certificates are being used, and that the private key and the KMIP client certificate match.

  - Server error

    To resolve this fault, check the details for the error. Troubleshooting on the External Key Server might be necessary based on the error returned.

- **memoryEccThreshold**

  A large number of correctable or uncorrectable ECC errors have been detected. This fault uses the following severities based on urgency:

Event	Severity	Description
A single DIMM cErrorCount reaches cDimmCorrectableErrWarnThreshold.	Warning	Correctable ECC memory errors above threshold on DIMM: <Processor> <DIMM Slot>

A single DIMM cErrorCount stays above cDimmCorrectableErrWarnThreshold until cErrorFaultTimer expires for the DIMM.	Error	Correctable ECC memory errors above threshold on DIMM: <Processor> <DIMM>
A memory controller reports cErrorCount above cMemCtlrCorrectableErrWarnThreshold, and cMemCtlrCorrectableErrWarnDuration is specified.	Warning	Correctable ECC memory errors above threshold on memory controller: <Processor> <Memory Controller>
A memory controller reports cErrorCount above cMemCtlrCorrectableErrWarnThreshold until cErrorFaultTimer expires for the memory controller.	Error	Correctable ECC memory errors above threshold on DIMM: <Processor> <DIMM>
A single DIMM reports a uErrorCount above zero, but less than cDimmUncorrectableErrFaultThreshold.	Warning	Uncorrectable ECC memory error(s) detected on DIMM: <Processor> <DIMM Slot>
A single DIMM reports a uErrorCount of at least cDimmUncorrectableErrFaultThreshold.	Error	Uncorrectable ECC memory error(s) detected on DIMM: <Processor> <DIMM Slot>
A memory controller reports a uErrorCount above zero, but less than cMemCtlrUncorrectableErrFaultThreshold.	Warning	Uncorrectable ECC memory error(s) detected on memory controller: <Processor> <Memory Controller>
A memory controller reports a uErrorCount of at least cMemCtlrUncorrectableErrFaultThreshold.	Error	Uncorrectable ECC memory error(s) detected on memory controller: <Processor> <Memory Controller>

To resolve this fault, contact NetApp Support for assistance.

- **memoryUsageThreshold**

  Memory usage is above normal. This fault uses the following severities based on urgency:

  See the **Details** heading in the error fault for more detailed information on the type of fault.

Severity	Description

Warning	System memory is low.
Error	System memory is very low.
Critical	System memory is completely consumed.

To resolve this fault, contact NetApp Support for assistance.

- **metadataClusterFull**

  There is not enough free metadata storage space to support a single node loss. See the GetClusterFullThreshold API method for details on cluster fullness levels. This cluster fault indicates one of the following conditions:

  - stage3Low (Warning): User-defined threshold was crossed. Adjust Cluster Full settings or add more nodes.
  - stage4Critical (Error): There is not enough space to recover from a 1-node failure. Creation of volumes, snapshots, and clones is not allowed.
  - stage5CompletelyConsumed (Critical)1; No writes or new iSCSI connections are allowed. Current iSCSI connections will be maintained. Writes will fail until more capacity is added to the cluster. Purge or delete data or add more nodes.

  To resolve this fault, purge or delete volumes or add another storage node to the storage cluster.

- **mtuCheckFailure**

  A network device is not configured for the proper MTU size.

  To resolve this fault, ensure that all network interfaces and switch ports are configured for jumbo frames (MTUs up to 9000 bytes in size).

- **networkConfig**

  This cluster fault indicates one of the following conditions:

  - An expected interface is not present.
  - A duplicate interface is present.
  - A configured interface is down.
  - A network restart is required.

  Contact NetApp Support for assistance.

- **noAvailableVirtualNetworkIPAddresses**

  There are no available virtual network addresses in the block of IP addresses.

  - virtualNetworkID # TAG(###) has no available storage IP addresses. Additional nodes cannot be added to the cluster.

  To resolve this fault, add more IP addresses to the block of virtual network addresses.

- **nodeHardwareFault (Network interface <name> is down or cable is unplugged)**

A network interface is either down or the cable is unplugged.

To resolve this fault, check network connectivity for the node or nodes.

- **nodeHardwareFault (Drive encryption capable state mismatches node's encryption capable state for the drive in slot <node slot><drive slot>)**

  A drive does not match encryption capabilities with the storage node it is installed in.

- **nodeHardwareFault (Incorrect <drive type> drive size <actual size> for the drive in slot <node slot><drive slot> for this node type - expected <expected size>)**

  A storage node contains a drive that is the incorrect size for this node.

- **nodeHardwareFault (Unsupported drive detected in slot <node slot><drive slot>; drive statistics and health information will be unavailable)**

  A storage node contains a drive it does not support.

- **nodeHardwareFault (The drive in slot <node slot><drive slot> should be using firmware version <expected version>, but is using unsupported version <actual version>)**

  A storage node contains a drive running an unsupported firmware version.

- **nodeMaintenanceMode**

  A node has been placed in maintenance mode. This fault uses the following severities based on urgency:

Severity	Description
Warning	Indicates that the node is still in maintenance mode.
Error	Indicates that maintenance mode has failed to disable, most likely due to failed or active standbys.

  To resolve this fault, disable maintenance mode once maintenance completes. If the Error level fault persists, contact NetApp Support for assistance.

- **nodeOffline**

  Element software cannot communicate with the specified node. Check network connectivity.

- **notUsingLACPBondMode**

  LACP bonding mode is not configured.

  To resolve this fault, use LACP bonding when deploying storage nodes; clients might experience performance issues if LACP is not enabled and properly configured.

- **ntpServerUnreachable**

  The storage cluster cannot communicate with the specified NTP server or servers.

  To resolve this fault, check the configuration for the NTP server, network, and firewall.

- **ntpTimeNotInSync**

  The difference between storage cluster time and the specified NTP server time is too large. The storage cluster cannot correct the difference automatically.

  To resolve this fault, use NTP servers that are internal to your network, rather than the installation defaults. If you are using internal NTP servers and the issue persists, contact NetApp Support for assistance.

- **nvramDeviceStatus**

  An NVRAM device has an error, is failing, or has failed. This fault has the following severities:

Severity	Description
Warning	A warning has been detected by the hardware. This condition may be transitory, such as a temperature warning. • nvmLifetimeError • nvmLifetimeStatus • energySourceLifetimeStatus • energySourceTemperatureStatus • warningThresholdExceeded
Error	An Error or Critical status has been detected by the hardware. The cluster master attempts to remove the slice drive from operation (this generates a drive removal event). If secondary slice services are not available the drive will not be removed. Errors returned in addition to the Warning level errors: • NVRAM device mount point doesn’t exist. • NVRAM device partition doesn’t exist. • NVRAM device partition exists, but not mounted.
Critical	An Error or Critical status has been detected by the hardware. The cluster master attempts to remove the slice drive from operation (this generates a drive removal event). If secondary slice services are not available the drive will not be removed. • persistenceLost • armStatusSaveNArmed • csaveStatusError

  Replace any failed hardware in the node. If this does not resolve the issue, contact NetApp Support for assistance.

- **powerSupplyError**

  This cluster fault indicates one of the following conditions:

  - A power supply is not present.
  - A power supply has failed.
  - A power supply input is missing or out of range.

  To resolve this fault, verify that redundant power is supplied to all nodes. Contact NetApp Support for assistance.

- **provisionedSpaceTooFull**

  The overall provisioned capacity of the cluster is too full.

  To resolve this fault, add more provisioned space, or delete and purge volumes.

- **remoteRepAsyncDelayExceeded**

  The configured asynchronous delay for replication has been exceeded. Check network connectivity between clusters.

- **remoteRepClusterFull**

  The volumes have paused remote replication because the target storage cluster is too full.

  To resolve this fault, free up some space on the target storage cluster.

- **remoteRepSnapshotClusterFull**

  The volumes have paused remote replication of snapshots because the target storage cluster is too full.

  To resolve this fault, free up some space on the target storage cluster.

- **remoteRepSnapshotsExceededLimit**

  The volumes have paused remote replication of snapshots because the target storage cluster volume has exceeded its snapshot limit.

  To resolve this fault, increase the snapshot limit on the target storage cluster.

- **scheduleActionError**

  One or more of the scheduled activities ran, but failed.

  The fault clears if the scheduled activity runs again and succeeds, if the scheduled activity is deleted, or if the activity is paused and resumed.

- **sensorReadingFailed**

  A sensor could not communicate with the Baseboard Management Controller (BMC).

  Contact NetApp Support for assistance.

- **serviceNotRunning**

A required service is not running.

Contact NetApp Support for assistance.

- **sliceServiceTooFull**

  A slice service has too little provisioned capacity assigned to it.

  To resolve this fault, add more provisioned capacity.

- **sliceServiceUnhealthy**

  The system has detected that a slice service is unhealthy and is automatically decommissioning it.

  - Severity = Warning: No action is taken. This warning period will expire in 6 minutes.
  - Severity = Error: The system is automatically decommissioning data and re-replicating its data to other healthy drives.

  Check for network connectivity issues and hardware errors. There will be other faults if specific hardware components have failed. The fault will clear when the slice service is accessible or when the service has been decommissioned.

- **sshEnabled**

  The SSH service is enabled on one or more nodes in the storage cluster.

  To resolve this fault, disable the SSH service on the appropriate node or nodes or contact NetApp Support for assistance.

- **sslCertificateExpiration**

  The SSL certificate associated with this node is nearing expiration or has expired. This fault uses the following severities based on urgency:

Severity	Description
Warning	Certificate expires within 30 days.
Error	Certificate expires within 7 days.
Critical	Certificate expires within 3 days or has already expired.

  To resolve this fault, renew the SSL certificate. If needed, contact NetApp Support for assistance.

- **strandedCapacity**

  A single node accounts for more than half of the storage cluster capacity.

  In order to maintain data redundancy, the system reduces the capacity of the largest node so that some of its block capacity is stranded (not used).

  To resolve this fault, add more drives to existing storage nodes or add storage nodes to the cluster.

- **tempSensor**

  A temperature sensor is reporting higher than normal temperatures. This fault can be triggered in conjunction with powerSupplyError or fanSensor faults.

  To resolve this fault, check for airflow obstructions near the storage cluster. If needed, contact NetApp Support for assistance.

- **upgrade**

  An upgrade has been in progress for more than 24 hours.

  To resolve this fault, resume the upgrade or contact NetApp Support for assistance.

- **unresponsiveService**

  A service has become unresponsive.

  Contact NetApp Support for assistance.

- **virtualNetworkConfig**

  This cluster fault indicates one of the following conditions:

  - An interface is not present.
  - There is an incorrect namespace on an interface.
  - There is an incorrect netmask.
  - There is an incorrect IP address.
  - An interface is not up and running.
  - There is a superfluous interface on a node.

  Contact NetApp Support for assistance.

- **volumesDegraded**

  Secondary volumes have not finished replicating and synchronizing. The message is cleared when the synchronizing is complete.

- **volumesOffline**

  One or more volumes in the storage cluster are offline. The **volumeDegraded** fault will also be present.

  Contact NetApp Support for assistance.

View node performance activity

You can view performance activity for each node in a graphical format. This information provides real-time statistics for CPU and read/write I/O operations per second (IOPS) for each drive the node. The utilization graph is updated every five seconds, and the drive statistics graph updates every ten seconds.

1. Click **Cluster** > **Nodes**.

2. Click **Actions** for the node you want to view.
3. Click **View Details**.

> You can see specific points in time on the line and bar graphs by positioning your cursor over the line or bar.

View volume performance

You can view detailed performance information for all volumes in the cluster. You can sort the information by volume ID or by any of the performance columns. You can also use filter the information by certain criteria.

You can change how often the system refreshes performance information on the page by clicking the **Refresh every** list, and choosing a different value. The default refresh interval is 10 seconds if the cluster has less than 1000 volumes; otherwise, the default is 60 seconds. If you choose a value of Never, automatic page refreshing is disabled.

You can reenable automatic refreshing by clicking **Turn on auto-refresh**.

1. In the Element UI, select **Reporting** > **Volume Performance**.
2. In the volume list, click the Actions icon for a volume.
3. Click **View Details**.

   A tray is displayed at the bottom of the page containing general information about the volume.

4. To see more detailed information about the volume, click **See More Details**.

   The system displays detailed information as well as performance graphs for the volume.

Find more information

Volume performance details

Volume performance details

You can view performance statistics of volumes from the Volume Performance page of the Reporting tab in the Element UI.

The following list describes the details that are available to you:

- **ID**

  The system-generated ID for the volume.

- **Name**

  The name given to the volume when it was created.

- **Account**

  The name of the account assigned to the volume.

- **Access Groups**

  The name of the volume access group or groups to which the volume belongs.

- **Volume Utilization**

  A percentage value that describes how much the client is using the volume.

  Possible values:

  - 0 = Client is not using the volume
  - 100 = Client is using the max
  - >100 = Client is using the burst

- **Total IOPS**

  The total number of IOPS (read and write) currently being executed against the volume.

- **Read IOPS**

  The total number of read IOPS currently being executed against the volume.

- **Write IOPS**

  The total number of write IOPS currently being executed against the volume.

- **Total Throughput**

  The total amount of throughput (read and write) currently being executed against the volume.

- **Read Throughput**

  The total amount of read throughput currently being executed against the volume.

- **Write Throughput**

  The total amount of write throughput currently being executed against the volume.

- **Total Latency**

  The average time, in microseconds, to complete read and write operations to a volume.

- **Read Latency**

  The average time, in microseconds, to complete read operations to the volume in the last 500 milliseconds.

- **Write Latency**

  The average time, in microseconds, to complete write operations to a volume in the last 500 milliseconds.

- **Queue Depth**

  The number of outstanding read and write operations to the volume.

- **Average IO Size**

Average size in bytes of recent I/O to the volume in the last 500 milliseconds.

View iSCSI sessions

You can view the iSCSI sessions that are connected to the cluster. You can filter the information to include only the desired sessions.

1. In the Element UI, select **Reporting** > **iSCSI Sessions**.
2. To see the filter criteria fields, click **Filter**.

Find more information

iSCSI session details

iSCSI session details

You can view information about the iSCSI sessions that are connected to the cluster.

The following list describes the information that you can find about the iSCSI sessions:

- **Node**

  The node hosting the primary metadata partition for the volume.

- **Account**

  The name of the account that owns the volume. If value is blank, a dash (-) is displayed.

- **Volume**

  The volume name identified on the node.

- **Volume ID**

  ID of the volume associated with the Target IQN.

- **Initiator ID**

  A system-generated ID for the initiator.

- **Initiator Alias**

  An optional name for the initiator that makes finding the initiator easier when in a long list.

- **Initator IP**

  The IP address of the endpoint that initiates the session.

- **Initiator IQN**

  The IQN of the endpoint that initiates the session.

- **Target IP**

The IP address of the node hosting the volume.

- **Target IQN**

  The IQN of the volume.

- **Created On**

  Date the session was established.

View Fibre Channel sessions

You can view the Fibre Channel (FC) sessions that are connected to the cluster. You can filter information to include only those connections you want displayed in the window.

1. In the Element UI, select **Reporting** > **FC Sessions**.
2. To see the filter criteria fields, click **Filter**.

Find more information

Fibre Channel session details

Fibre Channel session details

You can find information about the active Fibre Channel (FC) sessions that are connected to the cluster.

The following list describes the information you can find about the FC sessions connected to the cluster:

- **Node ID**

  The node hosting the session for the connection.

- **Node Name**

  System-generated node name.

- **Initiator ID**

  A system-generated ID for the initiator.

- **Initiator WWPN**

  The initiating worldwide port name.

- **Initiator Alias**

  An optional name for the initiator that makes finding the initiator easier when in a long list.

- **Target WWPN**

  The target worldwide port name.

- **Volume Access Group**

  Name of the volume access group that the session belongs to.

- **Volume Access Group ID**

  System-generated ID for the access group.

Troubleshoot drives

You can replace a failed solid-state drive (SSD) with a replacement drive. SSDs for SolidFire storage nodes are hot-swappable. If you suspect an SSD has failed, contact NetApp Support to verify the failure and walk you through the proper resolution procedure. NetApp Support also works with you to get a replacement drive according to your service-level agreement.

How-swappable in this case means that you can remove a failed drive from an active node and replace it with a new SSD drive from NetApp. It is not recommended that you should remove non-failed drives on an active cluster.

You should maintain on-site spares suggested by NetApp Support to allow for immediate replacement of the drive if it fails.

> For testing purposes, if you are simulating a drive failure by pulling a drive from a node, you must wait 30 seconds before inserting the drive back into the drive slot.

If a drive fails, Double Helix redistributes the data on the drive across the nodes remaining on the cluster. Multiple drive failures on the same node are not an issue since Element software protects against two copies of data residing on the same node. A failed drive results in the following events:

- Data is migrated off of the drive.
- Overall cluster capacity is reduced by the capacity of the drive.
- Double Helix data protection ensures that there are two valid copies of the data.

> SolidFire storage systems do not support removal of a drive if it results in an insufficient amount of storage to migrate data.

For more information

- Remove failed drives from the cluster
- Basic MDSS drive troubleshooting
- Remove MDSS drives
- Replacing drives for SolidFire storage nodes
- Replacing drives for H600S series storage nodes
- H410S and H610S hardware information
- SF-series hardware information

Remove failed drives from the cluster

The SolidFire system puts a drive in a failed state if the drive's self-diagnostics tells the node it has failed or if communication with the drive stops for five and a half minutes or longer. The system displays a list of the failed drives. You must remove a failed drive from the failed drive list in NetApp Element software.

Drives in the **Alerts** list show as **blockServiceUnhealthy** when a node is offline. When restarting the node, if the node and its drives come back online within five and a half minutes, the drives automatically update and continue as active drives in the cluster.

1. In the Element UI, select **Cluster** > **Drives**.
2. Click **Failed** to view the list of failed drives.
3. Note the slot number of the failed drive.

   You need this information to locate the failed drive in the chassis.

4. Remove the failed drives using one of the following methods:

Option	Steps
To remove individual drives	a. Click **Actions** for the drive you want to remove. b. Click **Remove**.
To remove multiple drives	a. Select all the drives you want to remove, and click **Bulk Actions**. b. Click **Remove**.

Basic MDSS drive troubleshooting

You can recover metadata (or slice) drives by adding them back to the cluster in the event that one or both metadata drives fail. You can perform the recovery operation in the NetApp Element UI if the MDSS feature is already enabled on the node.

If either or both of the metadata drives in a node experiences a failure, the slice service will shut down and data from both drives will be backed up to different drives in the node.

The following scenarios outline possible failure scenarios, and provide basic recommendations to correct the issue:

System slice drive fails

- In this scenario, the slot 2 is verified and returned to an available state.
- The system slice drive must be repopulated before the slice service can be brought back online.
- You should replace the system slice drive, when the system slice drive becomes available, add the drive and the slot 2 drive at the same time.

You cannot add the drive in slot 2 by itself as a metadata drive. You must add both drives back to the node at the same time.

Slot 2 fails

- In this scenario, the system slice drive is verified and returned to an available state.
- You should replace slot 2 with a spare, when slot 2 becomes available, add the system slice drive and the slot 2 drive at the same time.

System slice drive and slot 2 fails

- You should replace both system slice drive and slot 2 with a spare drive. When both drives become available, add the system slice drive and the slot 2 drive at the same time.

Order of operations

- Replace the failed hardware drive with a spare drive (replace both drives if both have failed).
- Add drives back to the cluster when they have been repopulated and are in an available state.

Verify operations

- Verify that the drives in slot 0 (or internal) and slot 2 are identified as metadata drives in the Active Drives list.
- Verify that all slice balancing has completed (there are no further moving slices messages in the event log for at least 30 minutes).

For more information

Add MDSS drives

Add MDSS drives

You can add a second metadata drive on a SolidFire node by converting the block drive in slot 2 to a slice drive. This is accomplished by enabling the multi-drive slice service (MDSS) feature. To enable this feature, you must contact NetApp Support.

Getting a slice drive into an available state might require replacing a failed drive with a new or spare drive. You must add the system slice drive at the same time you add the drive for slot 2. If you try to add the slot 2 slice drive alone or before you add the system slice drive, the system will generate an error.

1. Click **Cluster** > **Drives**.
2. Click **Available** to view the list of available drives.
3. Select the slice drives to add.
4. Click **Bulk Actions**.
5. Click **Add**.
6. Confirm from the **Active Drives** tab that the drives have been added.

Remove MDSS drives

You can remove the multi-drive slice service (MDSS) drives. This procedure applies only if the node has multiple slice drives.

> If the system slice drive and the slot 2 drive fail, the system will shutdown slice services and remove the drives. If there is no failure and you remove the drives, both drives must be removed at the same time.

1. Click **Cluster** > **Drives**.
2. From the **Available** drives tab, click the check box for the slice drives being removed.
3. Click **Bulk Actions**.
4. Click **Remove**.
5. Confirm the action.

Troubleshoot nodes

You can remove nodes from a cluster for maintenance or replacement. You should use the NetApp Element UI or API to remove nodes before taking them offline.

An overview of the procedure to remove storage nodes is as follows:

- Ensure that there is sufficient capacity in the cluster to create a copy of the data on the node.
- Remove drives from the cluster by using the UI or the RemoveDrives API method.

  This results in the system migrating data from the node's drives to other drives in the cluster. The time this process takes is dependent on how much data must be migrated.

- Remove the node from the cluster.

Keep the following considerations in mind before you power down or power up a node:

- Powering down nodes and clusters involves risks if not performed properly.

  Powering down a node should be done under the direction of NetApp Support.

- If a node has been down longer than 5.5 minutes under any type of shutdown condition, Double Helix data protection begins the task of writing single replicated blocks to another node to replicate the data. In this case, contact NetApp Support for help with analyzing the failed node.
- To safely reboot or power down a node, you can use the Shutdown API command.
- If a node is in a down, or in an off state, you must contact NetApp Support before bringing it back online.
- After a node is brought back online, you must add the drives back to the cluster, depending on how long it has been out of service.

For more information

Replacing a failed SolidFire chassis

Replacing a failed H600S series node

Power down a cluster

Perform the following procedure to power down an entire cluster.

Steps

1. (Optional) Contact NetApp Support for assistance with completing the preliminary steps.
2. Verify that all I/O has stopped.
3. Disconnect all iSCSI sessions:
   a. Navigate to the management virtual IP (MVIP) address on the cluster to open the Element UI.
   b. Note the nodes listed in the Nodes list.
   c. Run the Shutdown API method with the halt option specified on each Node ID in the cluster.

> When you restart the cluster, you must follow certain steps to verify that all nodes come online:
>
> 1. Verify that all Critical severity and `volumesOffline` cluster faults have been resolved.
> 2. Wait for 10 to 15 minutes for the cluster to settle.
> 3. Start bringing up the hosts to access the data.
>
> If you want to allow more time when powering on nodes and verifying that they are healthy after maintenance, contact technical support for assistance with delaying data synchronization to prevent unnecessary bin syncing.

Find more information

How to gracefully shut down and power on a NetApp Solidfire/HCI storage cluster

Work with per-node utilities for storage nodes

You can use the per-node utilities to troubleshoot network problems if the standard monitoring tools in the NetApp Element software UI do not give you enough information for troubleshooting. Per-node utilities provide specific information and tools that can help you troubleshoot network problems between nodes or with the management node.

Find more information

- Access per-node settings using the per-node UI
- Network settings details from the per-node UI
- Cluster settings details from the per-node UI
- Run system tests using the per-node UI
- Run system utilities using the per-node UI

Access per-node settings using the per-node UI

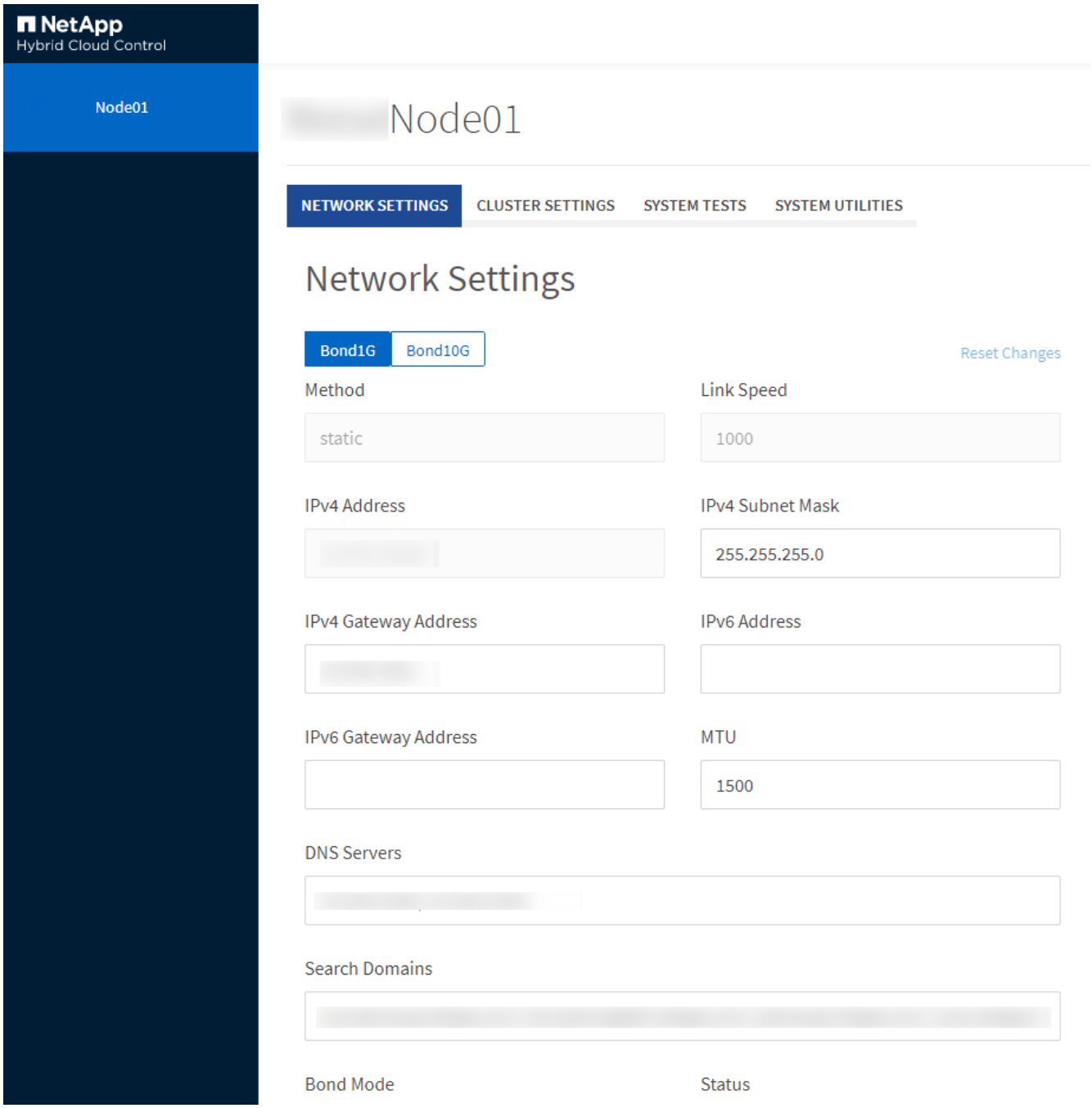
You can access network settings, cluster settings, and system tests and utilities in the per-node user interface after you enter the management node IP and authenticate.

If you want to modify settings of a node in an Active state that is part of a cluster, you must log in as a cluster

administrator user.

You should configure or modify one node at a time. You should ensure that the network settings specified are having the expected effect, and that the network is stable and performing well before you make modifications to another node.

1. Open the per-node UI using one of the following methods:
   - Enter the management IP address followed by :442 in a browser window, and log in using an admin user name and password.
   - In the Element UI, select **Cluster** > **Nodes**, and click the management IP address link for the node you want to configure or modify.
     In the browser window that opens, you can edit the settings of the node.



Network settings details from the per-node UI

You can change the storage node network settings to give the node a new set of network attributes.

You can see the network settings for a storage node on the **Network Settings** page when you log in to the node (https://<node IP>:442/hcc/node/network-settings). You can select either **Bond1G** (management) or **Bond10G** (storage) settings. The following list describes the settings that you can modify when a storage node is in Available, Pending, or Active state:

- **Method**

  The method used to configure the interface. Possible methods:

  - loopback: Used to define the IPv4 loopback interface.
  - manual: Used to define interfaces for which no configuration is done by default.
  - dhcp: Used to obtain an IP address via DHCP.
  - static: Used to define Ethernet interfaces with statically allocated IPv4 addresses.

- **Link Speed**

  The speed negotiated by the virtual NIC.

- **IPv4 Address**

  The IPv4 address for the eth0 network.

- **IPv4 Subnet Mask**

  Address subdivisions of the IPv4 network.

- **IPv4 Gateway Address**

  Router network address to send packets out of the local network.

- **IPv6 Address**

  The IPv6 address for the eth0 network.

- **IPv6 Gateway Address**

  Router network address to send packets out of the local network.

- **MTU**

  Largest packet size that a network protocol can transmit. Must be greater than or equal to 1500. If you add a second storage NIC, the value should be 9000.

- **DNS Servers**

  Network interface used for cluster communication.

- **Search Domains**

  Search for additional MAC addresses available to the system.

- **Bond Mode**

  Can be one of the following modes:

  - ActivePassive (default)
  - ALB
  - LACP
- **Status**

  Possible values:

  - UpAndRunning
  - Down
  - Up
- **Virtual Network Tag**

  Tag assigned when the virtual network was created.
- **Routes**

  Static routes to specific hosts or networks via the associated interface the routes are configured to use.

Cluster settings details from the per-node UI

You can verify cluster settings for a storage node after cluster configuration and modify the node hostname.

The following list describes the cluster settings for a storage node indicated from the **Cluster Settings** page of the per-node UI (https://<node IP>:442/hcc/node/cluster-settings).

- **Role**

  Role the node has in the cluster. Possible values:

  - Storage: Storage or Fibre Channel node.
  - Management: Node is a management node.
- **Hostname**

  Name of the node.
- **Cluster**

  Name of the cluster.
- **Cluster Membership**

  State of the node. Possible values:

  - Available: The node has no associated cluster name and is not yet part of a cluster.
  - Pending: The node is configured and can be added to a designated cluster. Authentication is not

required to access the node.

- PendingActive: The system is in the process of installing compatible software on the node. When complete, the node will move to the Active state.
- Active: The node is participating in a cluster. Authentication is required to modify the node.

- **Version**

  Version of the Element software running on the node.

- **Ensemble**

  Nodes that are part of the database ensemble.

- **Node ID**

  ID assigned when a node is added to the cluster.

- **Cluster Interface**

  Network interface used for cluster communication.

- **Management Interface**

  Management network interface. This defaults to Bond1G but can also use Bond10G.

- **Storage Interface**

  Storage network interface using Bond10G.

- **Encryption Capable**

  Indicates whether or not the node supports drive encryption.

Run system tests using the per-node UI

You can test changes to the network settings after you commit them to the network configuration. You can run the tests to ensure that the storage node is stable and can be brought online without any issues.

You have logged in to the per-node UI for the storage node.

1. Click **System Tests**.
2. Click **Run Test** next to the test you want to run or select **Run All Tests**.

   Running all test operations can be time consuming and should be done only at the direction of NetApp Support.

   - **Test Connected Ensemble**

     Tests and verifies the connectivity to a database ensemble. By default, the test uses the ensemble for the cluster the node is associated with. Alternatively you can provide a different ensemble to test connectivity.

- **Test Connect Mvip**

  Pings the specified management virtual IP (MVIP) address and then executes a simple API call to the MVIP to verify connectivity. By default, the test uses the MVIP for the cluster the node is associated with.

- **Test Connect Svip**

  Pings the specified storage virtual IP (SVIP) address using Internet Control Message Protocol (ICMP) packets that match the Maximum Transmission Unit (MTU) size set on the network adapter. It then connects to the SVIP as an iSCSI initiator. By default, the test uses the SVIP for the cluster the node is associated with.

- **Test Hardware Config**

  Tests that all hardware configurations are correct, validates firmware versions are correct, and confirms all drives are installed and running properly. This is the same as factory testing.

  This test is resource intensive and should only be run if requested by NetApp Support.

- **Test Local Connectivity**

  Tests the connectivity to all of the other nodes in the cluster by pinging the cluster IP (CIP) on each node. This test will only be displayed on a node if the node is part of an active cluster.

- **Test Locate Cluster**

  Validates that the node can locate the cluster specified in the cluster configuration.

- **Test Network Config**

  Verifies that the configured network settings match the network settings being used on the system. This test is not intended to detect hardware failures when a node is actively participating in a cluster.

- **Test Ping**

  Pings a specified list of hosts or, if none are specified, dynamically builds a list of all registered nodes in the cluster and pings each for simple connectivity.

- **Test Remote Connectivity**

  Tests the connectivity to all nodes in remotely paired clusters by pinging the cluster IP (CIP) on each node. This test will only be displayed on a node if the node is part of an active cluster.

Run system utilities using the per-node UI

You can use the per-node UI for the storage node to create or delete support bundles, reset configuration settings for drives, and restart network or cluster services.

You have logged in to the per-node UI for the storage node.

1. Click **System Utilities**.
2. Click the button for the system utility that you want to run.

- **Control Power**

  Reboots, power cycles, or shuts down the node.

  > This operation causes temporary loss of networking connectivity.

  Specify the following parameters:

  - Action: Options include Restart and Halt (power off).
  - Wakeup Delay: Any additional time before the node comes back online.

- **Collect Node Logs**

  Creates a support bundle under the node's /tmp/bundles directory.

  Specify the following parameters:

  - Bundle Name: Unique name for each support bundle created. If no name is provided, then "supportbundle" and the node name are used as the file name.
  - Extra Args: This parameter is fed to the sf_make_support_bundle script. This parameter should be used only at the request of NetApp Support.
  - Timeout Sec: Specify the number of seconds to wait for each individual ping response.

- **Delete Node Logs**

  Deletes any current support bundles on the node that were created using **Create Cluster Support Bundle** or the CreateSupportBundle API method.

- **Reset Drives**

  Initializes drives and removes all data currently residing on the drive. You can reuse the drive in an existing node or in an upgraded node.

  Specify the following parameter:

  - Drives: List of device names (not driveIDs) to reset.

- **Reset Network Config**

  Helps resolve network configuration issues for an individual node and resets an individual node's network configuration to the factory default settings.

- **Reset Node**

  Resets a node to the factory settings. All data is removed but network settings for the node are preserved during this operation. Nodes can only be reset if they are unassigned to a cluster and in Available state.

  > All data, packages (software upgrades), configurations, and log files are deleted from the node when you use this option.

- **Restart Networking**

  Restarts all networking services on a node.

This operation can cause temporary loss of network connectivity.

- **Restart Services**

  Restarts Element software services on a node.

  This operation can cause temporary node service interruption. You should perform this operation only at the direction of NetApp Support.

  Specify the following parameters:

  - Service: Service name to be restarted.
  - Action: Action to perform on the service. Options include start, stop and restart.

Work with the management node

You can use the management node (mNode) to upgrade system services, manage cluster assets and settings, run system tests and utilities, configure Active IQ for system monitoring, and enable NetApp Support access for troubleshooting.

As a best practice, only associate one management node with one VMware vCenter instance, and avoid defining the same storage and compute resources or vCenter instances in multiple management nodes.

See management node documentation for more information.

Understand cluster fullness levels

The cluster running Element software generates cluster faults to warn the storage administrator when the cluster is running out of capacity. There are three levels of cluster fullness, all of which are displayed in the NetApp Element UI: warning, error, and critical.

The system uses the BlockClusterFull error code to warn about cluster block storage fullness. You can view the cluster fullness severity levels from the Alerts tab of the Element UI.

The following list includes information about the BlockClusterFull severity levels:

- **Warning**

  This is a customer-configurable warning that appears when the cluster's block capacity is approaching the error severity level. By default, this level is set at three percent under the error level and can be tuned via the Element UI and API. You must add more capacity, or free up capacity as soon as possible.

- **Error**

  When the cluster is in this state, if a node is lost, there will not be enough capacity in the cluster to rebuild Double Helix data protection. New volume creation, clones, and snapshots are all blocked while the cluster is in this state. This is not a safe or recommended state for any cluster to be in. You must add more capacity, or free up capacity immediately.

- **Critical**

This critical error has occurred because the cluster is 100 percent consumed. It is in a read-only state and no new iSCSI connections can be made to the cluster. When this stage is reached, you must free up or add more capacity immediately.

The system uses the MetadataClusterFull error code to warn about cluster metadata storage fullness. You can view the cluster metadata storage fullness from the Cluster Capacity section on the Overview page of the Reporting tab in the Element UI.

The following list includes information about the MetadataClusterFull severity levels:

- **Warning**

  This is a customer-configurable warning that appears when the cluster's metatdata capacity is approaching the error severity level. By default, this level is set at three percent under the error level and can be tuned via the Element API. You must add more capacity, or free up capacity as soon as possible.

- **Error**

  When the cluster is in this state, if a node is lost, there will not be enough capacity in the cluster to rebuild Double Helix data protection. New volume creation, clones, and snapshots are all blocked while the cluster is in this state. This is not a safe or recommended state for any cluster to be in. You must add more capacity, or free up capacity immediately.

- **Critical**

  This critical error has occurred because the cluster is 100 percent consumed. It is in a read-only state and no new iSCSI connections can be made to the cluster. When this stage is reached, you must free up or add more capacity immediately.

The following applies to two-node cluster thresholds:

- Metadata fullness error is 20% below critical.
- Block fullness error is 1 block drive (including stranded capacity) below critical; meaning that it is two block drives worth of capacity below critical.

Manage and monitor storage with NetApp Hybrid Cloud Control

With NetApp SolidFire all-flash storage, you can manage and monitor storage assets and configure components in your storage system using the NetApp Hybrid Cloud Control.

- Add and manage storage clusters
- Configure Fully Qualified Domain Name web UI access
- Create and manage user accounts
- Create and manage volumes
- Create and manage volume access groups
- Create and manage initiators
- Create and manage volume QoS policies
- Monitor your SolidFire system with NetApp Hybrid Cloud Control

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Add and manage storage clusters using NetApp Hybrid Cloud Control

You can add storage clusters to the management node assets inventory so that they can be managed using NetApp Hybrid Cloud Control (HCC). The first storage cluster added during system setup is the default authoritative storage cluster, but additional clusters can be added using HCC UI.

After a storage cluster is added, you can monitor cluster performance, change storage cluster credentials for the managed asset, or remove a storage cluster from the management node asset inventory if it no longer needs to be managed using HCC.

What you'll need

- **Cluster administrator permissions**: You have permissions as administrator on the authoritative storage cluster. The authoritative cluster is the first cluster added to the management node inventory during system setup.
- **Element software**: Your storage cluster version is running NetApp Element software 11.3 or later.
- **Management node**: You have deployed a management node running version 11.3 or later.
- **Management services**: You have updated your management services bundle to version 2.17 or later.

Options

- Add a storage cluster
- Confirm storage cluster status

- Edit storage cluster credentials
- Remove a storage cluster
- Enable and disable maintenance mode

Add a storage cluster

You can add a storage cluster to the management node assets inventory using NetApp Hybrid Cloud Control. This allows you to manage and monitor the cluster using the HCC UI.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the authoritative storage cluster administrator credentials.
2. From the Dashboard, select the options menu on the top right and select **Configure**.
3. In the **Storage Clusters** pane, select **Storage Cluster Details**.
4. Select **Add Storage Cluster**.
5. Enter the following information:
   - Storage cluster management virtual IP address

     Only remote storage clusters that are not currently managed by a management node can be added.

   - Storage cluster user name and password
6. Select **Add**.

   After you add the storage cluster, the cluster inventory can take up to 2 minutes to refresh and display the new addition. You might need to refresh the page in your browser to see the changes.

Confirm storage cluster status

You can monitor the connection status of storage clusters assets using the NetApp Hybrid Cloud Control UI.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the authoritative storage cluster administrator credentials.
2. From the Dashboard, select the options menu on the top right and select **Configure**.
3. Review the status of storage clusters in the inventory.
4. From the **Storage Clusters** pane, select **Storage Cluster Details** for additional detail.

Edit storage cluster credentials

You can edit the storage cluster's administrator user name and password using the NetApp Hybrid Cloud Control UI.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the authoritative storage cluster administrator credentials.

2. From the Dashboard, select the options menu on the top right and select **Configure**.
3. In the **Storage Clusters** pane, select **Storage Cluster Details**.
4. Select the **Actions** menu for the cluster and select **Edit Cluster Credentials**.
5. Update the storage cluster user name and password.
6. Select **Save**.

Remove a storage cluster

Removing a storage cluster from NetApp Hybrid Cloud Control removes the cluster from the management node inventory. After you remove a storage cluster, the cluster can no longer be managed by HCC and you can access it only by navigating directly to its management IP address.

> You cannot remove the authoritative cluster from the inventory. To determine the authoritative cluster, go to **User Management > Users**. The authoritative cluster is listed next to the heading **Users**.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the authoritative storage cluster administrator credentials.
2. From the Dashboard, select the options menu on the top right and select **Configure**.
3. In the **Storage Clusters** pane, select **Storage Cluster Details**.
4. Select the **Actions** menu for the cluster and select **Remove Storage Cluster**.

   > Selecting **Yes** next removes the cluster from the installation.

5. Select **Yes**.

Enable and disable maintenance mode

If you need to take a storage node offline for maintenance such as software upgrades or host repairs, you can minimize the I/O impact to the rest of the storage cluster by enabling maintenance mode for that node. When you disable maintenance mode, the node is monitored to ensure certain criteria are met before the node can transition out of maintenance mode.

What you'll need

- **Element software**: Your storage cluster version is running NetApp Element software 12.2 or later.
- **Management node**: You have deployed a management node running version 12.2 or later.
- **Management services**: You have updated your management services bundle to version 2.19 or later.
- You have access to log in at the administrator level.

Enable maintenance mode

You can use the following procedure to enable maintenance mode for a storage cluster node.

> Only one node can be in maintenance mode at a time.

Steps

1. Open the IP address of the management node in a web browser. For example:

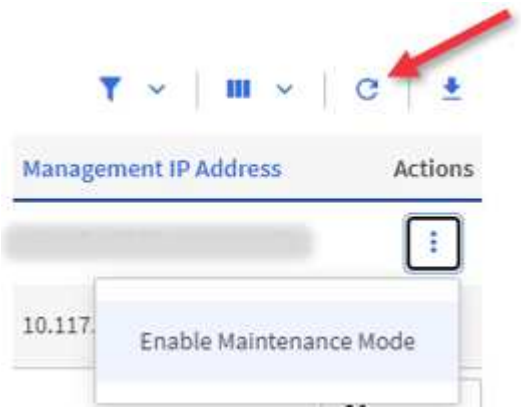
```
https://[management node IP address]
```

2. Log in to NetApp Hybrid Cloud Control by providing the SolidFire all-flash storage cluster administrator credentials.

   The maintenance mode feature options are disabled at the read-only level.

3. In the left navigation blue box, select the SolidFire all-flash installation.
4. In the left navigation pane, select **Nodes**.
5. To view storage inventory information, select **Storage**.
6. Enable maintenance mode on a storage node:

   The storage nodes table is updated automatically every two minutes for non-user initiated actions. Before an action, to ensure that you have the most up-to-date status, you can refresh the nodes table by using the refresh icon located on the upper-right side of the nodes table.



   a. Under **Actions**, select **Enable Maintenance Mode**.

      While **Maintenance Mode** is being enabled, maintenance mode actions are unavailable for the selected node and all other nodes on the same cluster.

      After **Enabling Maintenance Mode** completes, the **Node Status** column displays a wrench icon and the text "**Maintenance Mode**" for the node that is in maintenance mode.

Disable maintenance mode

After a node is successfully placed in maintenance mode, the **Disable Maintenance Mode** action is available for this node. Actions on the other nodes are unavailable until maintenance mode is disabled successfully on the node undergoing maintenance.

Steps

1. For the node under maintenance mode, under **Actions**, select **Disable Maintenance Mode**.

   While **Maintenance Mode** is being disabled, maintenance mode actions are unavailable for the selected

node and all other nodes on the same cluster.

After **Disabling Maintenance Mode** completes, the **Node Status** column displays **Active**.

> When a node is in maintenance mode, it does not accept new data. As a result, it can take longer to disable maintenance mode because the node must sync its data back up before it can exit maintenance mode. The longer you spend in maintenance mode, the longer it can take to disable maintenance mode.

Troubleshoot

If you encounter errors when you are either enabling or disabling maintenance mode, a banner error displays at the top of the nodes table. For more information on the error, you can select the **Show Details** link that is provided on the banner to show what the API returns are.

Find more information

- Create and manage storage cluster assets
- SolidFire and Element Software Documentation

Create and manage user accounts by using NetApp Hybrid Cloud Control

In Element-based storage systems, authoritative cluster users can be created to enable login access to NetApp Hybrid Cloud Control depending on the permissions you want to grant "Administrator" or "Read-only" users. In addition to cluster users, there are also volume accounts, which enable clients to connect to volumes on a storage node.

Manage the following types of accounts:

- Manage authoritative cluster accounts
- Manage volume accounts

Enable LDAP

To use LDAP for any user account, you must first enable LDAP.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, select on the top right Options icon and select **User Management**.
3. From the Users page, select **Configure LDAP**.
4. Define your LDAP configuration.
5. Select the authentication type of Search and Bind or Direct Bind.
6. Before you save the changes, select **Test LDAP Log In** at the top of the page, enter the user name and password of a user you know exists, and select **Test**.
7. Select **Save**.

Manage authoritative cluster accounts

Authoritative user accounts are managed from the top right menu User Management option in NetApp Hybrid Cloud Control. These types of accounts enable you to authenticate against any storage asset associated with a NetApp Hybrid Cloud Control instance of nodes and clusters. With this account, you can manage volumes, accounts, access groups, and more across all clusters.

Create an authoritative cluster account

You can create an account by using NetApp Hybrid Cloud Control.

This account can be used to log in to the Hybrid Cloud Control, the per-node UI for the cluster, and the storage cluster in NetApp Element software.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, select on the top right Options icon and select **User Management**.
3. Select **Create User**.
4. Select the authentication type of cluster or LDAP.
5. Complete one of the following:
   - If you selected LDAP, enter the DN.

     To use LDAP, you must first enable LDAP or LDAPS. See Enable LDAP.

   - If you selected Cluster as the Auth Type, enter a name and password for the new account.
6. Select either Administrator or Read-only permissions.

   To view the permissions from NetApp Element software, select **Show legacy permissions**. If you select a subset of these permissions, the account is assigned Read-only permissions. If you select all legacy permissions, the account is assigned Administrator permissions.

   To ensure that all children of a group inherit permissions, create a DN organization admin group in the LDAP server. All the children accounts of that group will inherit those permissions.

7. Check the box indicating that "I have read and accept the NetApp End User License Agreement."
8. Select **Create User**.

Edit an authoritative cluster account

You can change the permissions or password on a user account by using NetApp Hybrid Cloud Control.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, select on the icon in the top right and select **User Management**.
3. Optionally filter the list of user accounts by selecting **Cluster**, **LDAP**, or **Idp**.

   If you configured users on the storage cluster with LDAP, those accounts show a User Type of "LDAP." If

you configured users on the storage cluster with Idp, those accounts show a User Type of "Idp."

4. In the **Actions** column in the table, expand the menu for the account and select **Edit**.
5. Make changes as needed.
6. Select **Save**.
7. Log out of NetApp Hybrid Cloud Control.

> It might take the NetApp Hybrid Cloud Control UI up to 2 minutes to refresh the inventory. To manually refresh inventory, access the REST API UI inventory service `https://[management node IP]/inventory/1/` and run `GET /installations/{id}` for the cluster.

8. Log into NetApp Hybrid Cloud Control.

Delete an authoritative user account

You can delete one or more accounts when it is no longer needed. You can delete an LDAP user account.

You cannot delete the primary administrator user account for the authoritative cluster.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, select on the icon in the top right and select **User Management**.
3. In the **Actions** column in the Users table, expand the menu for the account and select **Delete**.
4. Confirm the deletion by selecting **Yes**.

Manage volume accounts

Volume accounts are managed within the NetApp Hybrid Cloud Control Volumes table. These accounts are specific only to the storage cluster on which they were created. These types of accounts enable you to set permissions on volumes across the network, but have no effect outside of those volumes.

A volume account contains the CHAP authentication required to access the volumes assigned to it.

Create a volume account

Create an account specific to this volume.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, select **Storage** > **Volumes**.
3. Select the **Accounts** tab.
4. Select the **Create Account** button.
5. Enter a name for the new account.
6. In the CHAP Settings section, enter the following information:
   - Initiator Secret for CHAP node session authentication
   - Target Secret for CHAP node session authentication

To auto-generate either password, leave the credential fields blank.

7. Select **Create Account**.

Edit a volume account

You can change the CHAP info and change whether an account is active or locked.

Deleting or locking an account associated with the management node results in an inaccessible management node.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, select **Storage** > **Volumes**.
3. Select the **Accounts** tab.
4. In the **Actions** column in the table, expand the menu for the account and select **Edit**.
5. Make changes as needed.
6. Confirm the changes by selecting **Yes**.

Delete a volume account

Delete an account that you no longer need.

Before you delete a volume account, delete and purge any volumes associated with the account first.

Deleting or locking an account associated with the management node results in an inaccessible management node.

Persistent volumes that are associated with management services are assigned to a new account during installation or upgrade. If you are using persistent volumes, do not modify or delete the volumes or their associated account. If you do delete these accounts, you could render your management node unusable.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, select **Storage** > **Volumes**.
3. Select the **Accounts** tab.
4. In the **Actions** column in the table, expand the menu for the account and select **Delete**.
5. Confirm the deletion by selecting **Yes**.

Find more information

- Learn about accounts
- Work with accounts using CHAP
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Create and manage volumes by using NetApp Hybrid Cloud Control

You can create a volume and associate the volume with a given account. Associating a volume with an account gives the account access to the volume through the iSCSI initiators and CHAP credentials.

You can specify QoS settings for a volume during creation.

You can manage volumes in NetApp Hybrid Cloud Control in the following ways:

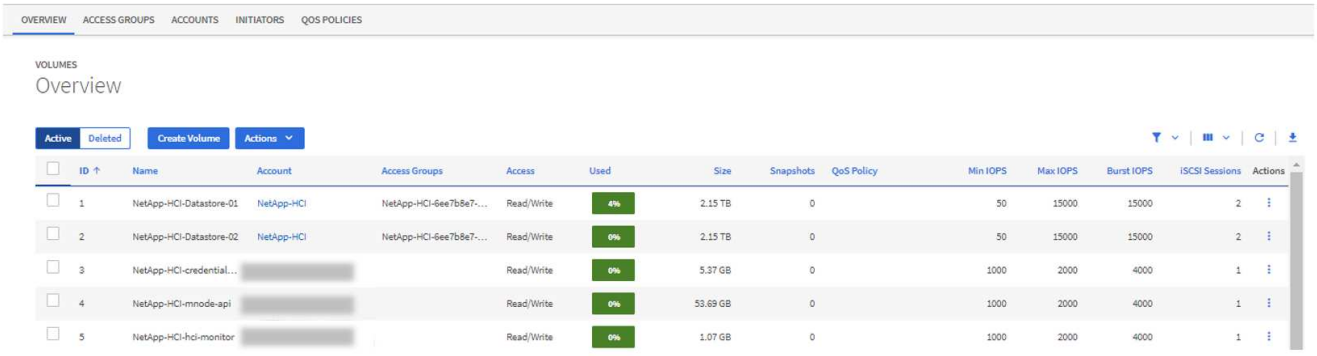
- Create a volume
- Apply a QoS policy to a volume
- Edit a volume
- Clone volumes
- Add volumes to a volume access group
- Delete a volume
- Restore a deleted volume
- Purge a deleted volume

Create a volume

You can create a storage volume using NetApp Hybrid Cloud Control.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes** > **Overview** tab.



4. Select **Create Volume**.
5. Enter a name for the new volume.
6. Enter the total size of the volume.

The default volume size selection is in GB. You can create volumes using sizes measured in GB or GiB:
1GB = 1 000 000 000 bytes
1GiB = 1 073 741 824 bytes

7. Select a block size for the volume.
8. From the **Account** list, select the account that should have access to the volume.

   If an account does not exist, select **Create New Account**, enter a new account name, and select **Create Account**. The account is created and associated with the new volume in the **Account** list.

   If there are more than 50 accounts, the list does not appear. Begin typing and the auto-complete feature displays values for you to choose.

9. To configure the Quality of Service for the volume, do one of the following:
   - Under **Quality of Service Settings**, set customized minimum, maximum, and burst values for IOPS or use the default QoS values.
   - Select an existing QoS policy by enabling the **Assign Quality of Service Policy** toggle and choosing an existing QoS policy from the resulting list.
   - Create and assign a new QoS policy by enabling the **Assign Quality of Service Policy** toggle and selecting **Create New QoS Policy**. In the resulting window, enter a name for the QoS policy and then enter QoS values. When finished, select **Create Quality of Service Policy**.

   Volumes that have a Max or Burst IOPS value greater than 20,000 IOPS might require high queue depth or multiple sessions to achieve this level of IOPS on a single volume.

10. Select **Create Volume**.

Apply a QoS policy to a volume

You can apply a QoS policy to existing storage volumes by using NetApp Hybrid Cloud Control. If instead you need to set custom QoS values for a volume, you can Edit a volume. To create a new QoS policy, see Create and manage volume QoS policies.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes** > **Overview**.
4. Select one or more volumes to associate with a QoS policy.
5. Select the **Actions** drop-down list at the top of the volumes table, and select **Apply QoS Policy**.
6. In the resulting window, select a QoS policy from the list and select **Apply QoS Policy**.

   If you are using QoS policies on a volume, you can set custom QoS to remove the QoS policy affiliation with the volume. Custom QoS values override QoS policy values for volume QoS settings.

Edit a volume

Using NetApp Hybrid Cloud Control, you can edit volume attributes such as QoS values, volume size, and the unit of measurement by which byte values are calculated. You can also modify account access for replication usage or to restrict access to the volume.

About this task

You can resize a volume when there is sufficient space on the cluster under the following conditions:

- Normal operating conditions.
- Volume errors or failures are being reported.
- The volume is being cloned.
- The volume is being resynced.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes** > **Overview**.
4. In the **Actions** column in the volumes table, expand the menu for the volume and select **Edit**.
5. Make changes as needed:
   a. Change the total size of the volume.

      You can increase, but not decrease, the size of the volume. You can only resize one volume in a single resizing operation. Garbage collection operations and software upgrades do not interrupt the resizing operation.

      If you are adjusting volume size for replication, first increase the size of the volume assigned as the replication target. Then you can resize the source volume. The target volume can be greater or equal in size to the source volume, but it cannot be smaller.

      The default volume size selection is in GB. You can create volumes using sizes measured in GB or GiB:
      1GB = 1 000 000 000 bytes
      1GiB = 1 073 741 824 bytes

   b. Select a different account access level:
      - Read Only
      - Read/Write
      - Locked
      - Replication Target
   c. Select the account that should have access to the volume.

      Begin typing and the auto-complete function displays possible values for you to choose.

      If an account does not exist, select **Create New Account**, enter a new account name, and select **Create**. The account is created and associated with the existing volume.

d. Change the Quality of Service by doing one of the following:

   i. Select an existing policy.

   ii. Under Custom Settings, set the minimum, maximum, and burst values for IOPS or use the default values.

> If you are using QoS policies on a volume, you can set custom QoS to remove the QoS policy affiliation with the volume. Custom QoS will override QoS policy values for volume QoS settings.

> When you change IOPS values, you should increment in tens or hundreds. Input values require valid whole numbers. Configure volumes with an extremely high burst value. This enables the system to process occasional large block, sequential workloads more quickly, while still constraining the sustained IOPS for a volume.

6. Select **Save**.

Clone volumes

You can create a clone of a single storage volume or clone a group of volumes to make a point-in-time copy of the data. When you clone a volume, the system creates a snapshot of the volume and then creates a copy of the data referenced by the snapshot.

Before you begin

- At least one cluster must be added and running.
- At least one volume has been created.
- A user account has been created.
- Available unprovisioned space must be equal to or more than the volume size.

About this task

The cluster supports up to two running clone requests per volume at a time and up to 8 active volume clone operations at a time. Requests beyond these limits are queued for later processing.

Volume cloning is an asynchronous process, and the amount of time the process requires depends on the size of the volume you are cloning and the current cluster load.

> Cloned volumes do not inherit volume access group membership from the source volume.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select the **Volumes** > **Overview** tab.
4. Select each volume you want to clone.
5. Select the **Actions** drop-down list at the top of the volumes table, and select **Clone**.
6. In the resulting window, do the following:
   a. Enter a volume name prefix (this is optional).
   b. Choose the access type from the **Access** list.

c. Choose an account to associate with the new volume clone (by default, **Copy from Volume** is selected, which will use the same account that the original volume uses).

d. If an account does not exist, select **Create New Account**, enter a new account name, and select **Create Account**. The account is created and associated with the volume.

Use descriptive naming best practices. This is especially important if multiple clusters or vCenter Servers are used in your environment.

Increasing the volume size of a clone results in a new volume with additional free space at the end of the volume. Depending on how you use the volume, you may need to extend partitions or create new partitions in the free space to make use of it.

e. Select **Clone Volumes**.

The time to complete a cloning operation is affected by volume size and current cluster load. Refresh the page if the cloned volume does not appear in the volume list.

Add volumes to a volume access group

You can add a single volume or a group of volumes to a volume access group.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes** > **Overview**.
4. Select one or more volumes to associate with a volume access group.
5. Select the **Actions** drop-down list at the top of the volumes table, and select **Add to Access Group**.
6. In the resulting window, select a volume access group from the **Volume Access Group** list.
7. Select **Add Volume**.

Delete a volume

You can delete one or more volumes from an Element storage cluster.

About this task

The system does not immediately purge deleted volumes; they remain available for approximately eight hours. After eight hours, they are purged and no longer available. If you restore a volume before the system purges it, the volume comes back online and iSCSI connections are restored.

If a volume used to create a snapshot is deleted, its associated snapshots become inactive. When the deleted source volumes are purged, the associated inactive snapshots are also removed from the system.

Persistent volumes that are associated with management services are created and assigned to a new account during installation or upgrade. If you are using persistent volumes, do not modify or delete the volumes or their associated account. If you do delete these volumes, you could render your management node unusable.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes** > **Overview**.
4. Select one or more volumes to delete.
5. Select the **Actions** drop-down list at the top of the volumes table, and select **Delete**.
6. In the resulting window, confirm the action by selecting **Yes**.

Restore a deleted volume

After a storage volume is deleted, you can still restore it if you do so before eight hours after deletion.

The system does not immediately purge deleted volumes; they remain available for approximately eight hours. After eight hours, they are purged and no longer available. If you restore a volume before the system purges it, the volume comes back online and iSCSI connections are restored.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes** > **Overview**.
4. Select **Deleted**.
5. In the **Actions** column of the Volumes table, expand the menu for the volume and select **Restore**.
6. Confirm the process by selecting **Yes**.

Purge a deleted volume

After storage volumes are deleted, they remain available for approximately eight hours. After eight hours, they are purged automatically and no longer available. If you do not want to wait for the eight hours, you can delete

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes** > **Overview**.
4. Select **Deleted**.
5. Select one or more volumes to purge.
6. Do one of the following:
   - If you selected multiple volumes, select the **Purge** quick filter at the top of the table.
   - If you selected a single volume, in the **Actions** column of the Volumes table, expand the menu for the volume and select **Purge**.
7. In the **Actions** column of the Volumes table, expand the menu for the volume and select **Purge**.
8. Confirm the process by selecting **Yes**.

Find more information

- Learn about volumes

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Create and manage volume access groups

You can create new volume access groups, make changes to the name, associated initiators, or associated volumes of access groups, or delete existing volume access groups using NetApp Hybrid Cloud Control.

What you'll need

- You have administrator credentials for this SolidFire all-flash storage system.
- You have upgraded your management services to at least version 2.15.28. NetApp Hybrid Cloud Control storage management is not available in earlier service bundle versions.
- Ensure you have a logical naming scheme for volume access groups.

Add a volume access group

You can add a volume access group to a storage cluster by using NetApp Hybrid Cloud Control.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes**.
4. Select the **Access Groups** tab.
5. Select the **Create Access Group** button.
6. In the resulting dialog, enter a name for the new volume access group.
7. (Optional) In the **Initiators** section, select one or more initiators to associate with the new volume access group.

   If you associate an initiator with the volume access group, that initiator can access each volume in the group without the need for authentication.

8. (Optional) In the **Volumes** section, select one or more volumes to include in this volume access group.
9. Select **Create Access Group**.

Edit a volume access group

You can edit the properties of an existing volume access group by using NetApp Hybrid Cloud Control. You can make changes to the name, associated initiators, or associated volumes of an access group.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes**.

4. Select the **Access Groups** tab.
5. In the **Actions** column of the table of access groups, expand the options menu for the access group you need to edit.
6. In the options menu, select **Edit**.
7. Make any needed changes to the name, associated initiators, or associated volumes.
8. Confirm your changes by selecting **Save**.
9. In the **Access Groups** table, verify that the access group reflects your changes.

Delete a volume access group

You can remove a volume access group by using NetApp Hybrid Cloud Control, and at the same time remove the initiators associated with this access group from the system.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes**.
4. Select the **Access Groups** tab.
5. In the **Actions** column of the table of access groups, expand the options menu for the access group you need to delete.
6. In the options menu, select **Delete**.
7. If you do not wish to delete the initiators that are associated with the access group, deselect the **Delete initiators in this access group** checkbox.
8. Confirm the delete operation by selecting **Yes**.

Find more information

- Learn about volume access groups
- Add initiator to a volume access group
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Create and manage initiators

You can use initiators for CHAP-based rather than account-based access to volumes. You can create and delete initiators, and give them friendly aliases to simplify administration and volume access. When you add an initiator to a volume access group, that initiator enables access to all volumes in the group.

What you'll need

- You have cluster administrator credentials.
- You have upgraded your management services to at least version 2.17. NetApp Hybrid Cloud Control initiator management is not available in earlier service bundle versions.

Options

- Create an initiator
- Add initiators to a volume access group
- Change an initiator alias
- Delete initiators

Create an initiator

You can create iSCSI or Fibre Channel initiators and optionally assign them aliases.

About this task

The accepted format of an initiator IQN is `iqn.yyyy-mm` where y and m are digits followed by text which must only contain digits, lower-case alphabetic characters, a period (`.`), colon (`:`) or dash (`-`).
A sample of the format is as follows:

```
iqn.2010-01.com.solidfire:c2r9.fc0.2100000e1e09bb8b
```

The accepted format of a Fibre Channel initiator WWPN is `:Aa:bB:CC:dd:11:22:33:44` or `AabBCCdd11223344`.
A sample of the format is as follows:

```
5f:47:ac:c0:5c:74:d4:02
```

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes**.
4. Select the **Initiators** tab.
5. Select the **Create Initiators** button.

Option	Steps
Create one or more initiators	a. Enter the IQN or WWPN for the initiator in the **IQN/WWPN** field. b. Enter a friendly name for the initiator in the **Alias** field. c. (Optional) Select **Add Initiator** to open new initiator fields or use the bulk create option instead. d. Select **Create Initiators**.

Option	Steps
Bulk create initiators	a. Select **Bulk Add IQNs/WWPNs**. b. Enter a list of IQNs or WWPNs in the text box. Each IQN or WWPN must be comma or space separated or on its own line. c. Select **Add IQNs/WWPNs**. d. (Optional) Add unique aliases to each initiator. e. Remove any initiator from the list that might already exist in the installation. f. Select **Create Initiators**.

Add initiators to a volume access group

You can add initiators to an volume access group. When you add an initiator to a volume access group, the initiator enables access to all volumes in that volume access group.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes**.
4. Select the **Initiators** tab.
5. Select one or more initiators you want to add.
6. Select **Actions > Add to Access Group**.
7. Select the access group.
8. Confirm your changes by selecting **Add Initiator**.

Change an initiator alias

You can change the alias of an existing initiator or add an alias if one does not already exist.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes**.
4. Select the **Initiators** tab.
5. In the **Actions** column, expand the options menu for the initiator.
6. Select **Edit**.
7. Make any needed changes to the alias or add a new alias.
8. Select **Save**.

Delete initiators

You can delete one or more initiators. When you delete an initiator, the system removes it from any associated volume access group. Any connections using the initiator remain valid until the connection is reset.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the name of your storage cluster on the left navigation menu.
3. Select **Volumes**.
4. Select the **Initiators** tab.
5. Delete one or more initiators:
   a. Select one or more initiators you want to delete.
   b. Select **Actions > Delete**.
   c. Confirm the delete operation and select **Yes**.

Find more information

- Learn about initiators
- Learn about volume access groups
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Create and manage volume QoS policies

A QoS (Quality of Service) policy enables you to create and save a standardized quality of service setting that can be applied to many volumes. The selected cluster must be Element 10.0 or later to use QoS policies; otherwise, QoS policy functions are not available.

See SolidFire all-flash storage Concepts content for more information about using QoS policies instead of individual volume QoS.

Using NetApp Hybrid Cloud Control, you can create and manage QoS policies by completing the following tasks:

- Create a QoS policy
- Apply a QoS policy to a volume
- Change the QoS policy assignment of a volume
- Edit a QoS policy
- Delete a QoS policy

Create a QoS policy

You can create QoS policies and apply them to volumes that should have equivalent performance.

If you are using QoS policies, do not use custom QoS on a volume. Custom QoS will override and adjust QoS policy values for volume QoS settings.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the menu for your storage cluster.
3. Select **Storage > Volumes**.
4. Select the **QoS Policies** tab.
5. Select **Create Policy**.
6. Enter the **Policy Name**.

   Use descriptive naming best practices. This is especially important if multiple clusters or vCenter Servers are used in your environment.

7. Enter the minimum IOPS, maximum IOPS, and burst IOPS values.
8. Select **Create QoS Policy**.

   A system ID is generated for the policy and the policy appears on the QoS Policies page with its assigned QoS values.

Apply a QoS policy to a volume

You can assign an existing QoS policy to a volume using NetApp Hybrid Cloud Control.

What you'll need

The QoS policy you want to assign has been created.

About this task

This task describes how to assign a QoS policy to an individual volume by changing its settings. The latest version of NetApp Hybrid Cloud Control does not have a bulk assign option for more than one volume. Until the functionality to bulk assign is provided in a future release, you can use the Element web UI or vCenter Plug-in UI to bulk assign QoS policies.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the menu for your storage cluster.
3. Select **Storage > Volumes**.
4. Select the **Actions** menu next to the volume you intend to modify.
5. In the resulting menu, select **Edit**.
6. In the dialog box, enable **Assign QoS Policy** and select the QoS policy from the drop-down list to apply to the selected volume.

   Assigning QoS will override any individual volume QoS values that have been previously applied.

7. Select **Save**.

Change the QoS policy assignment of a volume

You can remove the assignment of a QoS policy from a volume or select a different QoS policy or custom QoS.

What you'll need

The volume you want to modify is assigned a QoS policy.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the menu for your storage cluster.
3. Select **Storage > Volumes**.
4. Select the **Actions** menu next to the volume you intend to modify.
5. In the resulting menu, select **Edit**.
6. In the dialog box, do one of the following:
   - Disable **Assign QoS Policy** and modify the **Min IOPS**, **Max IOPS**, and **Burst IOPS** values for individual volume QoS.

     When QoS policies are disabled, the volume uses default QoS IOPS values unless otherwise modified.

   - Select a different QoS policy from the drop-down list to apply to the selected volume.
7. Select **Save**.

Edit a QoS policy

You can change the name of an existing QoS policy or edit the values associated with the policy. Changing QoS policy performance values affects QoS for all volumes associated with the policy.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the menu for your storage cluster.
3. Select **Storage > Volumes**.
4. Select the **QoS Policies** tab.
5. Select the **Actions** menu next to the QoS policy you intend to modify.
6. Select **Edit**.
7. In the **Edit QoS Policy** dialog box, change one or more of the following:
   - **Name**: The user-defined name for the QoS policy.
   - **Min IOPS**: The minimum number of IOPS guaranteed for the volume. Default = 50.
   - **Max IOPS**: The maximum number of IOPS allowed for the volume. Default = 15,000.
   - **Burst IOPS**: The maximum number of IOPS allowed over a short period of time for the volume. Default = 15,000.
8. Select **Save**.

You can select on the link in the **Active Volumes** column for a policy to show a filtered list of the volumes assigned to that policy.

Delete a QoS policy

You can delete a QoS policy if it is no longer needed. When you delete a QoS policy, all volumes assigned with the policy maintain the QoS values previously defined by the policy but as individual volume QoS. Any association with the deleted QoS policy is removed.

Steps

1. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
2. From the Dashboard, expand the menu for your storage cluster.
3. Select **Storage > Volumes**.
4. Select the **QoS Policies** tab.
5. Select the **Actions** menu next to the QoS policy you intend to modify.
6. Select **Delete**.
7. Confirm the action.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Monitor your SolidFire system with NetApp Hybrid Cloud Control

Monitor storage resources on the Hybrid Cloud Control Dashboard

With the NetApp Hybrid Cloud Control Dashboard, you can view all your storage resources at a glance. Additionally, you can monitor storage capacity and storage performance.

When you launch a new NetApp Hybrid Cloud Control session for the first time, there might be a delay with loading the NetApp Hybrid Cloud Control Dashboard view when the management node is managing many clusters. The loading time varies depending on the number of clusters being actively managed by the management node. For subsequent launches, you will experience faster loading times.

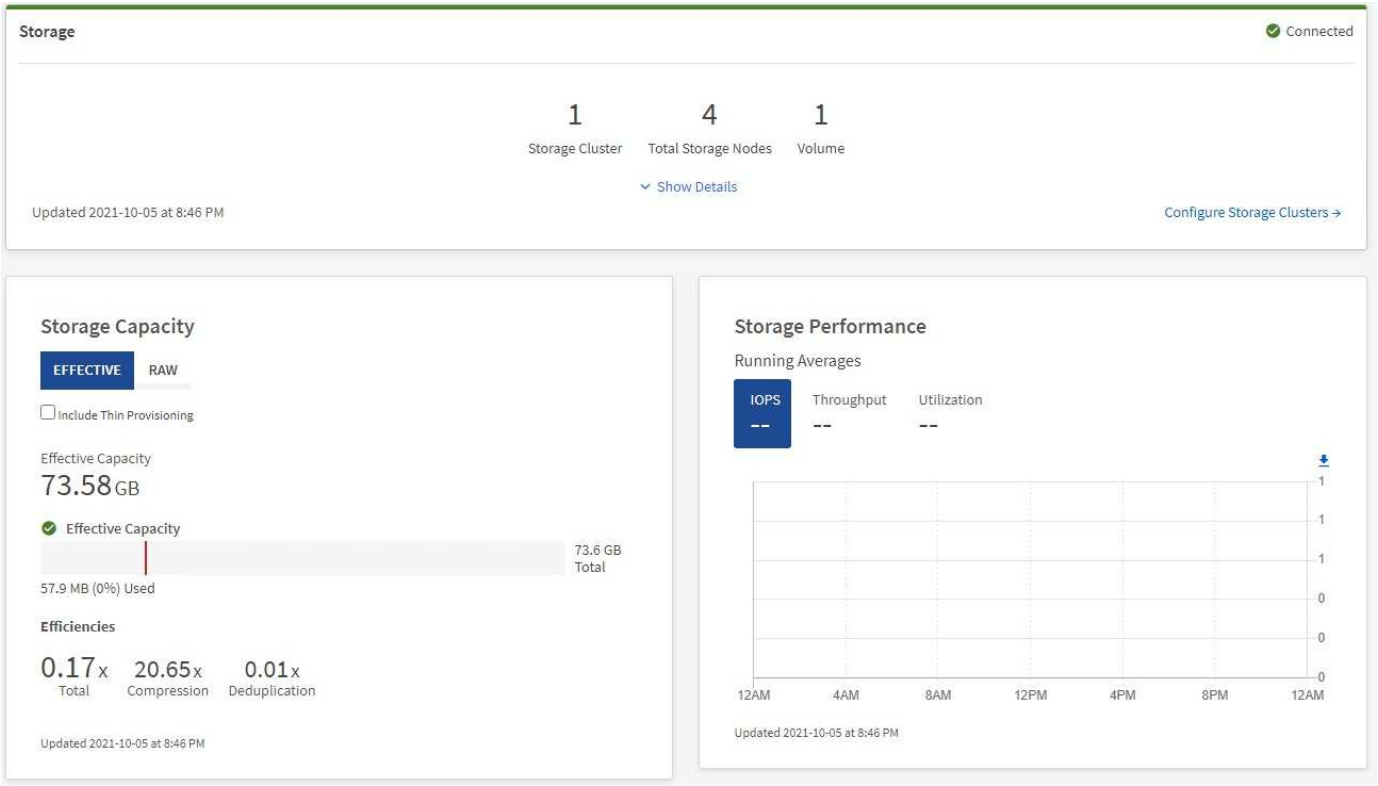
- Access the NetApp HCC Dashboard
- Monitor storage resources
- Monitor storage capacity
- Monitor storage performance

Access the NetApp HCC Dashboard

1. Open the IP address of the management node in a web browser. For example:

```
https://[management node IP address]
```

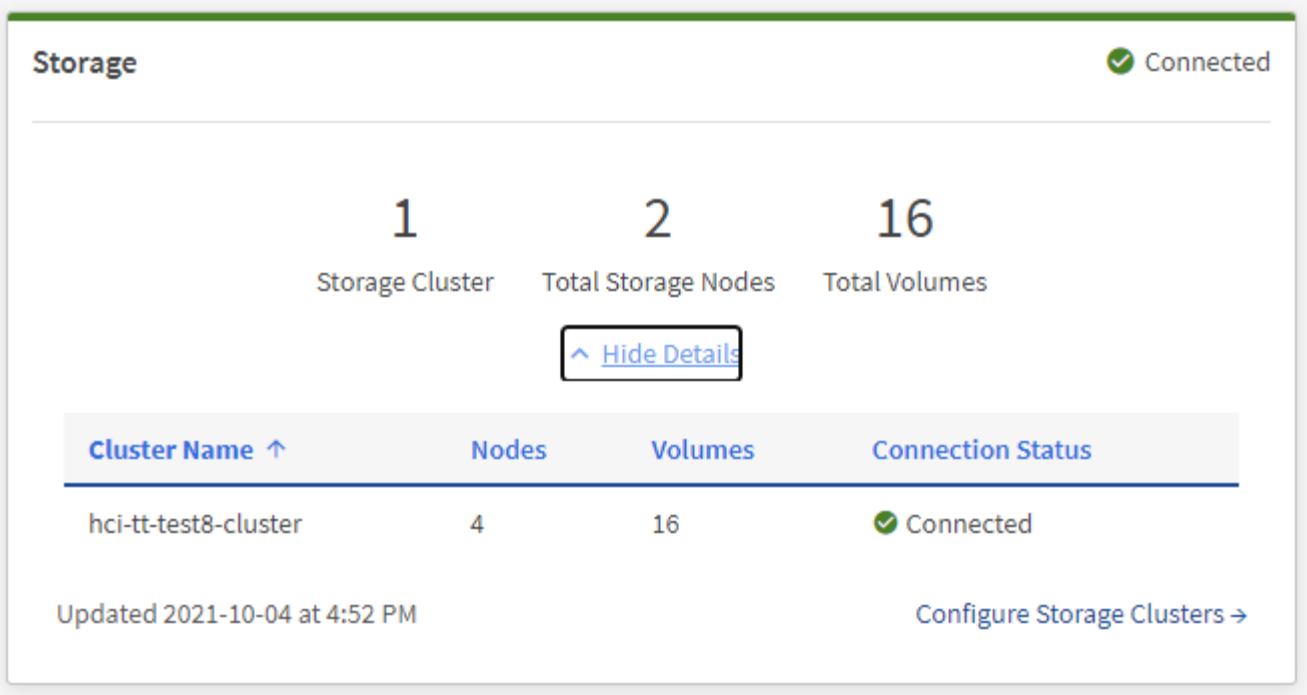
2. Log in to NetApp Hybrid Cloud Control by providing the SolidFire all-flash storage cluster administrator credentials.
3. View the Hybrid Cloud Control Dashboard.



Monitor storage resources

Use the **Storage** pane to see your total storage environment. You can monitor the number of storage clusters, storage nodes, and total volumes.

To see details, in the Storage pane, select **Show Details**.



The Total Storage Nodes number does not include Witness Nodes from two-node storage clusters. The Witness Nodes are included in the Nodes number in the details section for that cluster.

To see the most recent storage cluster data, use the Storage Clusters page, where polling occurs more frequently than on the Dashboard.

Monitor storage capacity

Monitoring the storage capacity of your environment is critical. Using the Storage Capacity pane, you can determine your storage capacity efficiency gains with or without compression, deduplication, and thin provisioning features enabled.

You can see the total physical storage space available in your cluster on the **RAW** tab, and information about the provisioned storage on the **EFFECTIVE** tab.

Steps

1. Select the **RAW** tab, to see the total physical storage space used and available in your cluster.

   Look at the vertical lines to determine whether your used capacity is less than the total or less than Warning, Error, or Critical thresholds. Hover over the lines to see details.

   You can set the threshold for Warning, which defaults to 3% below the Error threshold. The Error and Critical thresholds are preset and not configurable by design. The Error threshold indicates that less than one node of capacity remains in the cluster. For steps on setting the threshold, see Setting cluster full threshold.

   For details about the related cluster thresholds Element API, see "getClusterFullThreshold" in the *Element software API documentation*. To view details about block and metadata capacity, see Understanding cluster fullness levels in the *Element software documentation*.

2. Select the **EFFECTIVE** tab, to see information about total storage provisioned to connected hosts and to see efficiency ratings.
   a. Optionally, check **Include Thin Provisioning** to see thin provisioning efficiency rates in the Effective Capacity bar chart.
   b. **Effective Capacity bar chart**: Look at the vertical lines to determine whether your used capacity is less than the total or less than Warning, Error, or Critical thresholds. Similar to the Raw tab, you can hover over the vertical lines to see details.
   c. **Efficiencies**: Look at these ratings to determine your storage capacity efficiency gains with compression, deduplication, and thin provisioning features enabled. For example, if compression shows as "1.3x", this means that storage efficiency with compression enabled is 1.3 times more efficient than without it.

      Total Efficiencies equals (maxUsedSpace * efficiency factor) / 2, where efficiencyFactor = (thinProvisioningFactor * deDuplicationFactor * compressionFactor). When Thin Provisioning is unchecked, it is not included in the Total Efficiency.

   d. If the effective storage capacity nears an Error or Critical threshold, consider clearing the data on your system.
3. For further analysis and historical context, look at NetApp SolidFire Active IQ details.

Monitor storage performance

You can look at how much IOPS or throughput you can get out of a cluster without surpassing the useful performance of that resource by using the Storage Performance pane. Storage performance is the point at which you get the maximum utilization before latency becomes an issue.

The Storage Performance pane helps you identify whether the performance is reaching the point where the performance might degrade if the workloads increase.

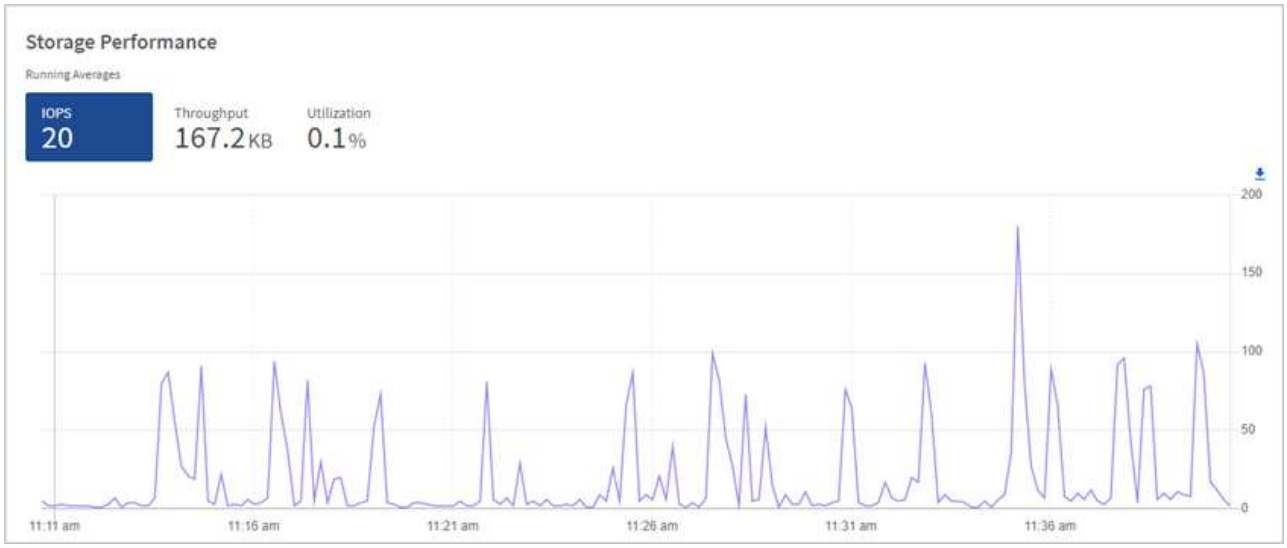
The information on this pane refreshes every 10 seconds and shows an average of all the points on the graph.

For details about the associated Element API method, see the GetClusterStats method in the *Element software API documentation*.

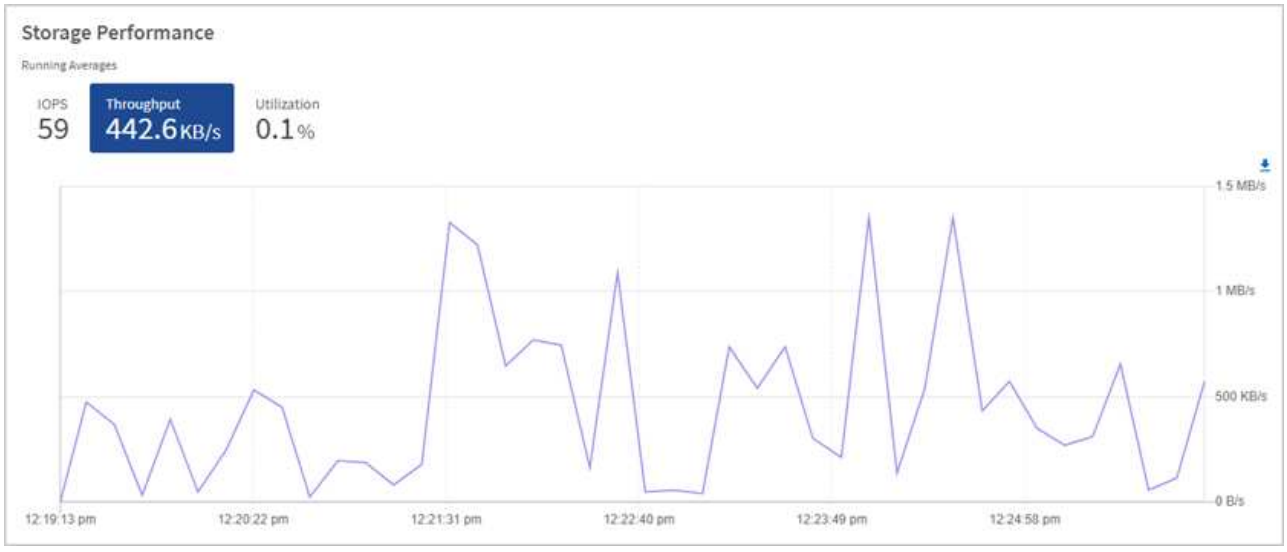
Steps

1. View the Storage Performance pane. For details, hover over points in the graph.

a. **IOPS** tab: See the current operations per second. Look for trends in data or spikes. For example, if you see that the maximum IOPS is 160K and 100K of that is free or available IOPS, you might consider adding more workloads to this cluster. On the other hand, if you see that only 140K is available, you might consider offloading workloads or expanding your system.



b. **Throughput** tab: Monitor patterns or spikes in throughput. Also monitor for continuously high throughput values, which might indicate that you are nearing the maximum useful performance of the resource.



c. **Utilization** tab: Monitor the utilization of IOPS in relation to the total IOPS available summed up at the cluster level.

2. For further analysis, look at storage performance by using the NetApp Element Plug-in for vCenter Server.

   Performance shown in the NetApp Element Plug-in for vCenter Server.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

View your inventory on the Nodes page

You can view your storage assets in your system and determine their IP addresses, names, and software versions.

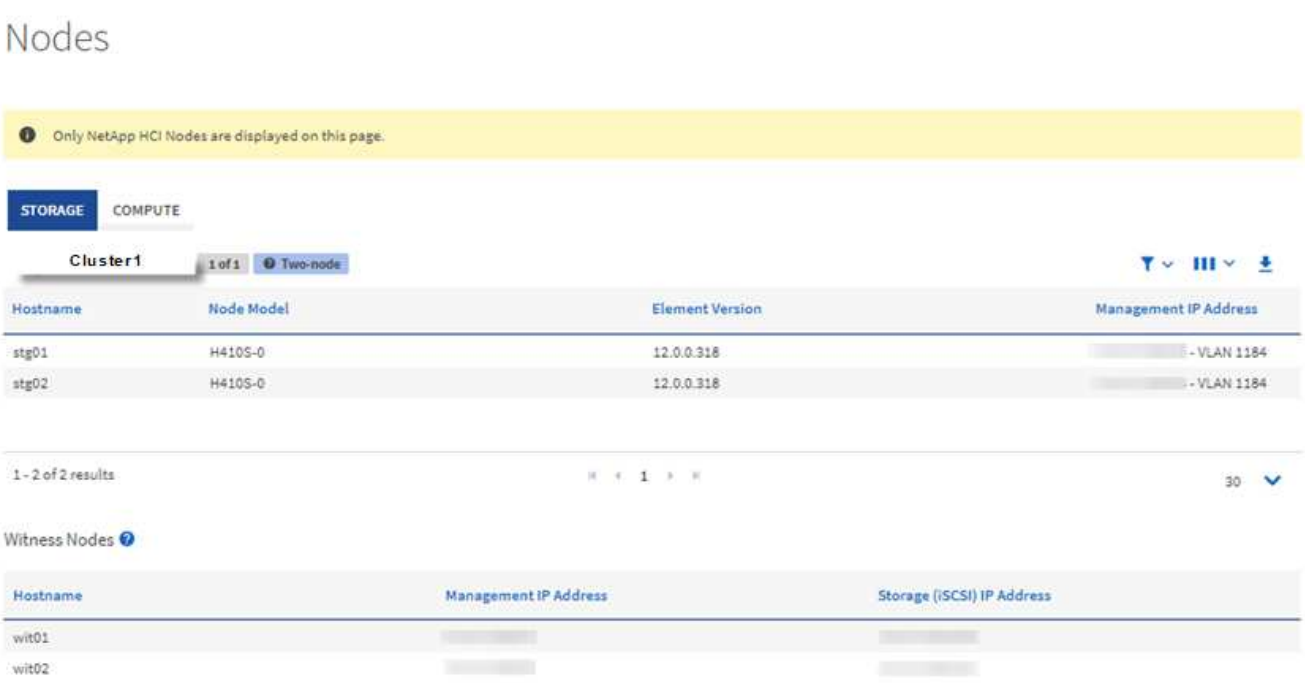
You can view storage information for your multiple node systems. If custom protection domains are assigned, you can see which protection domains are assigned to specific nodes.

Steps

1. Open the IP address of the management node in a web browser. For example:

   ```
   https://[management node IP address]
   ```

2. Log in to NetApp Hybrid Cloud Control by providing the SolidFire all-flash storage cluster administrator credentials.
3. In the left navigation, select **Nodes**.



When you launch a new NetApp Hybrid Cloud Control session for the first time, there might be a delay with loading the NetApp Hybrid Cloud Control Nodes page when the management node is managing many clusters. The loading time varies depending on the number of clusters being actively managed by the management node. For subsequent launches, you will experience faster loading times.

4. On the **Storage** tab of the Nodes page, review the following information:
   a. Two-node clusters: A “two-node” label appears on the Storage tab and the associated Witness Nodes are listed.
   b. Three-node clusters: The storage nodes and associated Witness Nodes are listed. Three-node clusters have a Witness Node deployed on standby to maintain high availability in the case of node failure.
   c. Clusters with four nodes or more: Information for clusters with four or more nodes appears. Witness Nodes do not apply. If you started with two or three storage nodes and added more nodes, the Witness Nodes still appear. Otherwise, the Witness Nodes table does not appear.
   d. The firmware bundle version: Starting with management services version 2.14, if you have clusters running Element 12.0 or later, you can see the firmware bundle version for these clusters. If the nodes in a cluster have different firmware versions on them, you can see **Multiple** in the **Firmware Bundle Version** column.
   e. Custom protection domains: If custom protection domains are in use on the cluster, you can see custom protection domain assignments for each node in the cluster. If custom protection domains are not enabled, this column does not appear.
5. You can manipulate the information on these pages in several ways:
   a. To filter the list of items in the results, select the **Filter** icon and select the filters. You can also enter text for the filter.
   b. To show or hide columns, select the **Show/Hide Columns** icon.
   c. To download the table, select the **Download** icon.

To view the number of storage, look at the NetApp Hybrid Cloud Control (HCC) Dashboard. See Monitor storage resources with the HCC Dashboard.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Monitor volumes on your storage cluster

The SolidFire system provisions storage using volumes. Volumes are block devices accessed over the network by iSCSI or Fibre Channel clients. You can monitor details about access groups, accounts, initiators, used capacity, Snapshot data protection status, number of iSCSI sessions, and the Quality of Service (QoS) policy associated with the volume.

You can also see details on active and deleted volumes.

With this view, you might first want to monitor the Used capacity column.

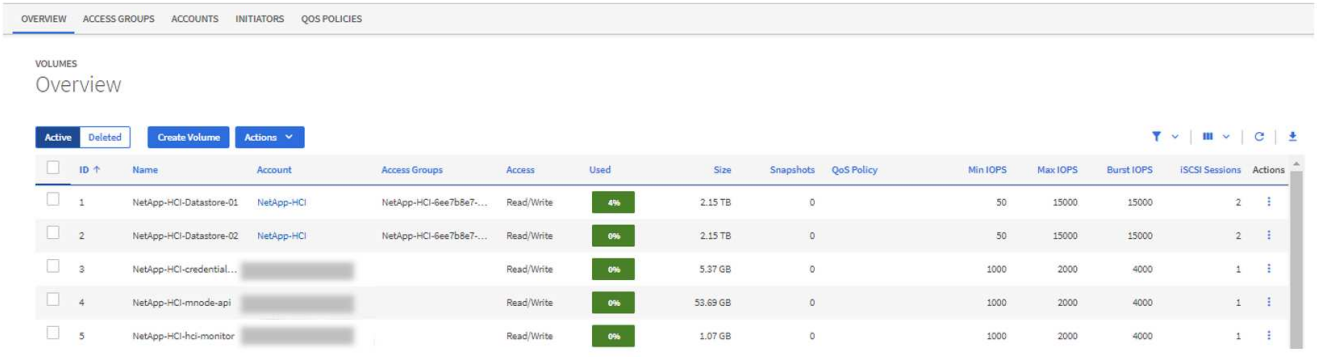
You can access this information only if you have NetApp Hybrid Cloud Control administrative privileges.

Steps

1. Open the IP address of the management node in a web browser. For example:

```
https://[management node IP address]
```

2. Log in to NetApp Hybrid Cloud Control by providing the SolidFire all-flash storage cluster administrator credentials.
3. In the left navigation blue box, select the SolidFire all-flash storage installation.
4. In the left navigation, select the cluster and select **Storage** > **Volumes**.



5. On the Volumes page, use the following options:

a. Filter the results by selecting the **Filter** icon.

b. Hide or show columns by selecting the **Hide/Show** icon.

c. Refresh data by selecting the **Refresh** icon.

d. Download a CSV file by selecting on the **Download** icon.

6. Monitor the Used capacity column. If Warning, Error, or Critical thresholds are reached, the color represents the used capacity status:

   a. Warning - Yellow

   b. Error - Orange

   c. Critical - Red

7. From the Volumes view, select the tabs to see additional details about the volumes:

   a. **Access Groups**: You can see the volume access groups that are mapped from initiators to a collection of volumes for secured access.

      See information about volume access groups.

   b. **Accounts**: You can see the user accounts, which enable clients to connect to volumes on a node. When you create a volume, it is assigned to a specific user account.

      See information about SolidFire all-flash storage system user accounts.

   c. **Initiators**: You can see the iSCSI initiator IQN or Fibre Channel WWPNs for the volume. Each IQN added to an access group can access each volume in the group without requiring CHAP authentication. Each WWPN added to an access group enables Fibre Channel network access to the volumes in the access group.

   d. **QoS Policies**: You can see the QoS policy applied to the volume. A QoS policy applies standardized settings for minimum IOPS, maximum IOPS, and burst IOPS to multiple volumes.

      See information about performance and QoS policies.

Find more information

- SolidFire and Element documentation
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Collect logs for troubleshooting

If you have trouble with your SolidFire all-flash storage installation, you can collect logs to send to NetApp Support to help with diagnosis. You can either use NetApp Hybrid Cloud Control or the REST API to collect logs on an Element system.

What you'll need

- Ensure that your storage cluster version is running NetApp Element software 11.3 or later.
- Ensure that you have deployed a management node running version 11.3 or later.

Log collection options

Choose one of the following options:

- Use NetApp Hybrid Cloud Control to collect logs
- Use the REST API to collect logs

Use NetApp Hybrid Cloud Control to collect logs

You can access the log collection area from the NetApp Hybrid Cloud Control Dashboard.

Steps

1. Open the IP address of the management node in a web browser. For example:

   ```
   https://[management node IP address]
   ```

2. Log in to NetApp Hybrid Cloud Control by providing the Element storage cluster administrator credentials.
3. From the Dashboard, select the menu on the upper right.
4. Select **Collect Logs**.

   If you have collected logs before, you can download the existing log package, or begin a new log collection.

5. Select a date range in the **Date Range** drop-down menu to specify what dates the logs should include.

   If you specify a custom start date, you can select the date to begin the date range. Logs will be collected from that date up to the present time.

6. In the **Log Collection** section, select the types of log files the log package should include.

   For storage logs, you can expand the list of storage nodes and select individual nodes to collect logs from (or all nodes in the list).

7. Select **Collect Logs** to start log collection.

   Log collection runs in the background, and the page shows the progress.

   > Depending on the logs you collect, the progress bar might remain at a certain percentage for several minutes, or progress very slowly at some points.

8. Select **Download Logs** to download the log package.

   The log package is in a compressed UNIX .tgz file format.

Use the REST API to collect logs

You can use REST API to collect Element logs.

Steps

1. Locate the storage cluster ID:
   a. Open the management node REST API UI on the management node:

```
https://[management node IP]/logs/1/
```

b. Select **Authorize** and complete the following:

i. Enter the cluster user name and password.

ii. Enter the client ID as `mnode-client` if the value is not already populated.

iii. Select **Authorize** to begin a session.

2. Collect logs from Element:

a. Select **POST /bundle**.

b. Select **Try it out**.

c. Change the values of the following parameters in the **Request body** field depending on which type of logs you need to collect and for what time range:

Parameter	Type	Description
`modifiedSince`	Date string	Only include logs modified after this date and time. For example, the value "2020-07-14T20:19:00.000Z" defines a start date of July 14, 2020 at 20:19 UTC.
`mnodeLogs`	Boolean	Set this parameter to `true` to include management node logs.
`storageCrashDumps`	Boolean	Set this parameter to `true` to include storage node crash debug logs.
`storageLogs`	Boolean	Set this parameter to `true` to include storage node logs.
`storageNodeIds`	UUID array	If `storageLogs` is set to `true`, populate this parameter with the storage cluster node IDs to limit log collection to those specific storage nodes. Use the `GET https://[management node IP]/logs/1/bundle/options` endpoint to see all possible node IDs you can use.

d. Select **Execute** to begin log collection.
The response should return a response similar to the following:

```
{
  "_links": {
    "self": "https://10.1.1.5/logs/1/bundle"
  },
  "taskId": "4157881b-z889-45ce-adb4-92b1843c53ee",
  "taskLink": "https://10.1.1.5/logs/1/bundle"
}
```

3. Check on the status of the log collection task:
   a. Select **GET /bundle**.
   b. Select **Try it out**.
   c. Select **Execute** to return a status of the collection task.
   d. Scroll to the bottom of the response body.

      You should see a `percentComplete` attribute detailing the progress of the collection. If the collection is complete, the `downloadLink` attribute contains the full download link including the file name of the log package.

   e. Copy the file name at the end of the `downloadLink` attribute.
4. Download the collected log package:
   a. Select **GET /bundle/{filename}**.
   b. Select **Try it out**.
   c. Paste the file name you copied earlier into the `filename` parameter text field.
   d. Select **Execute**.

      After execution, a download link appears in the response body area.

   e. Select **Download file** and save the resulting file to your computer.

      The log package is in a compressed UNIX .tgz file format.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Manage storage with Element API

You can manage Element storage clusters using the Element software API.

The Element API is based on the JSON-RPC protocol over HTTPS. JSON-RPC is a simple text- based RPC protocol based on the lightweight JSON data-interchange format. Client libraries are available for all major programming languages.

- About the Element software API
- Common objects
- Common methods
- Account API methods
- Administrator API methods
- Cluster API methods
- Cluster creation API Methods
- Drive API methods
- Fibre Channel API methods
- Initiator API methods
- LDAP API methods
- Multi-factor authentication API methods
- Session authentication API methods
- Node API methods
- Replication API methods
- Security API methods
- SnapMirror API methods
- System configuration API methods
- Multitenant networking API methods
- Volume API methods
- Volume access group API methods
- Volume snapshot API methods
- Virtual volume API methods
- Access control
- Response examples

Find more information

- SolidFire All-Flash Storage Resources page
- SolidFire and Element Software Documentation Center

About the Element software API

The Element API is based on the JSON-RPC protocol over HTTPS. JSON-RPC is a simple text-based RPC protocol based on the lightweight JSON data-interchange format. Client libraries are available for all major programming languages.

You can make API requests via HTTPS POST requests to the API endpoint. The body of the POST request is a JSON-RPC request object. The API does not currently support batch requests (multiple request objects in a single POST). When submitting API requests, you must use "application/json-rpc" as the content-type of the request, and ensure that the body is not form-encoded.

The Element web UI makes use of the API methods described in this document. You can monitor API operations in the UI by enabling the API Log; this enables you to see the methods that are being issued to the system. You can enable both requests and responses to see how the system replies to the methods that are issued.

Unless stated otherwise, all date strings in the API responses are in UTC+0 format.

When the storage cluster is heavily loaded or you submit many consecutive API requests with no intervening delay, a method might fail and return the error "xDBVersionMismatch". If this happens, retry the method call.

- Request object members
- Response object members
- Request endpoints
- API authentication
- Asynchronous methods
- Attributes

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Request object members

Each Element software API request has the following basic parts:

Name	Description	Type	Default value	Required
method	Name of the method to be invoked.	string	None	Yes

Name	Description	Type	Default value	Required
parameters	Object containing the parameters to the method being invoked. Named parameters are required. Positional parameters (passed as an array) are not allowed.	JSON object	{}	No
id	Identifier used to match the request to response, returned in the result.	string or integer	{}	No

Response object members

Each Element software API response body has the following basic parts:

Name	Description	Type
result	The object returned by the method. The system returns an object with named members corresponding to the documented return value for the method. This member is not present if an error has occurred.	JSON object
error	The object returned when an error occurs. This member is present only if an error has occurred.	Object
id	An identifier used to match the request to response, as provided in the request.	string or integer
unusedParameters	A warning message that at least one incorrect parameter has been passed to the API method and has not been used.	Object

Request endpoints

There are three types of request endpoints used in the API (storage cluster, storage cluster creation, and per-node). You should always use the latest endpoint supported by your version of Element software.

The three request endpoints in the API are designated in the following ways:

Cluster API methods

The HTTPS endpoint for storage-cluster-wide API requests is `https://<mvip>/json-rpc/<api-version>`, where:

- `<mvip>` is the management virtual IP address for the storage cluster.
- `<api-version>` is the version of the API you are using.

Cluster creation and bootstrap API methods

The HTTPS endpoint for creating a storage cluster and accessing bootstrap API requests is `https://<nodeIP>/json-rpc/<api-version>`, where:

- `<nodeIP>` is the IP address of the node you are adding to the cluster.
- `<api-version>` is the version of the API you are using.

Per-node API methods

The HTTPS endpoint for individual storage node API requests is `https://<nodeIP>:442/json-rpc/<api-version>`, where:

- `<nodeIP>` is the management IP address of the storage node; 442 is the port the HTTPS server is running on.
- `<api-version>` is the version of the API you are using.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

API authentication

You can authenticate with the system when using the API by including an HTTP Basic authentication header with all API requests. If you omit authentication information, the system rejects the unauthenticated request with an HTTP 401 response. The system supports HTTP Basic authentication over TLS.

Use the cluster admin account for API authentication.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Asynchronous methods

Some API methods are asynchronous, which means that the operation they perform might not be complete when the method returns. Asynchronous methods return a handle that you can query to see the status of the operation; status information for some

operations might include a percentage of completion.

When you query an asynchronous operation, its result can be one of the following types:

- `DriveAdd`: The system is adding a drive to the cluster.
- `BulkVolume`: The system is performing a copy operation between volumes, such as a backup or restore.
- `Clone`: The system is cloning a volume.
- `DriveRemoval`: The system is copying data from a drive in preparation to remove it from the cluster.
- `RtfiPendingNode`: The system is installing compatible software on a node before adding it to the cluster.

Note the following points when using asynchronous methods or obtaining the status of a running asynchronous operation:

- Asynchronous methods are indicated in the individual method documentation.
- Asynchronous methods return an “asyncHandle”, which is a handle that is known by the issuing API method. You can use the handle to poll for the status or result of the asynchronous operation.
- You can obtain the result of individual asynchronous methods with the GetAsyncResult method. When you use GetAsyncResult to query a completed operation, the system returns the result and automatically purges the result from the system. When you use GetAsyncResult to query an incomplete operation, the system returns the result but does not purge it.
- You can obtain the status and results of all running or completed asynchronous methods using the ListAsyncResults method. In this case, the system does not purge results for completed operations.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Attributes

Many of the API requests and responses use objects as well as simple types. Objects are a collection of key-value pairs, where the value is a simple type or possibly another object. Attributes are custom name-value pairs that can be set by the user in JSON objects. Some methods enable you to add attributes when creating or modifying objects.

There is a 1000-byte limit on encoded attribute objects.

Object member

This object contains the following member:

Name	Description	Type
attributes	List of name-value pairs in JSON object format.	JSON object

Request example

The following request example uses the AddClusterAdmin method:

```
{
   "method": "AddClusterAdmin",
   "params": {
      "username": "joeadmin",
      "password": "68!5Aru268)$",
      "access": [
         "volume",
         "reporting"
      ],
      "attributes": {
         "name1": "value1",
         "name2": "value2",
         "name3": "value3"
      }
   }
}
```

Common objects

The Element software API uses JSON objects to represent organized data concepts. Many of these API methods make use of these objects for data input and output. This section documents these commonly used objects; objects that are only used within a single method are documented with that method instead of in this section.

- account
- authSessionInfo
- bulkVolumeJob
- binding (virtual volumes)
- certificateDetails
- cluster
- clusterAdmin
- clusterCapacity
- clusterConfig
- clusterInfo
- clusterPair
- clusterStats
- clusterStructure
- drive

- driveStats
- error
- event
- fault
- fibreChannelPort
- fipsErrorNodeReport
- fipsNodeReport
- fipsReport
- groupSnapshot
- hardwareInfo
- host (virtual volumes)
- idpConfigInfo
- initiator
- ISCSIAuthentication
- keyProviderKmip
- keyServerKmip
- ldapConfiguration
- loggingServer
- network (bonded interfaces)
- network (all interfaces)
- network (Ethernet interfaces)
- network (local interfaces)
- network (SNMP)
- networkInterface
- node
- nodeProtectionDomains
- nodeStats
- ontapVersionInfo
- pendingActiveNode
- pendingNode
- protectionDomain
- protectionDomainLevel
- protectionDomainResiliency
- protectionDomainTolerance
- protectionSchemeResiliency
- protectionSchemeTolerance
- protocolEndpoint

- QoS
- QoSPolicy
- remoteClusterSnapshotStatus
- schedule
- session (Fibre Channel)
- session (iSCSI)
- snapMirrorAggregate
- snapMirrorClusterIdentity
- snapMirrorEndpoint
- snapMirrorJobScheduleCronInfo
- snapMirrorLunInfo
- snapMirrorNetworkInterface
- snapMirrorNode
- snapMirrorPolicy
- snapMirrorPolicyRule
- snapMirrorRelationship
- snapMirrorVolume
- snapMirrorVolumeInfo
- snapMirrorVserver
- snapMirrorVserverAggregateInfo
- snapshot
- snmpTrapRecipient
- storageContainer
- syncJob
- task (virtual volumes)
- usmUser
- virtualNetwork
- virtualVolume
- volume
- volumeAccessGroup
- volumePair
- volumeStats

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

account

The `account` object contains information about an account. This object includes only "configured" information about the account, not any runtime or usage information.

Object members

This object contains the following members:

Name	Description	Type
`accountID`	The unique account ID for the account.	integer
`attributes`	List of name-value pairs in JSON object format.	JSON object
`enableChap`	Specifies whether CHAP account credentials can be used by an initiator to access volumes.	boolean
`initiatorSecret`	The initiator CHAP secret.	string
`status`	The current status of the account. Possible values: • active: An active account. • locked: A locked account. • removed: An account that has been deleted and purged.	string
`storageContainerID`	The unique ID of the virtual volume storage container associated with this account.	UUID
`targetSecret`	The target CHAP secret.	string
`username`	The username for the account.	string
`volumes`	A list of volume IDs for volumes owned by this account.	integer array

Find more information

- AddAccount
- GetAccountByID
- GetAccountByName

- ListAccounts

authSessionInfo

The `authSessionInfo` object contains information about an auth session.

Object members

This object contains the following members:

Name	Description	Type
`accessGroupList`	List of access groups for the user.	string array
`authMethod`	The type of authorization the cluster admin user has. Possible values: • LDAP - authenticated via LDAP. • Cluster - authenticated via a username and password stored in the cluster database. • IdP - authenticated via a third-party Identity Provider.	string
`clusterAdminIDs`	List of cluster AdminID(s) associated with this session. For sessions related to LDAP or a third-party Identity Provider (IdP), this will be an aggregate list of matching Cluster AdminIDs associated with this session.	integer array
`finalTimeout`	Time at which the session becomes invalid.This is set when the session is created and cannot be changed.	string
`idpConfigVersion`	IdP configuration version when the session was created.	integer
`lastAccessTimeout`	Time at which the session becomes invalid due to inactivity.It is set to a new value when the session is accessed for use, up to the time where the session becomes invalid due to finalTimeout being reached.	string

Name	Description	Type
sessionCreationTime	Time at which the session is created.	string
sessionID	UUID for this session.	UUID
username	Username associated with this session. For sessions related to LDAP, this will be the user's LDAP DN. For sessions related to a third-party IdP, this will be an arbitrary name-value pair that will be used for auditing operations within the session. It will not necessarily match a cluster admin name on the cluster. For example, a SAML Subject NameID,but this will be dictated by the configuration of the IdP and the resultant content of the SAML assertion.	string

bulkVolumeJob

The bulkVolumeJob object contains information about bulk volume read or write operations, such as cloning or snapshot creation.

Object members

This object contains the following members:

Name	Description	Type
attributes	JSON attribute of the bulk volume job.	JSON object
bulkVolumeID	The internal bulk volume job ID.	integer
createTime	Timestamp created for the bulk volume job in UTC+0 format.	ISO 8601 date string
elapsedTime	The number of seconds since the job began.	string
format	The format of the bulk volume operation. Possible values: • native • uncompressed	string

Name	Description	Type
`key`	The unique key created by the bulk volume session.	string
`percentComplete`	The completed percentage reported by the operation.	integer
`remainingTime`	The estimated time remaining in seconds.	integer
`srcVolumeID`	The source volume ID.	integer
`status`	The status of the operation. Possible values: • preparing • running • complete • failed	string
`script`	The name of the script if one is provided.	string
`snapshotID`	The ID of the snapshot if a snapshot is in the source of the bulk volume job.	integer
`type`	The type of bulk operation. Possible values: • read • write	string

binding (virtual volumes)

The binding object contains information about the binding for a virtual volume. You can retrieve a list of this information for all virtual volumes using the `ListVirtualVolumeBindings` API method.

Object members

This object contains the following members:

Name	Description	Type
protocolEndpointID	The unique ID of the protocol endpoint.	UUID
protocolEndpointInBandID	The scsiNAADeviceID of the protocol endpoint.	string
protocolEndpointType	The type of protocol endpoint. SCSI is the only value returned for the protocol endpoint type.	string
virtualVolumeBindingID	The unique ID of the virtual volume binding object.	integer
virtualVolumeHostID	The unique ID of the virtual volume host.	UUID
virtualVolumeID	The unique ID of the virtual volume.	UUID
virtualVolumeSecondaryID	The secondary ID of the virtual volume.	string

Find more information

- ListVirtualVolumeBindings
- protocolEndpoint

certificateDetails

The `certificateDetails` object contains the decoded information about a security certificate.

Object members

This object contains the following members:

Name	Description	Type
issuer	The name of the issuer.	string
modulus	The modulus of the public key.	string
notAfter	The expiry date of the certificate.	ISO 8601 string
notBefore	The start date of the certificate.	ISO 8601 string

Name	Description	Type
`serial`	The certificate serial number.	string
`sha1Fingerprint`	The digest of the DER-encoded version of the certificate.	string
`subject`	The subject name.	string

cluster

The cluster object contains information that the node uses to communicate with the cluster. You can retrieve this information with the GetClusterConfig API method.

Object members

This object contains the following members:

Name	Description	Type
cipi	Network interface used for cluster communication.	string
cluster	Unique cluster name.	string
encryptionCapable	Indicates whether the node supports drive encryption.	boolean
ensemble	The nodes that are participating in the cluster.	string array
fipsDriveConfiguration	Indicates whether the node supports FIPS 140-2 certified drives.	boolean
mipi	The network interface used for node management.	string
name	The cluster name.	string
nodeID	The node ID of the node in the cluster.	string
pendingNodeID	The ID of the pending node in the cluster.	integer
role	Identifies the role of the node.	integer

Name	Description	Type
sipi	The network interface used for storage traffic.	string
state	The current state of the node. Possible values: • Available: The node has not been configured with a cluster name. • Pending: The node is pending for a specific named cluster and can be added. • Active: The node is an active member of a cluster and cannot be added to another cluster. • PendingActive: The node is currently being returned to the factory software image, and is not yet an active member of a cluster. When complete, it will transition to the Active state.	string
version	The version of software running on the node.	string

Member modifiability and node states

This table indicates whether or not the object parameters can be modified at each possible node state.

Parameter name	Available state	Pending state	Active state
cipi	No	No	No
cluster	Yes	Yes	No
encryptionCapable	No	No	No
ensemble	No	No	No
mipi	Yes	Yes	No
name	Yes	Yes	Yes
nodeID	No	No	No
pendingNodeID	No	No	No

role	No	No	No
sipi	No	No	No
state	No	No	No
version	No	No	No

Find more information

GetClusterConfig

clusterAdmin

The clusterAdmin object contains information about the current cluster administrator user. You can retrieve admin user information with the GetCurrentClusterAdmin API method.

Object members

This object contains the following members:

Name	Description	Type
access	The methods this cluster admin can use.	string array
authMethod	The type of authorization the cluster admin user has. Possible values: • LDAP • Cluster • Local	string
attributes	List of name-value pairs in JSON object format.	JSON object
clusterAdminID	The cluster administrator ID for this cluster admin user.	integer
username	User name for this cluster admin.	string

Find more information

GetCurrentClusterAdmin

clusterCapacity

The clusterCapacity object contains high-level capacity measurements for the cluster. You can get cluster capacity information with the GetClusterCapacity API method. Space measurements in the object members are calculated in bytes.

Object members

This object contains the following members:

Name	Description	Type
activeBlockSpace	The amount of space on the block drives. This includes additional information such as metadata entries and space which can be cleaned up.	integer
activeSessions	The number of active iSCSI sessions communicating with the cluster.	integer
averageIOPS	The average IOPS for the cluster since midnight Coordinated Universal Time (UTC).	integer
clusterRecentIOSize	The average size of IOPS to all volumes in the cluster.	integer
currentIOPS	The average IOPS for all volumes in the cluster over the last 5 seconds.	integer
maxIOPS	The estimated maximum IOPS capability of the current cluster.	integer
maxOverProvisionableSpace	The maximum amount of provisionable space. This is a computed value. You cannot create new volumes if the current provisioned space plus the new volume size would exceed this number. The value is calculated as follows: `maxOverProvisionableSpace = maxProvisionedSpace * maxMetadataOverProvisionFactor`	integer

Name	Description	Type
maxProvisionedSpace	The total amount of provisionable space if all volumes are 100% filled (no thin provisioned metadata).	integer
maxUsedMetadataSpace	The number of bytes on volume drives used to store metadata.	integer
maxUsedSpace	The total amount of space on all active block drives.	integer
nonZeroBlock	The total number of 4KiB blocks that contain data after the last garbage collection operation has completed.	integer
peakActiveSessions	The peak number of iSCSI connections since midnight UTC.	integer
peakIOPS	The highest value for currentIOPS since midnight UTC.	integer
provisionedSpace	The total amount of space provisioned in all volumes on the cluster.	integer
timestamp	The date and time, in UTC+0 format, that this cluster capacity sample was taken.	ISO 8601 string
totalOps	The total number of I/O operations performed throughout the lifetime of the cluster.	integer
uniqueBlocks	The total number of blocks stored on the block drives. The value includes replicated blocks.	integer
uniqueBlocksUsedSpace	The total amount of data the uniqueBlocks take up on the block drives. See the GetclusterCapacity method for information about how this number relates to the uniqueBlocks value.	integer
usedMetadataSpace	The total number of bytes on volume drives used to store metadata.	integer

Name	Description	Type
usedMetadataSpaceInSnapshots	The number of bytes on volume drives used for storing unique data in snapshots. This number provides an estimate of how much metadata space would be regained by deleting all snapshots on the system.	integer
usedSpace	The total amount of space used by all block drives in the system.	integer
zeroBlocks	The total number of empty 4KiB blocks without data after the last round of garbage collection operation has completed.	integer

Find more information

GetClusterCapacity

clusterConfig

The `clusterConfig` object returns information the node uses to communicate with the cluster.

Object members

This object contains the following members:

Name	Description	Type
`cipi`	Network interface used for cluster communication.	string
`cluster`	Unique name of the cluster.	string
`encryptionCapable`	Specifies whether the node supports encryption.	boolean
`ensemble`	Nodes that are participating in the cluster.	string array
`fipsDriveConfiguration`	Specifies whether the node supports FIPS 140-2 certified drives.	boolean

Name	Description	Type
hasLocalAdmin	Specifies whether the cluster has a local administrator.	boolean
mipi	Network interface used for node management.	string
name	Unique identifier for the cluster.	string
nodeID	Unique identifier for the node.	integer
pendingNodeID	Unique identifier for the pending node.	integer
role	Identifies the role of the node.	string
sipi	Network interface used for storage.	string
state	Indicates the state of the node.	string
version	Indicates the version of the node.	string

clusterInfo

The clusterInfo object contains information that the node uses to communicate with the cluster. You can get this information with the GetClusterInfo API method.

Object members

This object contains the following members:

Name	Description	Type
attributes	List of name-value pairs in JSON object format.	JSON object
defaultProtectionScheme	The protection scheme used by default for new volumes, unless a protection scheme is provided with the CreateVolume method call. This protection scheme must always be in the set of enabled protection schemes.	string

Name	Description	Type
enabledProtectionSchemes	A list of all protection schemes that have been enabled on this storage cluster.	string array
encryptionAtRestState	The state of the Encryption at Rest feature. Possible values: • Enabling: Encryption at rest is being enabled. • Enabled: Encryption at rest is enabled. • Disabling: Encryption at rest is being disabled. • Disabled: Encryption at rest is disabled.	string
ensemble	The nodes that are participating in the cluster.	string array
mvip	The floating (virtual) IP address for the cluster on the management network.	string
mvipInterface	The physical interface associated with the MVIP address.	string
mvipNodeID	The node that holds the master MVIP address.	integer
mvipVlanTag	The VLAN identifier for the MVIP address.	string
name	The unique cluster name.	string
repCount	The number of replicas of each piece of data to store in the cluster. The valid value is "2".	integer
softwareEncryptionAtRestState	Software-based encryption-at-rest state.	string
supportedProtectionSchemes	A list of all protection schemes that are supported on this storage cluster.	string array

Name	Description	Type
svip	The floating (virtual) IP address for the cluster on the storage (iSCSI) network.	string
svipInterface	The physical interface associated with the master SVIP address.	string
svipNodeID	The node holding the master SVIP address.	integer
svipVlanTag	The VLAN identifier for the master SVIP address.	string
uniqueID	The unique ID for the cluster.	string
uuid	The unique identifier for the cluster.	UUID

Find more information

- GetClusterInfo
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

clusterPair

The clusterPair object contains information about clusters paired with the local cluster. You can retrieve a list of clusterPair objects for the local cluster with the ListClusterPairs method.

Object members

This object contains the following members:

Name	Description	Type
clusterName	The name of the other cluster in the pair.	string
clusterPairID	A unique ID given to each cluster in the pair.	integer
clusterPairUUID	The universally unique identifier for the cluster pair.	string

Name	Description	Type
UUID	Unique identifier for the remote cluster in the cluster pair.	integer
latency	The latency, in milliseconds, between clusters.	integer
mvip	The IP address of the management connection for paired clusters.	string
status	The status of the connection between the paired clusters. Possible values: • Unconfigured • Connected • Misconfigured • Disconnected	string
version	The Element version of the other cluster in the pair.	string

Find more information

ListClusterPairs

clusterStats

The clusterStats object contains statistical data for a cluster. Many of the volume-related statistics contained in the object are averaged for all volumes in the cluster. You can use the GetClusterStats method to retrieve this information for a cluster.

Object members

This object contains the following members:

Name	Description	Calculation	Type
actualIOPS	Current actual IOPS for the entire cluster in the last 500 milliseconds.	Point in time	integer
averageIOPSize	Average size in bytes of recent I/O to the cluster in the last 500 milliseconds.	Point in time	integer

Name	Description	Calculation	Type
clientQueueDepth	The number of outstanding read and write operations to the cluster.	N/A	integer
clusterUtilization	The percentage of the cluster's max IOPS currently being utilized. This is computed as clusterUtilization = normalizedIOPS / maxIOPS (from GetClusterCapacity).	N/A	float
latencyUSec	The average time, in microseconds, to complete operations to a cluster in the last 500 milliseconds.	Point in time	integer
normalizedIOPS	Average number of IOPS for the entire cluster in the last 500 milliseconds.	Point in time	integer
readBytes	The total cumulative bytes read from the cluster since the creation of the cluster.	Monotonically increasing	integer
readBytesLastSample	The total number of bytes read from the cluster during the last sample period.	Point in time	integer
readLatencyUSec	The average time, in microseconds, to complete read operations to the cluster in the last 500 milliseconds.	Point in time	integer
readLatencyUSecTotal	The total time spent performing read operations since the creation of the cluster.	Monotonically increasing	integer

Name	Description	Calculation	Type
readOps	The total cumulative read operations to the cluster since the creation of the cluster.	Monotonically increasing	integer
readOpsLastSample	The total number of read operations during the last sample period.	Point in time	integer
samplePeriodMSec	The length of the sample period, in milliseconds.	N/A	integer
servicesCount	The number of services running on the cluster. If equal to the servicesTotal, this indicates that valid statistics were collected from all nodes.	Point in time	integer
servicesTotal	The total number of expected services running on the cluster.	N/A	integer
timestamp	The current time in UTC+0 format.	N/A	ISO 8601 date string
unalignedReads	The total cumulative unaligned read operations to a cluster since the creation of the cluster.	Monotonically increasing	integer
unalignedWrites	The total cumulative unaligned write operations to a cluster since the creation of the cluster.	Monotonically increasing	integer
writeBytes	The total cumulative bytes written to the cluster since the creation of the cluster.	Monotonically increasing	integer
writeBytesLastSample	The total number of bytes written to the cluster during the last sample period.	Monotonically increasing	integer

Name	Description	Calculation	Type
writeLatencyUSec	The average time, in microseconds, to complete write operations to a cluster in the last 500 milliseconds.	Point in time	integer
writeLatencyUSecTotal	The total time spent performing write operations since the creation of the cluster.	Monotonically increasing	integer
writeOps	The total cumulative write operations to the cluster since the creation of the cluster.	Monotonically increasing	integer
writeOpsLastSample	The total number of write operations during the last sample period.	Point in time	integer

Find more information

GetClusterStats

clusterStructure

The clusterStructure object holds cluster configuration backup information created by the GetClusterStructure method. You can use the SetClusterStructure method to restore this information to a storage cluster you are rebuilding.

Object members

This object contains the combined return information from the following methods:

- GetClusterInfo
- ListAccounts
- ListInitiators
- ListVolumes (with includeVirtualVolumes=false)
- ListVolumeAccessGroups
- ListStorageContainers
- ListQoSPolicies
- GetSnmpInfo
- GetNtpInfo
- ListVirtualNetworks

- ListClusterAdmins
- ListSchedules
- ListSnapMirrorEndpoints
- GetFeatureStatus
- GetLdapConfiguration
- GetRemoteLoggingHosts
- GetDefaultQoS
- GetVolumeAccessGroupLunAssignments

Find more information

- GetClusterStructure
- SetClusterStructure

drive

The drive object contains information about individual drives in the cluster's active nodes. This object contains details on drives that have been added as volume metadata or block drives, as well as drives that have not yet been added and are available. You can retrieve this information with the `ListDrives` API method.

Object members

This object contains the following members:

Name	Description	Type
attributes	List of name-value pairs in JSON object format. This object is always null and is not modifiable.	JSON object
capacity	The total capacity of the drive, in bytes.	integer
chassisSlot	For HCI platforms, this value is the node letter and slot number in the server chassis where this drive is located. For storage platforms, the slot number is a string representation of the "slot" integer.	string
driveFailureDetail	If a drive's status is "Failed", this field provides more detail on why the drive was marked failed.	string
driveID	The ID of this drive.	integer

Name	Description	Type
driveSecurityFaultReason	If enabling or disabling drive security failed, the reason why it failed. If the value is "none", there was no failure.	string
keyID	The keyID used by the key provider to acquire the authentication key for unlocking this drive.	UUID
keyProviderID	Identifies the provider of the authentication key for unlocking this drive.	integer
nodeID	The ID of the node containing this drive.	integer
segmentFileSize	The segment file size of the drive, in bytes.	integer
serial	The drive serial number.	string
slot	The slot number in the server chassis where this drive is located, or -1 if a SATADimm device is used for the internal metadata drive.	integer
status	The status of the drive. Possible values: • available: An available drive. • active: An active drive. • erasing: A drive is in the process of being secure erased. Any data on that drive is permanently removed. • failed: A drive that has failed. Any data that was previously on the drive has been migrated to other drives in the cluster. • removing: A drive is in the process of being removed. Any data previously on the drive is being migrated to other drives in the cluster.	string

Name	Description	Type
type	The type of drive. Possible values: • volume: Stores volume metadata. • block: Stores block data. • unknown: Drive type not yet active and is yet to be determined.	string
usableCapacity	The usable capacity of the drive, in bytes.	integer

Find more information

ListDrives

driveStats

The driveStats object contains high-level activity measurements for a single drive. You can retrieve measurement information with the API method `GetDriveStats`.

Object members

This object contains the following members:

Name	Description	Type
activeSessions	Number of iSCSI sessions currently using this drive (only present for metadata drives).	integer
driveID	Unique ID of the drive in the cluster.	integer
failedDieCount	Number of failed drive hardware elements.	integer
iosInProgress	The number of I/Os to this drive that are in progress.	integer
lifeRemainingPercent	Drive media wear out indicator.	integer
lifetimeReadBytes	Total bytes read from this drive for the lifetime of the drive.	integer

Name	Description	Type
lifetimeWriteBytes	Total bytes written to this drive for the lifetime of the drive.	integer
powerOnHours	Number of hours this drive has been powered on.	integer
reads	The number of read() calls per second to this drive.	integer
readBytes	Total bytes read from the drive due to client operations.	integer
readsCombined	The number of read() calls to adjacent sectors that could be combined into a larger read.	integer
readMsec	The number of milliseconds spent reading.	integer
readOps	Total read operations on the drive due to client operations.	integer
reallocatedSectors	Number of bad sectors replaced in this drive.	integer
reserveCapacityPercent	The available reserve capacity of the drive.	integer
timestamp	The current time in UTC+0 format.	ISO 8601 date string
totalCapacity	Total capacity of the drive, in bytes.	integer
uncorrectableErrors	The Reported Uncorrectable Errors value from the Self-Monitoring, Analysis and Reporting Technology (SMART) monitoring system in the drive.	integer
usedCapacity	Used capacity of the drive, in bytes.	integer
usedMemory	Amount of memory currently used by the node hosting this drive.	integer
writes	The number of write() calls per second to this drive.	integer

Name	Description	Type
writeBytes	Total bytes written to the drive due to client activity.	integer
writesCombined	The number of write() calls to adjacent sectors that could be combined into a larger write.	integer
writeMsec	The number of milliseconds spent writing.	integer
writeOps	Total write operations to the drive due to client activity.	integer

Find more information

GetDriveStats

error

The error object contains an error code and message if an error occurs during a method call. All system-generated errors have an error code of 500.

Object members

This object contains the following members:

Name	Description	Type
code	The numeric code used to identify the error. All system-generated errors return a code of 500.	integer
name	The unique identifier for the specific error that occurred. Each method returns a documented set of errors, although you should be prepared to handle unrecognized errors as well.	string
message	A description of the error, possibly with additional details.	string

event

The event object contains details of events that occur during an API method call or while the system is performing an operation.

Object members

This object contains the following members:

Name	Description	Type
details	Extra information about the event.	JSON object
driveID	The driveID of the drive reporting the failure. 0 if not applicable.	integer
driveIDs	A list of the driveIDs of the drives reporting the failure. An empty list if not applicable.	integer array
eventID	Unique ID associated with each event.	integer
eventInfoType	The type of fault.	string
message	A string description of the event that occurred.	string
nodeID	The nodeID of the node reporting the failure. 0 if not applicable.	integer
serviceID	The serviceID of the service reporting the failure. 0 if not applicable.	integer
severity	Severity the event is reporting.	integer
timeOfPublish	The time at which the cluster's event log received the event, in UTC+0 format.	ISO 8601 date string
timeOfReport	The time at which the event occurred on the cluster, in UTC+0 format.	ISO 8601 date string

Note: There might be a slight difference between timeOfReport and timeOfPublish if the event occurred and was not able to be immediately published.

Event types

The following list describes the possible event types that the eventInfoType member can contain:

- apiEvent: Events initiated through the API or web UI that modify settings.
- binAssignmentsEvent: Events related to the assignment of data to internal containers.

- binSyncEvent: Events related to a reassignment of data among block services.
- bsCheckEvent: Events related to block service checks.
- bsKillEvent: Events related to block service terminations.
- bulkOpEvent: Events that operate on an entire volume, such as a volume backup, restore, snapshot, or clone.
- cloneEvent: Events related to volume cloning.
- clusterMasterEvent: Cluster configuration change events such as adding or removing nodes.
- dataEvent: Events related to reading and writing data.
- dbEvent: Events related to the ensemble node database.
- driveEvent: Events related to drive operations.
- encryptionAtRestEvent: Events related to stored data encryption.
- ensembleEvent: Events related to ensemble size increase or decrease.
- fibreChannelEvent: Events related to Fibre Channel node configuration or connections.
- gcEvent: Events related to garbage collection. These processes run every 60 minutes to reclaim storage on block drives.
- ieEvent: Events related to internal system errors.
- installEvent: Eevnts related to automatic software installation on pending storage nodes.
- iSCSIEvent: Events related to iSCSI connection or configuration issues.
- limitEvent: Events related to the number of volumes or virtual volumes in an account or in the cluster nearing the maximum allowed.
- networkEvent: Events related to virtual networking.
- platformHardwareEvent: Events related to issues detected on hardware devices.
- remoteClusterEvent: Events related to remote cluster pairing.
- schedulerEvent: Events related to scheduled snapshots.
- serviceEvent: Events related to system service status.
- statEvent: Events related to system statistics.
- sliceEvent: Events related to metadata storage.
- snmpTrapEvent: Events related to SNMP traps.
- tsEvent: System transport service events.
- unexpectedException: Events related to unexpected errors.
- vasaProviderEvent: Events related to a VMware VASA provider.

Find more information

ListEvents

fault

The fault object contains information about faults that are detected in the cluster. The `ListClusterFaults` method returns cluster fault information.

Object members

This object contains the following members:

Name	Description	Type
blocksUpgrade	The fault blocks an upgrade. Possible values: • true: The fault blocks an upgrade. • false: The fault does not block an upgrade.	boolean
clusterFaultID	The unique ID associated with each cluster fault.	integer
code	The fault code for the specific fault that was detected. For further details, see Cluster Fault Codes.	string
data	Additional fault-specific information.	JSON object
date	The current time in UTC+0 format.	ISO 8601 string
details	The description of the fault with additional details.	string
driveID	The first drive ID in the driveIDs list. If the driveIDs list is empty (which means that no faults were returned that deal with drives), this value is 0.	integer
driveIDs	A list of driveID values for the drives that this fault refers to. Included for faults dealing with drives. If none, this is an empty array.	integer array
nodeHardwareFaultID	The identifier assigned to a hardware fault on the cluster.	integer
nodeID	The node ID for the node that this fault refers to. Included for node and drive faults, otherwise set to 0.	integer

Name	Description	Type
resolved	The resolved status of the fault. Possible values: • true: The fault is no longer detected. • false: The fault is still present.	boolean
resolvedDate	The date and time the fault was resolved.	ISO 8601 string
serviceID	The service associated with the fault. This value is "0" (zero) if the fault is not associated with a service.	integer
severity	The severity of the fault. Possible values: • warning: A minor issue. The cluster is functioning and upgrades are allowed at this severity level. • error: A failure that generally should not affect service (except possible performance degradation or loss of HA). Some features might be disabled. • critical: A serious failure that is affecting service. The system is unable to serve API requests or client I/O and is at risk of data loss. • bestPractice: Faults triggered by sub-optimal system configuration.	string

Name	Description	Type
type	The type of fault. Possible values: • node: A fault affecting an entire node. • drive: A fault affecting an individual drive. • cluster: A fault affecting the entire cluster. • service: A fault affecting a service on the cluster. • volume: A fault affecting an individual volume.	string

Find more information

- ListClusterFaults
- Cluster fault codes

fibreChannelPort

The fibreChannelPort object contains information about individual ports on a node, or for an entire node in the cluster. You can retrieve this information using the `ListNodeFibreChannelPortInfo` method.

Object members

This object contains the following members:

Name	Description	Type
firmware	The version of the firmware installed on the Fibre Channel port.	integer
hbaPort	The ID of the individual host bus adapter (HBA) port.	integer
model	Model of the HBA on the port.	string
nPortID	The unique port node ID.	string
pciSlot	The slot containing the PCI card in the Fibre Channel node chassis.	integer
serial	The serial number on the Fibre Channel port.	string

Name	Description	Type
speed	The speed of the HBA on the port.	string
state	Possible values: • Unknown • NotPresent • Online • Offline • Blocked • Bypassed • Diagnostics • Linkdown • Error • Loopback • Deleted	string
switchWwn	The World Wide Name of the Fibre Channel switch port.	string
wwnn	World Wide Node Name of the HBA node.	string
wwpn	World Wide Port Name assigned to the physical port of the HBA.	string

Find more information

ListNodeFibreChannelPortInfo

fipsErrorNodeReport

The fipsErrorNodeReport object contains error information for each node that does not respond with information about FIPS 140-2 support when you query it with the `GetFipsReport` method.

Object members

This object contains the following members:

Name	Description	Type
nodeID	The ID of the node that did not respond.	integer

Name	Description	Type
error	A JSON object containing error information.	JSON object

fipsNodeReport

The fipsNodeReport object contains information about FIPS 140-2 support for a single node in the storage cluster. You can retrieve this information using the `GetFipsReport` method.

Object members

This object contains the following members:

Name	Description	Type
nodeID	The ID of the node reporting the information.	integer
fipsDrives	Whether or not FIPS 140-2 drive encryption is enabled for this node. Possible values: • None: This node is not capable of FIPS drive encryption. • Partial: Node is capable of FIPS drive encryption but not all drives present are FIPS-capable drives. • Ready: Node is capable of FIPS drive encryption and either all drives present are FIPS-capable drives, or there are no drives present.	FipsDrivesStatusType
httpsEnabled	Whether or not FIPS 140-2 HTTPS encryption is enabled for this node. Possible values: • true: enabled • false: disabled	boolean

fipsReport

The fipsReport object contains information about FIPS 140-2 support for all nodes in the storage cluster. You can retrieve this information using the `GetFipsReport` method.

Object members

This object contains the following members:

Name	Description	Type
nodes	A report on FIPS 140-2 support status for each node in the storage cluster.	fipsNodeReport
errorNodes	Error information for each node that did not respond with FIPS 140-2 support status.	fipsErrorNodeReport

groupSnapshot

The groupSnapshot object contains information about a snapshot for a group of volumes. You can use the `ListGroupSnapshots` API method to retrieve group snapshot information.

Object members

This object contains the following members:

Name	Description	Type
attributes	List of name-value pairs in JSON object format.	JSON object
createTime	The UTC+0 formatted day and time on which the group snapshot was created.	ISO 8601 date string
enableRemoteReplication	Identifies if the snapshot is enabled for remote replication.	boolean
groupSnapshotID	The unique ID of the group snapshot.	integer
groupSnapshotUUID	The UUID of the group snapshot.	string
members	An array of objects containing information about each member of the group snapshot.	snapshot array
name	The name of the group snapshot, or, if none was given, the UTC formatted day and time on which the snapshot was created.	string or ISO 8601 date string

Name	Description	Type
remoteStatuses	An array containing the universal identifier and replication status of each remote snapshot on the target cluster as seen from the source cluster.	remoteClusterSnapshotStatus array
status	Current status of the snapshot. Possible values: • Unknown: There was an error obtaining the status of the snapshot. • Preparing: This snapshot is being prepared for use and is not yet writable. • RemoteSyncing: This snapshot is being replicated from a remote cluster. • Done: This snapshot has finished preparation or replication and is now usable. • Active: This snapshot is the active branch. • Cloning: This snapshot is involved in a CopyVolume operation.	string

Find more information

ListGroupSnapshots

hardwareInfo

The hardwareInfo object contains detailed information about the hardware and status of each node in the cluster. You can retrieve this information with the `GetHardwareInfo` API method.

Object members

This object contains the following members:

Name	Description	Type
boardSerial	The DMI board serial number.	string

Name	Description	Type
bus	Motherboard media bus information.	JSON object
chassisSerial	The serial number of the chassis.	string
driveHardware	A list of information for each drive in the node.	JSON object array
fibreChannelPorts	A list of Fibre Channel ports on the node.	integer array
hardwareConfig	Motherboard peripheral configuration information.	JSON object
kernelCrashDumpState	The crash dump configuration of the operating system kernel.	string
memory	Firmware and system memory hardware information.	JSON object
network	Descriptions of the hardware of each of the node's network interfaces.	JSON object
networkInterfaces	The status of the node's network interfaces.	JSON object
nodeSlot	For HCI platforms, the letter corresponding to the chassis slot this node is in ("A", "B", "C", or "D"). For storage platforms, this value is null.	string
nvram	NVRAM statistics for the node.	JSON object
origin	The vendor of the motherboard.	string
platform	A description of the chassis platform.	JSON object
serial	The serial number of the product.	string
storage	Storage controller information.	JSON object

Name	Description	Type
systemMemory	Operating system memory usage and performance information.	JSON object
system	The type of node chassis.	JSON object
uuid	The unique ID of the node.	UUID

Find more information

GetHardwareInfo

host (virtual volumes)

The host object contains information about a virtual volume host. You can use the `ListVirtualVolumeHosts` method to get this information for all virtual volume hosts.

Object members

This object contains the following members:

Name	Description	Type
bindings	A list of objects describing the bindings for the virtual volume host.	integer array
clusterID	The unique ID of the cluster this host is associated with.	UUID
hostAddress	The IP address or DNS name of the virtual volume host.	string
initiatorNames	A list of initiator IQNs for the virtual volume host.	string array
virtualVolumeHostID	The unique ID of this virtual volume host.	UUID
visibleProtocolEndpointIDs	A list of IDs of protocol endpoints visible on this host.	UUID array

Find more information

ListVirtualVolumeHosts

idpConfigInfo

The idpConfigInfo object contains configuration and integration details regarding a third-party Identity Provider (IdP).

Object members

This object contains the following members:

Name	Description	Type
enabled	Specifies whether this third party IdPconfiguration is enabled.	boolean
idpConfigurationID	UUID for the third-party IdP configuration.	UUID
idpMetadata	Metadata for configuration and integration details for SAML 2.0 single sign-on.	string
idpName	Name for retrieving IdP provider for SAML 2.0 single sign-on.	string
serviceProviderCertificate	A PEM format Base64 encoded PKCS#10 X.509 certificate to be used for communication with this IdP.	string
spMetadataUrl	URL for retrieving Service Provider (SP) Metadata from the Cluster to provide to the IdP for establish a trust relationship.	string

initiator

The initiator object contains information about an iSCSI or Fibre Channel initiator. An initiator object can contain IQN or WWPN identifiers. You can use the `ListInitiators` method to get a list of all initiators known on the system. You use initiator objects to configure SCSI initiator access to a set of volumes through volume access groups. An initiator can only be a member of one volume access group at a time. You can restrict initiator access to one or more VLANs by specifying one or more virtualNetworkIDs using the `CreateInitiators` and `ModifyInitiators` methods. If you don’t specify any virtual networks, the initiator can access all networks.

Object members

This object contains the following members:

Name	Description	Type
alias	The friendly name assigned to the initiator, if any.	string
attributes	A set of JSON attributes assigned to this initiator. Empty if no attributes are assigned.	JSON object
chapUsername	The unique CHAP username for this initiator.	string
initiatorID	The numeric identifier for the initiator.	integer
initiatorName	The initiator name, in IQN or WWPN format.	string
initiatorSecret	The CHAP secret used to authenticate the initiator.	string
requireChap	True if CHAP is required for this initiator.	boolean
targetSecret	The CHAP secret used to authenticate the target (when using mutual CHAP authentication).	string
virtualNetworkIDs	The list of virtual network identifiers associated with this initiator. If one or more are defined, this initiator will only be able to login to the specified virtual networks. If no virtual networks are defined this initiator can login to all networks.	integer
volumeAccessGroups	A list of volume access group IDs that this initiator belongs to.	integer array

Find more information

ListInitiators

ISCSIAuthentication

The ISCSIAuthentication object contains authentication information about an ISCSI session.

Object members

This object contains the following members:

Name	Description	Type
authMethod	The authentication method used during iSCSI session login, for example, CHAP or None.	string
chapAlgorithm	The CHAP algorithm being used, for example, MD5, SHA1*, SHA-256*, or SHA3-256*	string
chapUsername	The CHAP username specified by the initiator during an iSCSI session login.	string
direction	The authentication direction, for example, one-way (initiator only) or two-way (both initiator and target).	string

- Available beginning with Element 12.7.

keyProviderKmip

The keyProviderKmip object describes a Key Management Interoperability Protocol (KMIP) key provider. A key provider is both a mechanism and a location for retrieving authentication keys for use with cluster features such as Encryption at Rest.

Object members

This object contains the following members:

Name	Description	Type
keyProviderID	The ID of the KMIP key provider. This is a unique value assigned by the cluster during key provider creation which cannot be changed.	integer
keyProviderIsActive	True if the KMIP key provider is active. A provider is considered active if there are outstanding keys which were created but not yet deleted and therefore assumed to still be in use.	boolean
keyProviderName	The name of the KMIP key provider.	string

Name	Description	Type
keyServerIDs	A key server ID that is associated with this provider. The server must be added before this provider can become active. The server cannot be removed while this provider is active. Only one server ID is supported for each provider.	integer array
kmipCapabilities	The capabilities of this KMIP key provider including details about the underlying library, FIPS compliance, SSL provider, etc.	string

keyServerKmip

The keyServerKmip object describes a Key Management Interoperability Protocol (KMIP) key server, which is a location for retrieving authentication keys for use with cluster features such as Encryption at Rest.

Object members

This object contains the following members:

Name	Description	Type
keyProviderID	If this KMIP key server is assigned to a provider, this member contains the ID of the KMIP key provider it is assigned to. Otherwise this member is null.	integer
keyServerID	The ID of the KMIP key server. This is a unique value assigned by the cluster during key server creation. This value cannot be changed.	integer
kmipAssignedProviderIsActive	If this KMIP key server is assigned to a provider (keyProviderID is not null), this member indicates whether that provider is active (providing keys which are currently in use). Otherwise, this member is null.	boolean

Name	Description	Type
kmipCaCertificate	The public key certificate of the external key server's root CA. This is used to verify the certificate presented by the external key server in the TLS communication. For key server clusters where individual servers use different CAs, this member contains a concatenated string of the root certificates of all the CAs.	string
kmipClientCertificate	A PEM format Base64 encoded PKCS#10 X.509 certificate used by the Element storage KMIP client.	string
kmipKeyServerHostnames	The hostnames or IP addresses associated with this KMIP key server.	string array
kmipKeyServerName	The name of the KMIP key server. This name is only used for display purposes and does not need to be unique.	string
kmipKeyServerPort	The port number associated with this KMIP key server (typically 5696).	integer

ldapConfiguration

The ldapConfiguration object contains information about the LDAP configuration on the storage system. You can retrieve LDAP information with the `GetLdapConfiguration` API method.

Object members

This object contains the following members:

Name	Description	Type
authType	Identifies which user authentication method to use. Possible values: • DirectBind • SearchAndBind	string

Name	Description	Type
enabled	Identifies whether or not the system is configured for LDAP. Possible values: • true • false	boolean
groupSearchBaseDN	The base DN of the tree to start the group search (the system will perform a subtree search from here).	string
groupSearchCustomFilter	The custom search filter used.	string
groupSearchType	Controls the default group search filter used. Possible values: • NoGroups: No group support. • ActiveDirectory: Nested membership of all of a user's AD groups. • MemberDN: MemberDN style groups (single-level).	string
searchBindDN	A fully qualified DN to log in with to perform an LDAP search for the user (needs read access to the LDAP directory).	string
serverURIs	A comma-separated list of LDAP server URIs (for example, `ldap://1.2.3.4` and `ldaps://1.2.3.4:123`.)	string
userDNTemplate	A string that is used to form a fully qualified user DN.	string
userSearchBaseDN	The base DN of the tree used to start the search (will do a subtree search from here).	string
userSearchFilter	The LDAP filter used.	string

Find more information

GetLdapConfiguration

loggingServer

The loggingServer object contains information about any logging hosts configured for the storage cluster. You can use `GetRemoteLoggingHosts` to determine what the current logging hosts are and then use `SetRemoteLoggingHosts` to set the desired list of current and new logging hosts.

Object members

This object contains the following members:

Name	Description	Type
host	IP address of the log server.	string
port	Port number used to communicate with the log server.	integer

network (bonded interfaces)

The network (bonded interfaces) object contains configuration information for bonded network interfaces on a storage node. You can use the `GetConfig` and `GetNetworkConfig` methods to obtain this information for a storage node.

Object members

This object contains the following members:

Name	Description	Type
address	The IPv4 address assigned to this interface on the node.	string
addressV6	The IPv6 management address assigned to the Bond1G interface on the node.	string
bond-downdelay	Time to wait, in milliseconds, before disabling a slave after a link failure has been detected.	string
bond-fail_over_mac	The configuration of the MAC address of the network interface.	string
bond-miimon	The frequency, in milliseconds, at which the MII link state is inspected for link failures.	string

bond-mode	The bonding mode. Possible values: • ActivePassive (Default) • ALB • LACP (Recommended)	string
bond-primary_reselect	Specifies when the primary bond slave is chosen as the active slave. Possible values: • Always • Better • Failure	string
bond-slaves	The list of slave interfaces for the bond.	string
bond-lacp_rate	When Bond Mode is LACP, the rate may change to one of the following: • LACP Fast (Default) • LACP Slow	string
bond-updelay	The time, in milliseconds, to wait before enabling a slave after a link is detected.	string
dns-nameservers	A list of addresses used for domain name services, separated by comma or space.	string
dns-search	A space or comma separated list of DNS search domains.	string
family	Address family that the interface is configured to use. Currently "inet" for IPv4 is supported.	string
gateway	The IPv4 router network address used to send traffic from the local network.	string
gatewayV6	The IPv6 router network address used to send traffic from the local Bond1G network.	string

ipV6PrefixLength	The subnet prefix length for static routes of type "net" for IPv6 traffic on the Bond1G network.	string
macAddress	The actual MAC address assigned to the interface and observed by the network.	string
macAddressPermanent	The immutable MAC address assigned by the manufacturer to the interface.	string
method	The method used to configure the interface. Possible values: • Loopback: Used to define the IPv4 loopback interface. • manual: Used to define interfaces that are not configured automatically. • dhcp: Can be used to obtain an IP address via DHCP. • static: Used to define Ethernet interfaces with statically allocated IPv4 addresses.	string
mtu	The largest packet size (in bytes) that the interface can transmit. Must be greater than or equal to 1500; up to 9000 is supported.	string
netmask	The bitmask that specifies the subnet for the interface.	string
network	Indicates where the IP address range begins based on the netmask.	string
routes	Comma separated array of route strings to apply to the routing table.	string array

status	The state of the interface. Possible values: • Down: The interface is inactive. • Up: The interface is ready, but has no link. • UpAndRunning: The interface is ready and a link is established.	string
symmetricRouteRules	The symmetric routing rules configured on the node.	string array
upAndRunning	Indicates if the interface is ready and has a link.	boolean
virtualNetworkTag	The virtual network identifier of the interface (VLAN tag).	string

Member modifiability and node states

This table indicates whether or not the object parameters can be modified at each possible node state.

Member name	Available state	Pending state	Active state
address	Yes	Yes	No
addressV6	Yes	Yes	No
bond-downdelay	Configured by the system	N/A	N/A
bond-fail_over_mac	Configured by the system	N/A	N/A
bond-miimon	Configured by the system	N/A	N/A
bond-mode	Yes	Yes	Yes
bond-primary_reselect	Configured by the system	N/A	N/A
bond-slaves	Configured by the system	N/A	N/A
bond-lacp_rate	Yes	Yes	Yes
bond-updelay	Configured by the system	N/A	N/A
dns-nameservers	Yes	Yes	Yes

dns-search	Yes	Yes	Yes
family	No	No	No
gateway	Yes	Yes	Yes
gatewayV6	Yes	Yes	Yes
ipV6PrefixLength	Yes	Yes	Yes
macAddress	Configured by the system	N/A	N/A
macAddressPermanent	Configured by the system	N/A	N/A
method	No	No	No
mtu	Yes	Yes	Yes
netmask	Yes	Yes	Yes
network	No	No	No
routes	Yes	Yes	Yes
status	Yes	Yes	Yes
symmetricRouteRules	Configured by the system	N/A	N/A
upAndRunning	Configured by the system	N/A	N/A
virtualNetworkTag	Yes	Yes	Yes

Find more information

- GetConfig
- GetNetworkConfig

network (all interfaces)

The network (all interfaces) object collects information about network interface configuration for a storage node. You can use the `GetConfig` and `GetNetworkConfig` methods to obtain this information for a storage node.

Object members

This object contains the following members:

Name	Description	Type
Bond10G	Configuration information for the Bond10G bonded interface.	network (bonded interfaces)
Bond1G	Configuration information for the Bond1G bonded interface.	network (bonded interfaces)
eth0-5	One object for each Ethernet interface in the storage node, describing configuration information for the interface. These objects are numbered 0 through 5 to match the interface name.	network (Ethernet interfaces)
lo	Configuration information for the loopback interface.	network (local interfaces)

Find more information

- GetConfig
- GetNetworkConfig

network (Ethernet interfaces)

The network (Ethernet interfaces) object contains configuration information for individual Ethernet interfaces. You can use the `GetConfig` and `GetNetworkConfig` methods to obtain this information for a storage node.

Object members

This object contains the following members:

Name	Description	Type
bond-master	Specifies which bonded interface this physical interface has joined as a bond slave.	string
family	Address family that the interface is configured to use. Currently "inet" for IPv4 is supported.	string

macAddress	The actual MAC address assigned to the interface and observed by the network.	string
macAddressPermanent	The immutable MAC address assigned by the manufacturer to the interface.	string
method	The method used to configure the interface. Possible values: • loopback: Used to define the IPv4 loopback interface. • manual: Used to define interfaces that are not configured automatically. • dhcp: Can be used to obtain an IP address via DHCP. • static: Used to define Ethernet interfaces with statically allocated IPv4 addresses.	string
status	The state of the interface. Possible values: • Down: The interface is inactive. • Up: The interface is ready, but has no link. • UpAndRunning: The interface is ready and a link is established.	string
upAndRunning	Indicates if the interface is ready and has a link.	boolean

Member modifiability and node states

This table indicates whether or not the object parameters can be modified at each possible node state.

Parameter name	Available state	Pending state	Active state
bond-master	No	No	No
family	No	No	No
macAddress	Configured by system	N/A	N/A

macAddressPermanent	Configured by system	N/A	N/A
method	No	No	No
status	Yes	Yes	Yes
upAndRunning	Configured by system	N/A	N/A

Find more information

- GetConfig
- GetNetworkConfig

network (local interfaces)

The network (local interfaces) object contains configuration information for local network interfaces, such as the loopback interface, on a storage node. You can use the `GetConfig` and `GetNetworkConfig` methods to obtain this information for a storage node.

Object members

This object contains the following members:

Name	Description	Type
family	Address family that the interface is configured to use. Currently "inet" for IPv4 is supported.	string
macAddress	The actual MAC address assigned to the interface and observed by the network.	string
macAddressPermanent	The immutable MAC address assigned by the manufacturer to the interface.	string

method	The method used to configure the interface. Possible values: • loopback: Used to define the IPv4 loopback interface. • manual: Used to define interfaces that are not configured automatically. • dhcp: Can be used to obtain an IP address via DHCP. • static: Used to define Ethernet interfaces with statically allocated IPv4 addresses.	string
status	The state of the interface. Possible values: • Down: The interface is inactive. • Up: The interface is ready, but has no link. • UpAndRunning: The interface is ready and a link is established.	string
upAndRunning	Indicates if the interface is ready and has a link.	boolean

Member modifiability and node states

This table indicates whether or not the object parameters can be modified at each possible node state.

Parameter name	Available state	Pending state	Active state
family	No	No	No
macAddress	Configured by system	N/A	N/A
macAddressPermanent	Configured by system	N/A	N/A
method	No	No	No
status	Yes	Yes	Yes
upAndRunning	Configured by system	N/A	N/A

Find more information

- GetConfig
- GetNetworkConfig

network (SNMP)

The SNMP network object contains information about SNMP v3 configuration for the cluster nodes.

Object members

This object contains the following members:

Name	Description	Type
access	The type of access allowed for SNMP information requests. Possible values: • ro: Read-only access. • rw: Read-write access. • rosys: Read-only access to a restricted set of system information.	string
cidr	A CIDR network mask. This network mask must be an integer greater than or equal to 0, and less than or equal to 32. It must also not be equal to 31.	integer
community	The SNMP community string.	string
network	This member, along with the cidr member, controls which network the access and community string apply to. The special value of "default" is used to specify an entry that applies to all networks. The CIDR mask is ignored when this member is either a host name or "default".	string

Find more information

GetSnmpInfo

networkInterface

The networkInterface object contains configuration information for individual network interfaces on a storage node.

Object members

This object contains the following members:

Name	Description	Type
address	The IPv4 management address of the interface.	string
addressV6	The IPv6 management address of the interface.	string
broadcast	The broadcast address of the interface.	string
macAddress	The MAC address of the interface.	string
mtu	The Maximum Transfer Unit, in bytes, of the interface.	integer
name	The name of the interface.	string
namespace	Whether or not this interface is assigned a virtual network namespace.	boolean
netmask	The subnet mask of the interface.	string
status	The operational status of the interface.	string
type	The type of interface (bond master, bond slave, etc).	string
virtualNetworkTag	The VLAN ID assigned to the interface on the virtual network.	integer

networkInterfaceStats

The networkInterfaceStats object contains network statistics, the total number of transmitted and received packets, and error information for individual network interfaces on a storage node. You can use the `ListNetworkInterfaceStats` API method to list this information for the network interfaces on a storage node.

Object members

This object contains the following members:

Name	Description	Type
collisions	The number of collisions detected.	integer
name	Name of the network interface.	string
rxBytes	The total number of bytes received.	integer
rxCrcErrors	The number of received packets that had a CRC error.	integer
rxDropped	The number of received packets that were dropped.	integer
rxErrors	The number of bad or malformed packets received.	integer
rxFifoErrors	The number of FIFO overrun errors in the received data.	integer
rxFrameErrors	The number of received packets with frame alignment errors.	integer
rxLengthErrors	The number of received packets with a length error.	integer
rxMissedErrors	The number of packets missed by the receiver.	integer
rxOverErrors	The number of receiver ring buffer overflow errors for this interface.	integer
rxPackets	The total number of packets received.	integer
txBytes	The total number of bytes transmitted.	integer
txCarrierErrors	The number of carrier errors for the transmit side.	integer
txErrors	The number of packet transmission errors.	integer
txFifoErrors	The number of FIFO overrun errors on the transmit side.	integer
txPackets	The total number of packets transmitted.	integer

node

The node object contains information about each node in the cluster. You can retrieve this information using the `ListActiveNodes` and `ListAllNodes` methods.

Object members

This object contains the following members:

Name	Description	Type
associatedFServiceID	The Fibre Channel service ID for the node. "0" if the node is not a Fibre Channel node.	integer
associatedMasterServiceID	Master service ID for the node.	integer
attributes	List of name-value pairs in JSON object format.	JSON object
chassisName	Uniquely identifies a chassis; identical for all nodes in a single chassis.	string
cip	The cluster IP address assigned to the node.	string
cipi	Network interface used for cluster communication.	string
customProtectionDomainName	Uniquely identifies a custom protection domain. This name is identical for all storage nodes within all chassis in a given custom protection domain.	string
fibreChannelTargetPortGroup	The target group associated with this node. "null" if the node is not a Fibre Channel node.	integer
maintenanceMode	Indicates which mode a node is in for maintenance.	n/a
mip	The IP address used for node management.	string
mipi	The network interface used for node management.	string
name	Host name for the node.	string
nodeID	NodeID for this node.	integer

Name	Description	Type
nodeSlot	For HCI platforms, the letter corresponding to the chassis slot this node is in ("A", "B", "C", or "D"). For storage platforms, this value is null.	string
platformInfo	Hardware information for the node. Members: • chassisType: The hardware platform of the node. • cpuModel: The CPU model of the hardware platform. • nodeMemoryGB: The amount of memory installed in the physical platform in GB. • nodeType: The node model name. • platformConfigVersion: The version of software configured for this node hardware.	JSON object
role	The node's role in the cluster. Possible values: • Management • Storage • Compute • Witness	
sip	The storage IP address assigned to the node.	string
sipi	The network interface used for storage traffic.	string
softwareVersion	Returns the current version of Element software running on the node.	string
uuid	The universally unique identifier associated with this node.	string
virtualNetworks	Object containing virtual network IP addresses and IDs.	virtualNetwork array

Find more information

- ListActiveNodes
- ListAllNodes

nodeProtectionDomains

The nodeProtectionDomains object contains information on the identify of a node and the protection domains associated with that node.

Object members

This object contains the following members:

Name	Description	Type
nodeID	Unique identifier for the node.	integer
protectionDomains	List of protection domains of which the node is a member.	protectionDomain

nodeStats

The nodeStats object contains high-level activity measurements for a node. You can use the `GetNodeStats` and `ListNodeStats` API methods to get some or all of the nodeStats objects.

Object members

This object contains the following members:

Name	Description	Type
count	The number of total samples in the nodeStats object.	integer
cpu	CPU usage, in %.	integer
cpuTotal	Monotonically increasing value of cpu utilization.	integer
cBytesIn	Bytes in on the cluster interface.	integer
cBytesOut	Bytes out on the cluster interface.	integer
sBytesIn	Bytes in on the storage interface.	integer
sBytesOut	Bytes out on the storage interface.	integer

Name	Description	Type
mBytesIn	Bytes in on the management interface.	integer
mBytesOut	Bytes out on the management interface.	integer
networkUtilizationCluster	Network interface utilization (in %) for the cluster network interface.	integer
networkUtilizationStorage	Network interface utilization (in %) for the storage network interface.	integer
readLatencyUSecTotal	Monotonically increasing value of total time spent performing read operations to the node.	integer
readOps	Monotonically increasing value of total read operations to a node.	integer
ssLoadHistogram	Histogram data illustrating slice service load over time.	JSON object
timestamp	The current time in UTC+0 format.	ISO 8601 date string
usedMemory	Total memory usage in bytes.	integer
writeLatencyUSecTotal	Monotonically increasing value of total time spent performing write operations to the node.	integer
writeOps	Monotonically increasing value of total write operations to a node.	integer

Find more information

- GetNodeStats
- ListNodeStats

ontapVersionInfo

The ontapVersionInfo object contains information about the API version of the ONTAP cluster in a SnapMirror relationship. The Element web UI uses the `GetOntapVersionInfo` API method to get this information.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
clientAPIMajorVesion	The ONTAP API major version in use by the Element API client.	string
clientAPIMinorVesion	The ONTAP API minor version in use by the Element API client.	string
ontapAPIMajorVersion	The current API major version supported by the ONTAP system.	string
ontapAPIMinorVesion	The current API minor version supported by the ONTAP system.	string
ontapVersion	The current software version running on the ONTAP cluster.	string

pendingActiveNode

The pendingActiveNode object contains information about a node that is currently in the pendingActive state, between the pending and active states. These are nodes that are currently being returned to the factory software image. Use the `ListPendingActiveNodes` API method to return a list of this information for all pendingActive nodes.

Object members

This object contains the following members:

Name	Description	Type
activeNodeKey	A unique key that allows the node to join the cluster automatically after a successful installation of software.	string
assignedNodeID	The assigned node ID for the node.	string
asyncHandle	The asynchronous method handle that you can use to query the status of the operation.	integer

Name	Description	Type
cip	The cluster IP address assigned to the node.	string
mip	The management IP address assigned to the node.	string
nodeSlot	For HCI platforms, the letter corresponding to the chassis slot this node is in ("A", "B", "C", or "D"). For storage platforms, this value is null.	string
pendingActiveNodeID	The pending node ID of the node.	integer
platformInfo	Hardware information for the node. Members: • chassisType: The hardware platform of the node. • cpuModel: The CPU model of the hardware platform. • nodeMemoryGB: The amount of memory installed in the physical platform in GB. • nodeType: The node model name. • platformConfigVersion: The version of software configured for this node hardware.	JSON object
role	The node's role in the cluster. Possible values: • Management • Storage • Compute • Witness	
sip	The storage (iSCSI) IP address assigned to the node.	string
softwareVersion	The current version of Element software running on the node.	string

Find more information

ListPendingActiveNodes

pendingNode

The pendingNode object contains information about a node that can be added to a cluster. Use the `ListPendingNodes` API method to return a list of this information for all pending nodes. You can add any of the listed nodes to a cluster using the `AddNodes` API method.

Object members

This object contains the following members:

Name	Description	Type
cipi	The cluster IP address assigned to the node.	string
activeNodeKey	A unique key that allows the node to join the cluster automatically after a successful installation of software.	string
assignedNodeID	The assigned node ID for the node.	string
asyncHandle	The asynchronous method handle that you can use to query the status of the operation.	integer
chassisName	Uniquely identifies a chassis; identical for all nodes in a single chassis.	string
cip	The cluster IP address assigned to the node.	string
mip	The management IP address assigned to the node.	string
nodeSlot	For HCI platforms, the letter corresponding to the chassis slot this node is in ("A", "B", "C", or "D"). For storage platforms, this value is null.	string
pendingActiveNodeID	The pending node ID of the node.	integer

Name	Description	Type
platformInfo	Hardware information for the node. Members: • chassisType: The hardware platform of the node. • cpuModel: The CPU model of the hardware platform. • nodeMemoryGB: The amount of memory installed in the physical platform in GB. • nodeType: The node model name. • platformConfigVersion: The version of software configured for this node hardware.	JSON object
role	The node's role in the cluster. Possible values: • Management • Storage • Compute • Witness	
sip	The storage (iSCSI) IP address assigned to the node.	string
softwareVersion	The current version of Element software running on the node.	string

Find more information

- AddNodes
- ListPendingNodes

protectionDomain

The protectionDomain object contains the name and type details for a protection domain.

Object members

This object contains the following members:

Name	Description	Type
protectionDomainName	The name of the protection domain.	string
protectionDomainType	The type of the protection domain. Possible values: • chassis: All storage nodes in a single chassis. • custom: All storage nodes in a single customer-defined protection domain.	string

protectionDomainLevel

The protectionDomainLevel object contains information about the storage cluster's current tolerance and resiliency levels. Tolerance levels indicate the cluster's ability to continue reading and writing data in the event of a failure, and resiliency levels indicate the cluster's ability to automatically heal itself from one or more failures within its associated type of protection domain.

Object members

This object contains the following members:

Name	Description	Type
protectionDomainType	The type of the protection domain which has the associated tolerance and resiliency. Possible values: • node: Any individual node. • chassis: Any individual node or all storage nodes in a single chassis. • custom: All storage nodes in a single customer-defined protection domain.	string
resiliency	The current resiliency of this cluster from the perspective of this protection domain type.	protectionDomainResiliency
tolerance	The current tolerance of this cluster from the perspective of this protection domain type.	protectionDomainTolerance

protectionDomainResiliency

The protectionDomainResiliency object contains the resiliency status of this storage cluster. Resiliency indicates the storage cluster's ability to automatically heal itself from one or more failures all within a single protection domain of its associated protection domain type. A storage cluster is considered healed when it can continue reading and writing data through the failure of any single storage node (a state known as node tolerance).

Object members

This object contains the following members:

Name	Description	Type
protectionSchemeResiliencies	A list of objects (one for each protection scheme) containing failure resiliency information for the associated type of protection domain.	protectionSchemeResiliency array
singleFailureThresholdBytesForBlockData	The maximum number of bytes that can be stored on the storage cluster before losing the ability to automatically heal to a state of node tolerance.	integer
sustainableFailuresForEnsemble	The predicted number of simultaneous failures that can occur without losing the ability to automatically heal to a state of node tolerance for the ensemble quorum.	integer

protectionDomainTolerance

The protectionDomainTolerance object contains information about the ability of the storage cluster to continue reading and writing data in the event of one or more failures all within a single protection domain of its associated protection domain type.

Object members

This object contains the following members:

Name	Description	Type
protectionSchemeTolerances	A list of objects (one for each protection scheme) containing failure tolerance information for the associated type of protection domain.	protectionSchemeTolerance array
sustainableFailuresForEnsemble	The number of simultaneous failures within the associated type of protection domain that can occur without losing the ensemble quorum.	integer

protectionSchemeResiliency

The protectionSchemeResiliency object contains information about whether a storage cluster, for a specific protection scheme, can automatically heal itself from one or more failures within its associated protectionDomainType. A storage cluster is considered healed when it can continue reading and writing data through the failure of any single storage node (a state known as node tolerance).

Object members

This object contains the following members:

Name	Description	Type
protectionScheme	The current protection scheme of this storage cluster. The only possible value is doubleHelix.	string
sustainableFailuresForBlockData	The predicted number of simultaneous failures which can occur without losing the ability to automatically heal to a state of node tolerance for data.	integer
sustainableFailuresForMetadata	The predicted number of simultaneous failures which can occur without losing the ability to automatically heal to a state of node tolerance for metadata.	integer

protectionSchemeTolerance

The protectionSchemeTolerance object contains information about whether a storage cluster, for a specific protection scheme, can continue to read and write data after failures.

Object members

This object contains the following members:

Name	Description	Type
protectionScheme	The current protection scheme of this storage cluster. The only possible value is doubleHelix.	string
sustainableFailuresForBlockData	The current number of simultaneous failures which can occur without losing block data availability for the associated protection scheme.	integer
sustainableFailuresForMetadata	The current number of simultaneous failures which can occur without losing metadata availability for the associated protection scheme.	integer

protocolEndpoint

The protocolEndpoint object contains the attributes of a protocol endpoint. You can retrieve this information for all protocol endpoints in the cluster using the `ListProtocolEndpoints` API method.

Object members

This object contains the following members:

Name	Description	Type
primaryProviderID	The ID of the primary protocol endpoint provider object for the protocol endpoint.	integer
protocolEndpointID	The unique ID of the protocol endpoint.	UUID

Name	Description	Type
protocolEndpointState	The status of the protocol endpoint. Possible values: • Active: The protocol endpoint is in use. • Start: The protocol endpoint is starting. • Failover: The protocol endpoint has failed over. • Reserved: The protocol endpoint is reserved.	string
providerType	The type of the protocol endpoint's provider. Possible values: • Primary • Secondary	string
scsiNAADeviceID	The globally unique SCSI device identifier for the protocol endpoint in NAA IEEE Registered Extended Format.	string
secondaryProviderID	The ID of the secondary protocol endpoint provider object for the protocol endpoint.	integer

Find more information

ListProtocolEndpoints

QoS

The QoS object contains information about Quality of Service (QoS) settings for volumes. Volumes created without specified QoS values are created using the default values. You can find default values using the `GetDefaultQoS` method.

Object members

This object contains the following members:

Name	Description	Type
burstIOPS	Maximum "peak" 4KB IOPS allowed for short periods of time. Allows for bursts of I/O activity over the normal maxIOPS value.	integer
burstTime	The length of time burstIOPS is allowed. The value returned is represented in seconds. This value is calculated by the system based on IOPS set for QoS.	integer
curve	The curve is a set of key-value pairs. The keys are I/O sizes in bytes. The values represent the cost of performing one IOP at a specific I/O size. The curve is calculated relative to a 4096 byte operation set at 100 IOPS.	JSON object
maxIOPS	The desired maximum 4KB IOPS allowed over an extended period of time.	integer
minIOPS	The desired minimum 4KB IOPS to guarantee. The allowed IOPS will only drop below this level if all volumes have been capped at their minIOPS value and there is still insufficient performance capacity.	integer

Find more information

GetDefaultQoS

QoSPolicy

The QoSPolicy object contains information about a QoS policy on a storage cluster running Element software.

Object members

This object contains the following members:

Name	Description	Type
qosPolicyID	A unique integer identifier for the QoSPolicy automatically assigned by the storage cluster.	integer

Name	Description	Type
name	The name of the QoS policy. For example: gold, platinum, or silver.	string
qos	The QoS settings that this policy represents.	QoS
volumeIDs	A list of volumes associated with this policy.	integer array

Find more information

GetQoSPolicy

remoteClusterSnapshotStatus

The `remoteClusterSnapshotStatus` object contains the UUID and status of a snapshot stored on a remote storage cluster. You can get this information with the `ListSnapshots` or `ListGroupSnapshots` API methods.

Object members

This object contains the following members:

Name	Description	Type
remoteStatus	The replication status of the remote snapshot on the target cluster as seen from the source cluster. Possible values: • Present: The snapshot exists on a remote cluster. • NotPresent: The snapshot does not exist on a remote cluster. • Syncing: This is a target cluster and it is currently replicating the snapshot. • Deleted: This is a target cluster. The snapshot has been deleted, and it still exists on the source.	string
volumePairUUID	The universal identifier of the volume pair.	UUID

schedule

The schedule object contains information about a schedule created to autonomously make a snapshot of a volume. You can retrieve schedule information for all schedules with the `ListSchedules` API method.

Object members

This object contains the following members:

Name	Description	Type
attributes	Indicates the frequency of the schedule occurrence. Possible values: • Day of Week • Day of Month • Time Interval	JSON object
hasError	Indicates whether or not the schedule has errors. Possible values: • true • false	boolean
hours	Shows the hours that will elapse before the next snapshot is created. Possible values are 0 through 24.	integer
lastRunStatus	Indicates the status of the last scheduled snapshot. Possible values: • Success • Failed	string
lastRunTimeStart	Indicates the last time the schedule started.	ISO 8601 date string
minutes	Shows the minutes that will elapse before the next snapshot is created. Possible values are 0 through 59.	integer
monthdays	Indicates the days of the month that a snapshot will be made.	array

Name	Description	Type
paused	Indicates whether or not the schedule is paused. Possible values: • true • false	boolean
recurring	Indicates whether or not the schedule is recurring. Possible values: • true • false	boolean
runNextInterval	Indicates whether or not the schedule will run the next time the scheduler is active. When true, the schedule will run the next time the scheduler is active and then this value is set back to false. Possible values: • true • false	boolean
scheduleID	The unique ID of the schedule.	integer

Name	Description	Type
scheduleInfo	Includes the unique name given to the schedule, the retention period for the snapshot that was created, and the volume ID of the volume from which the snapshot was created. Valid values: • `enableRemoteReplication`: Indicates if the snapshot should be included in remote replication. (boolean) • `ensureSerialCreation`: Specifies whether a new snapshot creation should be allowed if a previous snapshot replication is in progress. (boolean) • `name`: The snapshot name to be used. (string) • `retention`: The amount of time the snapshot is retained. Depending on the time, it displays in one of the following formats: ◦ `fifo`: The snapshot is retained on a First-In-First-Out (FIFO) basis. If empty, the snapshot is retained forever. (string) ◦ "HH:mm:ss" • `volumeID`: The ID of the volume to be included in the snapshot. (integer) • `volumes`: A list of volume IDs to be included in the group snapshot. (integer array)	JSON object
scheduleName	The unique name assigned to the schedule.	string
scheduleType	Only schedule types of snapshot are supported at this time.	string

Name	Description	Type
snapMirrorLabel	The snapMirrorLabel to be applied to the created Snapshot or Group Snapshot, contained in the scheduleInfo. If not set, this value is null.	string
startingDate	Indicates the date the first time the schedule began or will begin; formatted in UTC time.	ISO 8601 date string
toBeDeleted	Indicates if the schedule is marked for deletion. Possible values: • true • false	boolean
weekdays	Indicates the days of the week that a snapshot will be made.	array

Find more information

ListSchedules

session (Fibre Channel)

The session object contains information about each Fibre Channel session that is visible to the cluster and what target ports it is visible on. You can retrieve this information with the `ListFibreChannelSessions` API method.

Object members

This object contains the following members:

Name	Description	Type
initiatorWWPN	The World Wide Port Name (WWPN) of the initiator that is logged into the target port.	string
nodeID	The node that owns the Fibre Channel session.	integer

Name	Description	Type
initiator	Information about this Fibre Channel session's server initiator. Members: • alias: The friendly name assigned to the initiator. • attributes: The attributes of this initiator. • initiatorID: The ID of this initiator. • initiatorName: The name of this initiator. • volumeAccessGroups: A list of volume access groups associated with this initiator.	JSON object
serviceID	The service ID of the target port involved in this session.	integer
targetWWPN	The WWPN of the target port involved in this session.	string
volumeAccessGroupID	The ID of the volume access group to which the initiatorWWPN belongs. If not in a volume access group, this value is null.	integer

Find more information

ListFibreChannelSessions

session (iSCSI)

The session (iSCSI) object contains detailed information about each volume's iSCSI session. You can retrieve iSCSI session information with the `ListISCSISessions` API method.

Object members

This object contains the following members:

Name	Description	Type
accountID	The account ID of the account used for CHAP authentication, if any.	integer

Name	Description	Type
accountName	The name of the account used for CHAP authentication, if any.	string
authentication	Authentication information for this iSCSI session.	ISCSIAuthentication
createTime	The time of the creation of the iSCSI session, in UTC+0 format.	ISO 8601 date string
driveID	The driveID associated with the transport service hosting the session.	integer
driveIDs	A list of the driveIDs of the drives reporting the failure. An empty list if not applicable.	integer array
initiator	Information about this iSCSI session's server initiator. Members: • alias: The friendly name assigned to the initiator. • attributes: The attributes of this initiator. • initiatorID: The ID of this initiator. • initiatorName: The name of this initiator. • volumeAccessGroups: A list of volume access groups associated with this initiator.	JSON object
initiatorIP	The IP address and port number of the iSCSI server initiator.	string
initiatorName	The iSCSI Qualified Name (IQN) of the iSCSI server initiator.	string
initiatorPortName	The initiatorName combined with the initiatorSessionID; identifies the initiator port.	string
initiatorSessionID	A 48-bit ID provided by the initiator that identifies the iSCSI session as belonging to that initiator.	integer

Name	Description	Type
msSinceLastIscsiPDU	The time, in milliseconds, since the last iSCSI PDU was received for this session.	integer
msSinceLastScsiCommand	The time, in milliseconds, since the last SCSI command was received for this session.	integer
nodeID	The nodeID associated with the transport service hosting the session.	integer
serviceID	The serviceID of the transport service hosting the session.	integer
sessionID	The iSCSI session ID.	integer
targetIP	The IP address and port number of the iSCSI storage target.	string
targetName	The IQN of the iSCSI target.	string
targetPortName	The targetName combined with the target portal group tag; identifies the target port.	string
virtualNetworkID	The virtual network ID associated with the session.	integer
volumeID	The volumeID of the volume associated with the session, if any.	integer
volumeInstance	Identifies the volume object associated with the iSCSI session, if any.	integer

Find more information

ListISCSISessions

snapMirrorAggregate

The snapMirrorAggregate object contains information about the available ONTAP aggregates, which are collections of disks made available to volumes as storage. You can get this information using the ListSnapMirrorAggregates API method.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
aggregateName	The name of the aggregate.	string
nodeName	The name of the ONTAP node that owns this aggregate.	string
sizeAvailable	The number of available bytes remaining in the aggregate.	integer
sizeTotal	The total size (in bytes) of the aggregate.	integer
percentUsedCapacity	The percentage of disk space currently in use.	integer
volumeCount	The number of volumes in the aggregate.	integer

snapMirrorClusterIdentity

The snapMirrorClusterIdentity object contains identification information about the remote ONTAP cluster in a SnapMirror relationship.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
clusterName	The name of the destination ONTAP cluster.	string
clusterUUID	The 128-bit universally-unique identifier of the destination ONTAP cluster.	string

Name	Description	Type
clusterSerialNumber	The serial number of the destination ONTAP cluster.	string

snapMirrorEndpoint

The snapMirrorEndpoint object contains information about the remote SnapMirror storage systems communicating with the Element storage cluster. You can retrieve this information with the ListSnapMirrorEndpoints API method.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The unique identifier for the object in the local cluster.	integer
managementIP	The cluster management IP address of the endpoint.	string
clusterName	The ONTAP cluster name. This value is automatically populated with the value of “clusterName” from the snapMirrorClusterIdentity object.	string
username	The management user name for the ONTAP system.	string
ipAddresses	List of the inter-cluster storage IP addresses for all nodes in the cluster. You can get these IP addresses with the ListSnapMirrorNetworkInterfaces method.	string array
isConnected	The connectivity status of the control link to the ONTAP cluster.	boolean

snapMirrorJobScheduleCronInfo

The snapMirrorJobScheduleCronInfo object contains information about a cron job schedule on the ONTAP system.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
jobScheduleName	The name of the job schedule.	string
jobScheduleDescription	An automatically-generated human-readable summary of the schedule.	string

snapMirrorLunInfo

The snapMirrorLunInfo object contains information about the ONTAP LUN object.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
creationTimestamp	The creation time of the LUN.	ISO 8601 date string
lunName	The name of the LUN.	string
path	The path of the LUN.	string
size	The size of the LUN in bytes.	integer
sizeUsed	The number of bytes used by the LUN.	integer
state	The current access state of the LUN. Possible values: • online • offline • foreign_lun_error • nvfail • space_error	string

Name	Description	Type
volume	The name of the volume that contains the LUN.	string
vserver	The Vserver that contains the LUN.	string

snapMirrorNetworkInterface

The snapMirrorNetworkInterface object contains information about the intercluster Logical Interfaces (LIFs).

Object members

This object contains the following members:

Name	Description	Type
administrativeStatus	Whether the logical interface (LIF) is administratively enabled or disabled. Possible values: • up • down	string
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
interfaceName	The LIF name.	string
networkAddress	The IP address of the LIF.	string
networkMask	The network mask of the LIF.	string
interfaceRole	The role of the LIF. Possible values: • undef • cluster • data • node_mgmt • intercluster • cluster_mgmt	string

Name	Description	Type
operationalStatus	The operational state of the LIF (whether or not it has formed a successful connection). This status can differ from the administrative status if there is a network problem that prevents the interface from functioning. Possible values: • up • down	string
vserverName	The name of the Vserver.	string

snapMirrorNode

The snapMirrorNode object contains information about the nodes of the destination ONTAP cluster in a SnapMirror relationship.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
name	The name of the ONTAP node.	string
model	The model of the ONTAP node.	string
serialNumber	The serial number of the ONTAP node.	string
productVersion	The ONTAP product version.	string
isNodeHealthy	The health of a node in the ONTAP cluster. Possible values: • true • false	string

Name	Description	Type
isNodeEligible	Whether or not the node is eligible to participate in an ONTAP cluster. Possible values: • true • false	string

snapMirrorPolicy

The snapMirrorPolicy object contains information about a SnapMirror policy that is stored on an ONTAP system.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
policyName	The unique name assigned to the policy.	string
policyType	The type of policy. Possible values: • async_mirror • mirror_vault	string
comment	A human-readable description associated with the SnapMirror policy.	string
transferPriority	The priority at which a SnapMirror transfer runs. Possible values: • normal: The default priority. These transfers are scheduled before most low priority transfers. • low: These transfers have the lowest priority and are scheduled after most normal priority transfers.	string
policyRules	A list of objects describing the policy rules.	snapMirrorPolicyRule array

Name	Description	Type
totalKeepCount	The total retention count for all rules in the policy.	integer
totalRules	The total number of rules in the policy.	integer
vserverName	The name of the Vserver for the SnapMirror policy.	string

snapMirrorPolicyRule

The snapMirrorPolicyRule object contains information about the rules in a SnapMirror policy.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorLabel	The snapshot copy label, used for snapshot copy selection in extended data protection relationships.	string
keepCount	Specifies the maximum number of snapshot copies that are retained on the SnapMirror destination volume for a rule.	integer

snapMirrorRelationship

The snapMirrorRelationship object contains information about a SnapMirror relationship between a Element volume and an ONTAP volume.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer

Name	Description	Type
snapMirrorRelationshipID	The unique identifier for each snapMirrorRelationship object in an array as would be returned in ListSnapMirrorRelationships. This UUID is created and returned from the ONTAP system.	string
sourceVolume	An object describing the source volume.	snapMirrorVolumeInfo
destinationVolume	An object describing the destination volume.	snapMirrorVolumeInfo
currentMaxTransferRate	The current maximum transfer rate between the source and destination volumes, in kilobytes per second.	integer
isHealthy	Whether the relationship is healthy or not. Possible values: • true: The relationship is healthy. • false: The relationship is not healthy. This can be caused by a manual or scheduled update failing or being aborted, or by the last scheduled update being delayed.	boolean
lagtime	The amount of time in seconds by which the data on the destination volume lags behind the data on the source volume.	integer
lastTransferDuration	The amount of time in seconds it took for the last transfer to complete.	integer
lastTransferError	A message describing the cause of the last transfer failure.	string
lastTransferSize	The total number of bytes transferred during the last transfer.	integer
lastTransferEndTimestamp	The timestamp of the end of the last transfer.	ISO 8601 date string

Name	Description	Type
lastTransferType	The type of the previous transfer in the relationship.	string
maxTransferRate	Specifies the maximum data transfer rate between the volumes in kilobytes per second. The default value, 0, is unlimited and permits the SnapMirror relationship to fully utilize the available network bandwidth.	integer
mirrorState	The mirror state of the SnapMirror relationship. Possible values: • uninitialized: The destination volume has not been initialized. • snapmirrored: The destination volume has been initialized and is ready to receive SnapMirror updates. • broken-off: The destination volume is read-write and snapshots are present.	string
newestSnapshot	The name of the newest Snapshot copy on the destination volume.	string
policyName	Specifies the name of the ONTAP SnapMirror policy for the relationship. A list of available policies can be retrieved with ListSnapMirrorPolicies. Example values are “MirrorLatest” and “MirrorAndVault”.	string
policyType	The type of the ONTAP SnapMirror policy for the relationship. See ListSnapMirrorPolicies. Examples are: “async_mirror” or “mirror_vault”.	string

Name	Description	Type
relationshipProgress	The total number of bytes that have been processed so far for the current activity of the relationship as returned in the relationship-status. This is set only when the "relationshipStatus" member indicates that an activity is in progress.	integer
relationshipStatus	The status of the SnapMirror relationship. Possible values: • idle • transferring • checking • quiescing • quiesced • queued • preparing • finalizing • aborting • breaking	string
relationshipType	The type of the SnapMirror relationship. On storage clusters running Element software, this value is always “extended_data_protection”.	string
scheduleName	The name of the pre-existing cron schedule on the ONTAP system that is used to update the SnapMirror relationship. A list of available schedules can be retrieved with ListSnapMirrorSchedules.	string
unhealthyReason	The reason the relationship is not healthy.	string

snapMirrorVolume

The snapMirrorVolume object contains information about an ONTAP volume.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
name	The name of the volume.	string
type	The type of volume. Possible values: • rw: Read-write volume • ls: Load-sharing volume • dp: Data protection volume	string
vserver	The name of the Vserver that owns this volume.	string
aggrName	The containing aggregate name.	string
state	The state of volume. Possible values: • online • restricted • offline • mixed	string
size	The total filesystem size (in bytes) of the volume.	string
availSize	The size (in bytes) of the available space in the volume.	string

snapMirrorVolumeInfo

The snapMirrorVolumeInfo object contains information about a volume location in a SnapMirror relationship, such as its name and type.

Object members

This object contains the following members:

Name	Description	Type
type	The type of volume. Possible values: • solidfire: The volume resides on a storage cluster running Element software. • ontap: The volume resides on a remote ONTAP cluster.	string
volumeID	The ID of the volume. Only valid if "type" is solidfire.	integer
vserver	The name of the Vserver that owns this volume. Only valid if "type" is ontap.	string
name	The name of the volume.	string

snapMirrorVserver

The snapMirrorVserver object contains information about the Storage Virtual Machines (or Vservers) at the destination ONTAP cluster.

Object members

This object contains the following members:

Name	Description	Type
snapMirrorEndpointID	The ID of the destination ONTAP system.	integer
vserverName	The name of the Vserver.	string
vserverType	The type of Vserver. Possible values: • data • admin • system • node	string

Name	Description	Type
vserverSubtype	The subtype of the Vserver. Possible values: • default • dp_destination • data • sync_source • sync_destination	string
rootVolume	The root volume of the Vserver.	string
rootVolumeAggregate	The aggregate on which the root volume will be created.	string
vserverAggregateInfo	An array of snapMirrorVserverAggregateInfo objects.	JSON object
adminState	The detailed administrative state of the Vserver. Possible values: • running • stopped • starting • stopping • initializing • deleting	string
operationalState	The basic operational state of the Vserver. Possible values: • running • stopped	string

snapMirrorVserverAggregateInfo

The snapMirrorVserverAggregateInfo object contains information about the available data Storage Virtual Machines (also called Vservers) at the destination ONTAP cluster.

Object members

This object contains the following members:

Name	Description	Type
aggrName	The name of the aggregate assigned to a Vserver.	string
aggrAvailSize	The assigned aggregate's available size.	integer

snapshot

The snapshot object contains information about a snapshot made for a volume. You can use the `ListSnapshots` API method to retrieve a list of snapshot information for a volume or for all volumes. The object includes information about the active snapshot as well as each snapshot created for a volume.

Object members

This object contains the following members:

Name	Description	Type
attributes	List of name-value pairs in JSON object format.	JSON object
checksum	A small string representation of the data in the stored snapshot. This checksum can be used later to compare other snapshots to detect errors in the data.	string
createTime	The UTC+0 formatted time the snapshot was created.	ISO 8601 date string
enableRemoteReplication	Identifies if snapshot is enabled for remote replication.	boolean
expirationReason	Indicates how the snapshot expiration is set. Possible values: • Api: The expiration time is set by using the API. • None: No expiration time is set. • Test: The expiration time is set for testing. • fifo: Expiration occurs on a first-in-first-out basis.	string

Name	Description	Type
expirationTime	The time at which this snapshot will expire and be purged from the cluster.	ISO 8601 date string
groupID	The group ID if the snapshot is a member of a group snapshot.	integer
groupsnapshotUUID	Contains information about each snapshot in the group. Each of these members will have a UUID parameter for the snapshot's UUID.	string
instanceCreateTime	The time that the snapshot was created on the local cluster.	ISO 8601 date string
instanceSnapshotUUID	The universally unique ID of the snapshot on the local cluster. This ID does not get replicated to other clusters.	string
name	The unique name assigned to the snapshot. If no name is specified, the name is the UTC+0 formatted timestamp of when the snapshot was created.	string
remoteStatuses	An array containing the universal identifier and replication status of each remote snapshot on the target cluster as seen from the source cluster.	remoteClusterSnapshotStatus array
snapMirrorLabel	The label used by SnapMirror software to specify snapshot retention policy on SnapMirror endpoints. If not set, this value is null.	string
snapshotID	The unique ID of an existing snapshot.	string
snapshotUUID	The universally unique ID of an existing snapshot. When the snapshot is replicated across clusters, this ID is replicated along with it and is used to identify the snapshot across clusters.	string

Name	Description	Type
status	Current status of the snapshot. Possible values: • Unknown: There was an error obtaining the status of the snapshot. • Preparing: This snapshot is being prepared for use and is not yet writable. • RemoteSyncing: This snapshot is being replicated from a remote cluster. • Done: This snapshot has finished preparation or replication and is now usable. • Active: This snapshot is the active branch. • Cloning: This snapshot is involved in a CopyVolume operation.	string
totalSize	The total size in bytes of the snapshot.	integer
virtualVolumeID	The ID of the virtual volume associated with this snapshot.	UUID
volumeID	The ID of the volume the snapshot was created from.	integer
volumeName	The name of the volume at the time the snapshot was created.	string

Find more information

ListSnapshots

snmpTrapRecipient

The snmpTrapRecipient object contains information about a host that is configured to receive SNMP traps generated by the storage cluster. You can use the `GetSnmpTrapInfo` API method to get a list of hosts configured to receive SNMP traps.

Object members

This object contains the following members:

Name	Description	Type
host	The IP address or host name of the target host.	string
port	The UDP port number on the host where the trap should be sent. Valid range is 1 through 65535. 0 (zero) is not a valid port number. The default port is 162.	integer
community	SNMP community string.	string

storageContainer

The storageContainer object contains the attributes of a virtual volume storage container. You can retrieve this information for each storage container in the cluster using the `ListStorageContainers` API method.

Object members

This object contains the following members:

Name	Description	Type
accountID	The ID of the storage system account associated with the storage container.	integer
initiatorSecret	The CHAP authentication secret for the initiator associated with the storage container.	string
name	The name of the storage container.	string
protocolEndpointType	The storage container's protocol endpoint type. SCSI is the only valid value.	string
status	The status of the storage container. Possible values: • Active: The storage container is in use. • Locked: The storage container is locked.	string

Name	Description	Type
storageContainerID	The unique ID of the storage container.	UUID
targetSecret	The CHAP authentication secret for the target associated with the storage container.	string
virtualVolumes	A list of IDs of the virtual volumes associated with the storage container.	UUID array

Find more information

ListStorageContainers

syncJob

The syncJob object contains information about clone, remote replication, or slice synchronization jobs that are running on a cluster.

You can retrieve synchronization information with the `ListSyncJobs` API method.

Object members

This object contains the following members:

Name	Description	Type
blocksPerSecond	The number of data blocks being transferred per second from the source cluster to the target cluster. Present only if the type member is set to remote.	integer
branchType	Returned for remote replication sync jobs only. Possible values: • snapshot • volume	string
bytesPerSecond	The number of bytes the clone is processing per second. Present only if the type member is set to clone or slice.	float

Name	Description	Type
cloneID	The identifier of the clone operation that is in progress. Present only if the type member is set to clone.	integer
currentBytes	The number of bytes the clone has processed in the source volume. Present only if the type member is set to clone or slice.	integer
dstServiceID	The service identifier hosting the primary replica for the volume. Present only if the type member is set to remote.	integer
dstVolumeID	The destination volume ID. Present only if the type member is set to clone or remote.	integer
elapsedTime	The time elapsed, in seconds, since the sync job started.	float or integer depending on the type of sync operation
groupCloneID	The ID of the group clone operation that is in progress.	integer
nodeID	Specifies the node the clone is occurring on. Present only if the type member is set to clone.	integer
percentComplete	The percentage of sync job completion.	float or integer depending on the type of sync operation
remainingTime	The estimated time, in seconds, to complete the operation.	float
sliceID	The ID of the slice drive being synced.	integer

Name	Description	Type
stage	Present only if the type member is set to remote or clone. Possible values: • metadata: Replication is in the process of determining what data needs to be transferred to the remote cluster. Status is not reported for this stage of the replication process. • data: Replication is in the process of transferring the bulk of the data to the remote cluster. • whole: Indicates backward compatibility of the slice for slice sync jobs.	string
snapshotID	The ID of the snapshot the clone was created from. Present only if the type member is set to clone.	integer
srcServiceID	The source service ID.	integer
srcVolumeID	The source volume ID.	integer
totalBytes	The total number of bytes of the clone. Present only if the type member is set to clone or slice.	integer
type	The type of sync operation. Possible values: • clone • slice • block • remote	string

Find more information

ListSyncJobs

task (virtual volumes)

The task object contains information about a currently running or finished virtual volume task in the system. You can use the `ListVirtualVolumeTasks` method to retrieve this

information for all virtual volume tasks.

Object members

This object contains the following members:

Name	Description	Type
cancelled	Indicates whether or not the task was cancelled. Possible values: • true • false	boolean
cloneVirtualVolumeID	The unique virtual volume ID of the virtual volume being cloned (for clone tasks).	UUID
parentMetadata	An object containing metadata of the parent for tasks which clone or create snapshots of a virtual volume.	JSON object
parentTotalSize	The total space available (in bytes) on the parent for clone or snapshot tasks.	integer
parentUsedSize	The used space of the parent (in bytes) for clone or snapshot tasks.	integer

Name	Description	Type
operation	The type of operation the task is performing. Possible values: • unknown: The task operation is unknown. • prepare: The task is preparing a virtual volume. • snapshot: The task is creating a snapshot of a virtual volume. • rollback: The task is rolling back a virtual volume to a snapshot. • clone: The task is creating a clone of the virtual volume. • fastClone: The task is creating a fast clone of a virtual volume. • copyDiffs: The task is copying differing blocks to a virtual volume.	string
status	The current status of the virtual volume task. Possible values: • Error: The task has failed and returned an error. • Queued: The task is waiting to be run. • Running: The task is currently running. • Success: The task has completed successfully.	string
virtualVolumeHostID	The unique ID of the host that started the task.	UUID
virtualVolumeID	The new, unique virtual volume ID (for tasks that create a new virtual volume).	UUID
virtualVolumeTaskID	The unique ID of the task.	UUID

Find more information

ListVirtualVolumeTasks

usmUser

You can use the SNMP usmUser object with the `SetSnmpInfo` API method to configure SNMP on the storage cluster.

Object members

This object contains the following members:

Name	Description	Type
access	The type of SNMP access for this user. Possible values: • rouser: Read-only access. • rwuser: Read-write access. All Element software MIB objects are read-only.	string
name	The name of the user.	string
password	The password of the user.	string
passphrase	The passphrase of the user.	string
secLevel	The type of credentials required for this user. Possible values: • noauth: No password or passphrase is required. • auth: A password is required for user access. • priv: A password and passphrase are required for user access.	string

Find more information

SetSnmpInfo

virtualNetwork

The virtualNetwork object contains information about a specific virtual network. You can use the `ListVirtualNetworks` API method to retrieve a list of this information for all virtual networks in the system.

Object members

This object contains the following members:

Name	Description	Type
addressBlocks	The range of address blocks currently assigned to the virtual network. Members: • available: Binary string in "1"s and "0"s. "1" denotes that the IP address is available, and "0" denotes that the IP is not available. The string is read from right to left with the digit to the far right being the first IP address in the list of address blocks. • size: The size of this block of addresses. • start: The first IP address in the block.	JSON object array
attributes	List of name-value pairs in JSON object format.	JSON object
name	The name assigned to the virtual network.	string
netmask	The IP address of the netmask for the virtual network.	string
svip	The storage IP address for the virtual network.	string
gateway	The gateway used for the virtual network.	string
virtualNetworkID	The unique identifier for a virtual network.	integer
virtualNetworkTag	The VLAN tag identifier.	integer

Find more information

ListVirtualNetworks

virtualVolume

The virtualVolume object contains configuration information about a virtual volume as well as information about snapshots of the virtual volume. It does not include runtime or usage information. You can use the `ListVirtualVolumes` method to retrieve this information

for a cluster.

Object members

This object contains the following members:

Name	Description	Type
bindings	A list of binding IDs for this virtual volume.	UUID array
children	A list of virtual volume UUIDs that are children of this virtual volume.	UUID array
descendants	When you pass recursive: true to the ListVirtualVolumes method, contains a list of virtual volume UUIDs that are descendants of this virtual volume.	UUID array
metadata	Key-value pairs of the virtual volume's metadata, such as virtual volume type, guest OS type, and so on.	JSON object
parentVirtualVolumeID	The virtual volume ID of the parent virtual volume. If the ID is all zeros, this is an independent virtual volume with no link to a parent.	UUID
snapshotID	The ID of the underlying volume snapshot. This value is "0" if the virtual volume does not represent a snapshot.	integer
snapshotInfo	The snapshot object for the associated snapshot (null if nonixestent).	snapshot
status	Current status of the virtual volume. Possible values: • cloning: The virtual volume is being processed in response to a clone or snapshot operation. • waiting: The virtual volume is waiting for a snapshot operation to complete. • ready: The virtual volume is ready for general purpose use.	string

Name	Description	Type
storageContainer	An object describing the storage container that owns this virtual volume.	storageContainer
virtualVolumeID	The unique ID of the virtual volume.	UUID
virtualVolumeType	The type of the virtual volume.	string
volumeID	The ID of the underlying volume.	integer
volumeInfo	When you pass details: true to the ListVirtualVolumes method, this member is an object describing the volume.	volume

Find more information

- ListVirtualVolumes
- snapshot
- storageContainer
- volume

volume

The volume object contains configuration information about unpaired or paired volumes. It does not include runtime or usage information, and does not contain information about virtual volumes.

Object members

This object contains the following members:

Name	Description	Type
access	The type of access allowed for the volume. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. • `replicationTarget`: Designated as a target volume in a replicated volume pair.	string
accountID	The accountID of the account containing the volume.	integer
attributes	List of name-value pairs in JSON object format.	JSON object
blockSize	The size of blocks on the volume.	integer
createTime	The UTC+0 formatted time the volume was created.	ISO 8601 string
currentProtectionScheme	The protection scheme that is being used for this volume. If a volume is converting from one protection scheme to another, this member reflects the protection scheme to which the volume is converting.	string
deleteTime	The UTC+0 formatted time the volume was deleted.	ISO 8601 string
enable512e	If set to true, the volume provides 512 byte sector emulation.	boolean
enableSnapMirrorReplication	Whether or not the volume can be used for replication with SnapMirror endpoints.	boolean
fifoSize	Specifies the maximum number of snapshots of the volume to be maintained simultaneously if using the First-In-First-Out (FIFO) snapshot retention mode.	integer

Name	Description	Type
iqn	The iSCSI Qualified Name of the volume.	string
lastAccessTime	The last time any access (including I/O) to the volume occurred (formatted as UTC+0). If the last access time is not known, this value is null.	ISO 8601 string
lastAccessTimeIO	The last time any I/O to the volume occurred (formatted as UTC+0). If the last access time is not known, this value is null.	ISO 8601 string
minFifoSize	Specifies the minimum number of First-In-First-Out (FIFO) snapshot slots reserved simultaneously by the volume if using the First-In-First-Out (FIFO) snapshot retention mode.	integer
name	The name of the volume as provided at creation time.	string
previousProtectionScheme	If a volume is converting from one protection scheme to another, this member reflects the protection scheme from which the volume is converting. This member does not change until a conversion is started. If a volume has never been converted, this member is null.	string
purgeTime	The UTC+0 formatted time the volume was purged from the system.	ISO 8601 string
qos	The quality of service settings for this volume.	QoS
qosPolicyID	The QoS policy ID associated with the volume. The value is null if the volume is not associated with a policy.	integer
scsiEUIDeviceID	Globally unique SCSI device identifier for the volume in EUI-64 based 16-byte format.	string

Name	Description	Type
scsiNAADeviceID	Globally unique SCSI device identifier for the volume in NAA IEEE Registered Extended format.	string
sliceCount	The number of slices on the volume. This value is always "1".	integer
status	The current status of the volume. Possible values: • init: A volume that is being initialized and is not ready for connections. • active: An active volume ready for connections. • deleted: A volume that has been marked for deletion, but not yet purged.	string
totalSize	The total bytes of provisioned capacity.	integer
virtualVolumeID	The unique virtual volume ID associated with the volume, if any.	UUID
volumeAccessGroups	List of IDs pf volume access groups to which a volume belongs. This value is an empty list if a volume does not belong to any volume access groups.	integer array
volumeConsistencyGroupUUID	The universally unique ID of the volume consistency group of which the volume is a member.	UUID
volumeID	The unique volumeID for the volume.	integer
volumePairs	Information about a paired volume. Visible only if a volume is paired. This value is an empty list if the volume is not paired.	volumePair array
volumeUUID	The universally unique ID of the volume.	UUID

Find more information

- ListActiveVolumes
- ListDeletedVolumes
- ListVolumes
- ListVolumesForAccount
- QoS

volumeAccessGroup

The volumeAccessGroup object contains information about a specific volume access group. You can retrieve a list of this information for all access groups with the API method `ListVolumeAccessGroups`.

Object members

This object contains the following members:

Name	Description	Type
attributes	List of name-value pairs in JSON object format.	JSON object
deletedVolumes	Array of volumes that have been deleted from the volume access group that have not yet been purged from the system.	integer array
initiatorIDs	A list of IDs of initiators that are mapped to the volume access group.	integer array
initiators	Array of unique IQN/WWPN initiators that are mapped to the volume access group.	string array
name	Name of the volume access group.	string
volumeAccessGroupID	Unique VolumeAccessGroupID identifier for the volume access group.	integer
volumes	A list of VolumeIDs belonging to the volume access group.	integer array

Find more information

ListVolumeAccessGroups

volumePair

The volumePair object contains information about a volume that is paired with another volume on a different cluster. If the volume is not paired, this object is empty. You can use the `ListActivePairedVolumes` and `ListActiveVolumes` API methods to return information about paired volumes.

Object members

This object contains the following members:

Name	Description	Type
clusterPairID	The cluster on which the volume is paired.	integer
remoteReplication	Details on volume replication. Members: • mode: (string) One of "Async", "Sync", or "SnapshotsOnly". • pauseLimit: (integer) Internal use only. • remoteServiceID: (integer) The remote slice service ID. • resumeDetails: (string) Reserved for future use. • snapshotReplication (JSON object) ◦ state: (string) The state of the ongoing snapshot replication, if one is in progress. ◦ stateDetails: (string) Reserved for future use. • state: (string) The state of the volume replication. • stateDetails: (string) Reserved for future use.	JSON object
remoteSliceID	The cluster-defined slice ID on the remote cluster.	integer
remoteVolumeID	The ID of the volume on the remote cluster that the local volume is paired with.	integer

Name	Description	Type
remoteVolumeName	The name of the remote volume.	string
volumePairUUID	A universally unique, cluster-defined identifier for this pairing in a canonical format.	string

Find more information

- ListActivePairedVolumes
- ListActiveVolumes

volumeStats

The volumeStats object contains statistical data for an individual volume.

Object members

You can use the following methods to get volumeStats objects for some or all volumes:

- GetVolumeStats
- ListVolumeStatsByAccount
- ListVolumeStatsByVolume
- ListVolumeStatsByVolumeAccessGroup

This object contains the following members:

Name	Description	Calculation	Type
accountID	The ID of the account of the volume owner.	N/A	integer
actualIOPS	The current actual IOPS to the volume in the last 500 milliseconds.	Point in time	integer

Name	Description	Calculation	Type
asyncDelay	The length of time since the volume was last synced with the remote cluster. If the volume is not paired, this is null. **Note:** A target volume in an active replication state always has an asyncDelay of 0 (zero). Target volumes are system-aware during replication and assume asyncDelay is accurate at all times.	N/A	ISO 8601 duration string or null
averageIOPSize	The average size in bytes of recent I/O to the volume in the last 500 milliseconds.	Point in time	integer
burstIOPSCredit	The total number of IOP credits available to the user. When volumes are not using up to the configured maxIOPS, credits are accrued.	N/A	integer
clientQueueDepth	The number of outstanding read and write operations to the volume.	N/A	integer
desiredMetadataHosts	The metadata (slice) services being migrated to if the volume metadata is being migrated between metadata services. A "null" value means the volume is not migrating.	N/A	JSON object
latencyUSec	The average time, in microseconds, to complete operations to the volume in the last 500 milliseconds. A "0" (zero) value means there is no I/O to the volume.	Point in time	integer

Name	Description	Calculation	Type
metadataHosts	The metadata (slice) services on which the volume metadata resides. Possible values: • primary: The primary metadata services hosting the volume. • liveSecondaries: Secondary metadata services that are currently in a "live" state. • deadSecondaries: Secondary metadata services that are in a dead state.	N/A	JSON object
normalizedIOPS	Average number of IOPS for the entire cluster in the last 500 milliseconds.	Point in time	integer
nonZeroBlocks	The total number of 4KiB blocks that contain data after the last garbage collection operation has completed.	N/A	integer
readBytes	The total cumulative bytes read from the volume since the creation of the volume.	Monotonically increasing	integer
readBytesLastSample	The total number of bytes read from the volume during the last sample period.	Point in time	integer
readLatencyUSec	The average time, in microseconds, to complete read operations to the volume in the last 500 milliseconds.	Point in time	integer
readLatencyUSecTotal	The total time spent performing read operations from the volume.	Monotonically increasing	integer

Name	Description	Calculation	Type
readOps	The total read operations to the volume since the creation of the volume.	Monotonically increasing	integer
readOpsLastSample	The total number of read operations during the last sample period.	Point in time	integer
samplePeriodMSec	The length of the sample period, in milliseconds.	N/A	integer
throttle	A floating value between 0 and 1 that represents how much the system is throttling clients below their maxIOPS because of re-replication of data, transient errors, and snapshots taken.	N/A	float
timestamp	The current time in UTC+0 format.	N/A	ISO 8601 date string
unalignedReads	The total cumulative unaligned read operations to a volume since the creation of the volume.	Monotonically increasing	integer
unalignedWrites	The total cumulative unaligned write operations to a volume since the creation of the volume.	Monotonically increasing	integer
volumeAccessGroups	The list of IDs of volume access group(s) to which a volume belongs.	N/A	integer array
volumeID	The ID of the volume.	N/A	integer
volumeSize	Total provisioned capacity in bytes.	N/A	integer

Name	Description	Calculation	Type
volumeUtilization	A floating point value that describes how fully the client is using the volume's input / output capabilities in comparison with the maxIOPS QoS setting for that volume. Possible values: • 0: The client is not using the volume. • 0.01 to 0.99: The client is not fully utilizing the volume's IOPS capabilities. • 1.00: The client is fully utilizing the volume up to the IOPS limit set by the maxIOPS setting. • > 1.00: The client is utilizing more than the limit set by maxIOPS. This is possible when the burstIOPS QoS setting is set higher than maxIOPS. For example, if maxIOPS is set to 1000 and burstIOPS is set to 2000, the `volumeUtilization` value would be 2.00 if the client fully utilizes the volume.	N/A	float
writeBytes	The total cumulative bytes written to the volume since the creation of the volume.	Monotonically increasing	integer
writeBytesLastSample	The total number of bytes written to the volume during the last sample period.	Monotonically increasing	integer

Name	Description	Calculation	Type
writeLatencyUSec	The average time, in microseconds, to complete write operations to a volume in the last 500 milliseconds.	Point in time	integer
writeLatencyUSecTotal	The total time spent performing write operations to the volume.	Monotonically increasing	integer
writeOps	The total cumulative write operations to the volume since the creation of the volume.	Monotonically increasing	integer
writeOpsLastSample	The total number of write operations during the last sample period.	Point in time	integer
zeroBlocks	The total number of empty 4KiB blocks without data after the last round of garbage collection operation has completed.	Point in time	integer

Common methods

Common methods are methods used to retrieve information about the storage cluster, the API itself, or ongoing API operations.

- GetAPI
- GetAsyncResult
- GetCompleteStats
- GetLimits
- GetOrigin
- GetRawStats
- ListAsyncResults

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

GetAPI

You can use the `GetAPI` method to get a list of all the API methods and supported API endpoints that can be used in the system.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
<version>	A list of all supported API methods for this software version, where <version> is the current software version this system is running.	string array
currentVersion	The current version of the storage cluster software.	string
supportedVersions	A list of all API endpoints supported by the system.	string array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetAPI",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
    "result": {
        "12.0": [
            "AbortSnapMirrorRelationship",
            "AddAccount",
            "AddClusterAdmin",
```

```
            "AddDrives",
            "AddIdpClusterAdmin",
            "AddInitiatorsToVolumeAccessGroup",
            "AddKeyServerToProviderKmip",
            "AddLdapClusterAdmin",
            "AddNodes",
            "AddVirtualNetwork",
            "AddVolumesToVolumeAccessGroup",
            "BreakSnapMirrorRelationship",
            "BreakSnapMirrorVolume",
            "CancelClone",
            "CancelGroupClone",
            "CheckPingOnVlan",
            "CheckProposedCluster",
            "CheckProposedNodeAdditions",
            "ClearClusterFaults",
            "CloneMultipleVolumes",
            "CloneVolume",
            "CompleteClusterPairing",
            "CompleteVolumePairing",
            "CopyVolume",
            "CreateBackupTarget",
            "CreateClusterInterfacePreference",
            "CreateClusterSupportBundle",
            "CreateGroupSnapshot",
            "CreateIdpConfiguration",
            "CreateInitiators",
            "CreateKeyProviderKmip",
            "CreateKeyServerKmip",
            "CreatePublicPrivateKeyPair",
            "CreateQoSPolicy",
            "CreateSchedule",
            "CreateSnapMirrorEndpoint",
            "CreateSnapMirrorEndpointUnmanaged",
            "CreateSnapMirrorRelationship",
            "CreateSnapMirrorVolume",
            "CreateSnapshot",
            "CreateStorageContainer",
            "CreateSupportBundle",
            "CreateVolume",
            "CreateVolumeAccessGroup",
            "DeleteAllSupportBundles",
            "DeleteAuthSession",
            "DeleteAuthSessionsByClusterAdmin",
            "DeleteAuthSessionsByUsername",
            "DeleteClusterInterfacePreference",
```

```
            "DeleteGroupSnapshot",
            "DeleteIdpConfiguration",
            "DeleteInitiators",
            "DeleteKeyProviderKmip",
            "DeleteKeyServerKmip",
            "DeleteQoSPolicy",
            "DeleteSnapMirrorEndpoints",
            "DeleteSnapMirrorRelationships",
            "DeleteSnapshot",
            "DeleteStorageContainers",
            "DeleteVolume",
            "DeleteVolumeAccessGroup",
            "DeleteVolumes",
            "DisableAutoip",
            "DisableBmcColdReset",
            "DisableClusterSsh",
            "DisableEncryptionAtRest",
            "DisableIdpAuthentication",
            "DisableLdapAuthentication",
            "DisableSnmp",
            "EnableAutoip",
            "EnableBmcColdReset",
            "EnableClusterSsh",
            "EnableEncryptionAtRest",
            "EnableFeature",
            "EnableIdpAuthentication",
            "EnableLdapAuthentication",
            "EnableSnmp",
            "GetAccountByID",
            "GetAccountByName",
            "GetAccountEfficiency",
            "GetActiveTlsCiphers",
            "GetAsyncResult",
            "GetBackupTarget",
            "GetBinAssignmentProperties",
            "GetClientCertificateSignRequest",
            "GetClusterCapacity",
            "GetClusterConfig",
            "GetClusterFullThreshold",
            "GetClusterHardwareInfo",
            "GetClusterInfo",
            "GetClusterInterfacePreference",
            "GetClusterMasterNodeID",
            "GetClusterSshInfo",
            "GetClusterState",
            "GetClusterStats",
```

```
            "GetClusterStructure",
            "GetClusterVersionInfo",
            "GetCompleteStats",
            "GetConfig",
            "GetCurrentClusterAdmin",
            "GetDefaultQoS",
            "GetDriveHardwareInfo",
            "GetDriveStats",
            "GetFeatureStatus",
            "GetFipsReport",
            "GetHardwareConfig",
            "GetHardwareInfo",
            "GetIdpAuthenticationState",
            "GetIpmiConfig",
            "GetIpmiInfo",
            "GetKeyProviderKmip",
            "GetKeyServerKmip",
            "GetLdapConfiguration",
            "GetLimits",
            "GetLldpInfo",
            "GetLoginBanner",
            "GetLoginSessionInfo",
            "GetNetworkConfig",
            "GetNetworkInterface",
            "GetNodeFipsDrivesReport",
            "GetNodeHardwareInfo",
            "GetNodeStats",
            "GetNtpInfo",
            "GetNvramInfo",
            "GetOntapVersionInfo",
            "GetOrigin",
            "GetPendingOperation",
            "GetProtectionDomainLayout",
            "GetQoSPolicy",
            "GetRawStats",
            "GetRemoteLoggingHosts",
            "GetSSLCertificate",
            "GetSchedule",
            "GetSnapMirrorClusterIdentity",
            "GetSnmpACL",
            "GetSnmpInfo",
            "GetSnmpState",
            "GetSnmpTrapInfo",
            "GetStorageContainerEfficiency",
            "GetSupportedTlsCiphers",
            "GetSystemStatus",
```

```
            "GetVirtualVolumeCount",
            "GetVolumeAccessGroupEfficiency",
            "GetVolumeAccessGroupLunAssignments",
            "GetVolumeCount",
            "GetVolumeEfficiency",
            "GetVolumeStats",
            "InitializeSnapMirrorRelationship",
            "ListAccounts",
            "ListActiveAuthSessions",
            "ListActiveNodes",
            "ListActivePairedVolumes",
            "ListActiveVolumes",
            "ListAllNodes",
            "ListAsyncResults",
            "ListAuthSessionsByClusterAdmin",
            "ListAuthSessionsByUsername",
            "ListBackupTargets",
            "ListBulkVolumeJobs",
            "ListClusterAdmins",
            "ListClusterFaults",
            "ListClusterInterfacePreferences",
            "ListClusterPairs",
            "ListDeletedVolumes",
            "ListDriveHardware",
            "ListDriveStats",
            "ListDrives",
            "ListEvents",
            "ListFibreChannelPortInfo",
            "ListFibreChannelSessions",
            "ListGroupSnapshots",
            "ListISCSISessions",
            "ListIdpConfigurations",
            "ListInitiators",
            "ListKeyProvidersKmip",
            "ListKeyServersKmip",
            "ListNetworkInterfaces",
            "ListNodeFibreChannelPortInfo",
            "ListNodeStats",
            "ListPendingActiveNodes",
            "ListPendingNodes",
            "ListProtectionDomainLevels",
            "ListProtocolEndpoints",
            "ListQoSPolicies",
            "ListSchedules",
            "ListServices",
            "ListSnapMirrorAggregates",
```

```
            "ListSnapMirrorEndpoints",
            "ListSnapMirrorLuns",
            "ListSnapMirrorNetworkInterfaces",
            "ListSnapMirrorNodes",
            "ListSnapMirrorPolicies",
            "ListSnapMirrorRelationships",
            "ListSnapMirrorSchedules",
            "ListSnapMirrorVolumes",
            "ListSnapMirrorVservers",
            "ListSnapshots",
            "ListStorageContainers",
            "ListSyncJobs",
            "ListTests",
            "ListUtilities",
            "ListVirtualNetworks",
            "ListVirtualVolumeBindings",
            "ListVirtualVolumeHosts",
            "ListVirtualVolumeTasks",
            "ListVirtualVolumes",
            "ListVolumeAccessGroups",
            "ListVolumeStats",
            "ListVolumeStatsByAccount",
            "ListVolumeStatsByVirtualVolume",
            "ListVolumeStatsByVolume",
            "ListVolumeStatsByVolumeAccessGroup",
            "ListVolumes",
            "ListVolumesForAccount",
            "ModifyAccount",
            "ModifyBackupTarget",
            "ModifyClusterAdmin",
            "ModifyClusterFullThreshold",
            "ModifyClusterInterfacePreference",
            "ModifyGroupSnapshot",
            "ModifyInitiators",
            "ModifyKeyServerKmip",
            "ModifyQoSPolicy",
            "ModifySchedule",
            "ModifySnapMirrorEndpoint",
            "ModifySnapMirrorEndpointUnmanaged",
            "ModifySnapMirrorRelationship",
            "ModifySnapshot",
            "ModifyStorageContainer",
            "ModifyVirtualNetwork",
            "ModifyVolume",
            "ModifyVolumeAccessGroup",
            "ModifyVolumeAccessGroupLunAssignments",
```

```
            "ModifyVolumePair",
            "ModifyVolumes",
            "PurgeDeletedVolume",
            "PurgeDeletedVolumes",
            "QuiesceSnapMirrorRelationship",
            "RemoveAccount",
            "RemoveBackupTarget",
            "RemoveClusterAdmin",
            "RemoveClusterPair",
            "RemoveDrives",
            "RemoveInitiatorsFromVolumeAccessGroup",
            "RemoveKeyServerFromProviderKmip",
            "RemoveNodes",
            "RemoveSSLCertificate",
            "RemoveVirtualNetwork",
            "RemoveVolumePair",
            "RemoveVolumesFromVolumeAccessGroup",
            "ResetDrives",
            "ResetNetworkConfig",
            "ResetNode",
            "ResetSupplementalTlsCiphers",
            "RestartNetworking",
            "RestartServices",
            "RestoreDeletedVolume",
            "ResumeSnapMirrorRelationship",
            "ResyncSnapMirrorRelationship",
            "RollbackToGroupSnapshot",
            "RollbackToSnapshot",
            "SecureEraseDrives",
            "SetClusterConfig",
            "SetClusterStructure",
            "SetConfig",
            "SetDefaultQoS",
            "SetLoginBanner",
            "SetLoginSessionInfo",
            "SetNetworkConfig",
            "SetNtpInfo",
            "SetProtectionDomainLayout",
            "SetRemoteLoggingHosts",
            "SetSSLCertificate",
            "SetSnmpACL",
            "SetSnmpInfo",
            "SetSnmpTrapInfo",
            "SetSupplementalTlsCiphers",
            "Shutdown",
            "SnmpSendTestTraps",
```

```
            "StartBulkVolumeRead",
            "StartBulkVolumeWrite",
            "StartClusterPairing",
            "StartVolumePairing",
            "TestAddressAvailability",
            "TestConnectEnsemble",
            "TestConnectMvip",
            "TestConnectSvip",
            "TestDrives",
            "TestHardwareConfig",
            "TestKeyProviderKmip",
            "TestKeyServerKmip",
            "TestLdapAuthentication",
            "TestLocalConnectivity",
            "TestLocateCluster",
            "TestNetworkConfig",
            "TestPing",
            "TestRemoteConnectivity",
            "UpdateBulkVolumeStatus",
            "UpdateIdpConfiguration",
            "UpdateSnapMirrorRelationship"
        ],
        "currentVersion": "12.0",
        "supportedVersions": [
            "1.0",
            "2.0",
            "3.0",
            "4.0",
            "5.0",
            "5.1",
            "6.0",
            "7.0",
            "7.1",
            "7.2",
            "7.3",
            "7.4",
            "8.0",
            "8.1",
            "8.2",
            "8.3",
            "8.4",
            "8.5",
            "8.6",
            "8.7",
            "9.0",
            "9.1",
```

```
            "9.2",
            "9.3",
            "9.4",
            "9.5",
            "9.6",
            "10.0",
            "10.1",
            "10.2",
            "10.3",
            "10.4",
            "10.5",
            "10.6",
            "10.7",
            "11.0",
            "11.1",
            "11.3",
            "11.5",
            "11.7",
            "11.8",
            "12.0"
        ]
    }
}
```

GetAsyncResult

You can use `GetAsyncResult` to retrieve the result of asynchronous method calls. Some method calls require some time to run, and might not be finished when the system sends the initial response. To obtain the status or result of the method call, use `GetAsyncResult` to poll the asyncHandle value returned by the method.

`GetAsyncResult` returns the overall status of the operation (in progress, completed, or error) in a standard fashion, but the actual data returned for the operation depends on the original method call and the return data is documented with each method.

If the keepResult parameter is missing or false, the asyncHandle becomes inactive when the result is returned, and later attempts to query that asyncHandle return an error. You can keep the asyncHandle active for future queries by setting the keepResult parameter to true.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
asyncHandle	A value that was returned from the original asynchronous method call.	integer	None	Yes
keepResult	If true, GetAsyncResult does not remove the asynchronous result upon returning it, enabling future queries to that asyncHandle.	boolean	false	No

Return values

This method has the following return values:

Name	Description	Type
status	Status of the asynchronous method call. Possible values: • running: The method is still running. • complete: The method is complete and the result or error is available.	string
result	If the asynchronous method successfully completed, this is the result of the asynchronous operation. If the asynchronous operation failed, this member is not present.	string
error	If the status is complete and the asynchronous method failed, this member includes the error details. If the asynchronous operation succeeded, this member is not present.	string
resultType	The type of operation the asynchronous method call is or was performing.	string

Name	Description	Type
details	If the status is running, this member includes information relevant to the method's current operation. If the asynchronous method is not running, this member is not present.	JSON Object
createTime	The time that the asynchronous method was called, in UTC+0 format.	ISO 8601 date string
lastUpdateTime	The time that the asynchronous method's status was last updated, in UTC+0 format.	ISO 8601 date string

Note: The return value of GetAsyncResult is essentially a nested version of the standard JSON response with an additional status field.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetAsyncResult",
  "params": {
      "asyncHandle" : 389
},
"id" : 1
}
```

Response example: method error

This method returns a response similar to the following example:

```
{
   "error": {
      "code": 500,
      "message": "DBClient operation requested on a non-existent path at
[/asyncresults/1]",
      "name": "xDBNoSuchPath"
   },
   "id": 1
}
```

If "response" were the JSON response object from the GetAsyncResult call, then "response.error" would

correspond to an error with the GetAsyncResult method itself (such as querying a non-existent asyncHandle).

Response example: asynchronous task error

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "createTime": "2016-01-01T02:05:53Z",
     "error": {
       "bvID": 1,
       "message": "Bulk volume job failed",
       "name": "xBulkVolumeScriptFailure",
       "volumeID": 34
     },
     "lastUpdateTime": "2016-01-21T02:06:56Z",
     "resultType": "BulkVolume",
     "status": "complete"
   }
}
```

The “response.result.error” would correspond to an error result from the original method call.

Response example: asynchronous task success

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "createTime": "2016-01-01T22:29:18Z",
     "lastUpdateTime": "2016-01-01T22:45:51Z",
     "result": {
       "cloneID": 25,
       "message": "Clone complete.",
       "volumeID": 47
     },
     "resultType": "Clone",
     "status": "complete"
   }
}
```

The “response.result.result” is the return value for the original method call if the call completed successfully.

New since version

9.6

GetCompleteStats

NetApp engineering uses the `GetCompleteStats` API method to test new features. The data returned from `GetCompleteStats` is not documented, changes frequently, and is not guaranteed to be accurate. You should not use `GetCompleteStats` for collecting performance data or any other management integration with a storage cluster running Element software.

Use the following supported API methods to retrieve statistical information:

- GetVolumeStats
- GetClusterStats
- GetNodeStats
- GetDriveStats

New since version

9.6

GetLimits

You can use the `GetLimits` method to get the limit values set by the API. These values might change between releases of Element, but do not change without an update to the system. Knowing the limit values set by the API can be useful when writing API scripts for user-facing tools.

The `GetLimits` method returns the limits for the current software version regardless of the API endpoint version used to pass the method.

Parameters

This method has no input parameters.

Return values

This method returns a JSON object with name-value pairs containing the API limits.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetLimits",
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "accountCountMax": 5000,
        "accountNameLengthMax": 64,
        "accountNameLengthMin": 1,
        "backupTargetNameLengthMax": 64,
        "backupTargetNameLengthMin": 1,
        "bulkVolumeJobsPerNodeMax": 8,
        "bulkVolumeJobsPerVolumeMax": 2,
        "chapCredentialsCountMax": 15000,
        "cloneJobsPerNodeMax": 8,
        "cloneJobsPerVirtualVolumeMax": 8,
        "cloneJobsPerVolumeMax": 2,
        "clusterAdminAccountMax": 5000,
        "clusterAdminInfoNameLengthMax": 1024,
        "clusterAdminInfoNameLengthMin": 1,
        "clusterPairsCountMax": 4,
        "fibreChannelVolumeAccessMax": 16384,
        "initiatorAliasLengthMax": 224,
        "initiatorCountMax": 10000,
        "initiatorNameLengthMax": 224,
        "initiatorsPerVolumeAccessGroupCountMax": 128,
        "iscsiSessionsFromFibreChannelNodesMax": 4096,
        "maxAuthSessionsForCluster": 1024,
        "maxAuthSessionsPerUser": 1024,
        "nodesPerClusterCountMax": 100,
        "nodesPerClusterCountMin": 3,
        "qosPolicyCountMax": 500,
        "qosPolicyNameLengthMax": 64,
        "qosPolicyNameLengthMin": 1,
        "scheduleNameLengthMax": 244,
        "secretLengthMax": 16,
        "secretLengthMin": 12,
        "snapMirrorEndpointIPAddressesCountMax": 64,
        "snapMirrorEndpointsCountMax": 4,
```

```
        "snapMirrorLabelLengthMax": 31,
        "snapMirrorObjectAttributeValueInfoCountMax": 9900000,
        "snapshotNameLengthMax": 255,
        "snapshotsPerVolumeMax": 32,
        "storageNodesPerClusterCountMin": 2,
        "virtualVolumeCountMax": 8000,
        "virtualVolumesPerAccountCountMax": 10000,
        "volumeAccessGroupCountMax": 1000,
        "volumeAccessGroupLunMax": 16383,
        "volumeAccessGroupNameLengthMax": 64,
        "volumeAccessGroupNameLengthMin": 1,
        "volumeAccessGroupsPerInitiatorCountMax": 1,
        "volumeAccessGroupsPerVolumeCountMax": 64,
        "volumeBurstIOPSMax": 200000,
        "volumeBurstIOPSMin": 100,
        "volumeCountMax": 4000,
        "volumeMaxIOPSMax": 200000,
        "volumeMaxIOPSMin": 100,
        "volumeMinIOPSMax": 15000,
        "volumeMinIOPSMin": 50,
        "volumeNameLengthMax": 64,
        "volumeNameLengthMin": 1,
        "volumeSizeMax": 17592186044416,
        "volumeSizeMin": 1000000000,
        "volumesPerAccountCountMax": 2000,
        "volumesPerGroupSnapshotMax": 32,
        "volumesPerVolumeAccessGroupCountMax": 2000,
        "witnessNodesPerClusterCountMax": 4
    }
}
```


New since version

9.6

GetOrigin

You can use the `GetOrigin` method to get the origination certificate for where the node was built.

Parameters

This method returns "null" if there is no origination certification.

This method has no input parameters.

Return value

This method returns vendor origination certification information.

Request example

Requests for this method are similar to the following example:

```
{
 "method": "GetOrigin",
 "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "integrator": "SolidFire",
  "<signature>": {
    "pubkey": [public key info],
    "version": 1,
    "data": [signature info]
  },
   "contract-id": "none",
   "location": "Boulder, CO",
   "organization": "Engineering",
   "type": "element-x"
  }
 ]
}
```

New since version

9.6

GetRawStats

NetApp engineering uses the `GetRawStats` API method to test new features. The data returned from `GetRawStats` is not documented, changes frequently, and is not guaranteed to be accurate. You should not use `GetRawStats` for collecting performance data or any other management integration with a storage cluster running Element software.

Use the following supported API methods to retrieve statistical information:

- GetVolumeStats
- GetClusterStats
- GetNodeStats
- GetDriveStats

New since version

9.6

ListAsyncResults

You can use `ListAsyncResults` to list the results of all currently running and completed asynchronous methods on the system. Querying asynchronous results with `ListAsyncResults` does not cause completed asyncHandles to expire; you can use `GetAsyncResult` to query any of the asyncHandles returned by `ListAsyncResults`.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
asyncResultTypes	An optional list of types of results. You can use this list to restrict the results to only these types of operations. Possible values: • DriveAdd: Operations involving the system adding a drive to the cluster. • BulkVolume: Copy operations between volumes, such as backups or restores. • Clone: Volume cloning operations. • DriveRemoval: Operations involving the system copying data from a drive in preparation to remove it from the cluster. • RtfiPendingNode: Operations involving the system installing compatible software on a node before adding it to the cluster.	string array	None	No

Return value

This method has the following return value:

Name	Description	Type

asyncHandles	An array of serialized asynchronous method results.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListAsyncResults",
   "params": {
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "asyncHandles": [
       {
         "asyncResultID": 47,
         "completed": true,
         "createTime": "2016-01-01T22:29:19Z",
         "data": {
           "cloneID": 26,
           "message": "Clone complete.",
           "volumeID": 48
         },
         "lastUpdateTime": "2016-01-01T22:45:43Z",
         "resultType": "Clone",
         "success": true
       },
       ...]
   }
}
```

New since version

9.6

Find more information

GetAsyncResult

Account API methods

Account methods enable you to add, remove, view, and modify account and security information.

- AddAccount
- GetAccountByID
- GetAccountByName
- GetAccountEfficiency
- ListAccounts
- ModifyAccount
- RemoveAccount

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddAccount

You can use `AddAccount` to add a new account to the system. You can also use this method to create new volumes under the new account as the account is created. The Challenge-Handshake Authentication Protocol (CHAP) settings you specify for the account apply to all volumes owned by the account.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`attributes`	List of name-value pairs in JSON object format.	JSON object	None	No
`enableChap`	Specifies whether CHAP account credentials can be used by an initiator to access volumes.	boolean	true	No

Name	Description	Type	Default value	Required
`initiatorSecret`	The CHAP secret to use for the initiator. This secret must be 12 to 16 characters in length and should be impenetrable. The initiator CHAP secret must be unique and cannot be the same as the target CHAP secret. If not specified, a random secret is created.	string	None	No
`targetSecret`	The CHAP secret to use for the target (mutual CHAP authentication). This secret must be 12 to 16 characters in length and should be impenetrable. The target CHAP secret must be unique and cannot be the same as the initiator CHAP secret. If not specified, a random secret is created.	string	None	No
`username`	The unique username for this account. (Must be 1 to 64 characters in length).	string	None	Yes

Return value

This method has the following return values:

Name	Description	Type
account	An object containing information about the newly created account.	account
accountID	The ID of the newly created account object.	integer

Request example

Requests for this method are similar to the following example:

```
{
    "method": "AddAccount",
    "params": {
     "username"  : "bobsmith",
     "initiatorSecret" : "168[#5A757ru268)",
     "targetSecret" : "tlt&lt;,8TUYa7bC",
     "attributes" : {
       "billingcode" : 2345
     }
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "account": {
      "accountID": 90,
      "attributes": {
        "billingcode": 2345
      },
      "initiatorSecret": "168[#5A757ru268)",
      "status": "active",
      "storageContainerID": "00000000-0000-0000-0000-000000000000",
      "targetSecret": "tlt&lt;,8TUYa7bC",
      "username": "bobsmith",
      "volumes": [],
      "enableChap": true
    },
    "accountID": 90
  }
}
```

New since version

9.6

GetAccountByID

You can use `GetAccountByID` to get details about a specific account, given its accountID.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
accountID	The account ID of the account for which to get information.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
account	Account details.	account

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetAccountByID",
   "params": {
     "accountID"  : 3
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "account": {
        "attributes": {},
        "username": "account3",
        "targetSecret": "targetsecret",
        "volumes": [],
        "enableChap": true,
        "status": "active",
        "accountID": 3,
        "storageContainerID": "abcdef01-1234-5678-90ab-cdef01234567",
        "initiatorSecret": "initiatorsecret"
    }
}
```


New since version

9.6

GetAccountByName

You can use `GetAccountByName` to get details about a specific account, given its username.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
username	User name for the account.	string	None	Yes

Return value

This method has the following return value:

Name	Description	Type
account	Account details.	account

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetAccountByName",
   "params": {
     "username"  : "jimmyd"
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "account": {
        "attributes": {},
        "username": "jimmyd",
        "targetSecret": "targetsecret",
        "volumes": [],
        "enableChap": true,
        "status": "active",
        "accountID": 1,
        "storageContainerID": "abcdef01-1234-5678-90ab-cdef01234567",
        "initiatorSecret": "initiatorsecret"
    }
}
```

New since version

9.6

GetAccountEfficiency

You can use `GetAccountEfficiency` to get efficiency statistics about a volume account. This method returns efficiency information only for the account you give as a parameter.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
accountID	Specifies the volume account for which efficiency statistics are returned.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
compression	The amount of space saved by data compression for all volumes in the account. Stated as a ratio where a value of "1" means data has been stored with no compression.	float
deduplication	The amount of space saved by not duplicating data for all volumes in the account. Stated as a ratio.	float
missingVolumes	The volumes that could not be queried for efficiency data. Missing volumes can be caused by the Garbage Collection (GC) cycle being less than an hour old, temporary loss of network connectivity, or restarted services since the GC cycle.	integer array
thinProvisioning	The ratio of space used to the amount of space allocated for storing data. Stated as a ratio.	float
timestamp	The last time efficiency data was collected after Garbage Collection (GC), in UTC+0 format.	ISO 8601 date string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetAccountEfficiency",
   "params": {
     "accountID": 3
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
       "compression": 2.020468042933262,
       "deduplication": 2.042488619119879,
       "missingVolumes": [],
       "thinProvisioning": 1.010087163391013,
       "timestamp": "2014-03-10T14:06:02Z"
   }
}
```

New since version

9.6

ListAccounts

You can use `ListAccounts` to get the entire list of storage tenant accounts, with optional paging support. Element accounts enable access to volumes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeStorageContainers	Virtual volume storage containers are included in the response by default. To exclude storage containers, set to false.	boolean	true	No

Name	Description	Type	Default value	Required
startAccountID	Starting accountID to return. If no account exists with this accountID, the next account by accountID order is used as the start of the list. To page through the list, pass the accountID of the last account in the previous response + 1.	integer	None	No
limit	Maximum number of account objects to return.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
accounts	The list of accounts.	account array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListAccounts",
   "params": {
     "startAccountID"  : 0,
     "limit" : 1000
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "result" : {
    "accounts": [
    {
        "attributes": {},
        "username": "jamesw",
        "targetSecret": "168#5A757ru268)",
        "volumes": [],
        "enableChap": false,
        "status": "active",
        "accountID": 16,
        "storageContainerID": "abcdef01-1234-5678-90ab-cdef01234567",
        "initiatorSecret": "168#5A757ru268)"
    },
    {
        "attributes": {},
        "username": "jimmyd",
        "targetSecret": "targetsecret",
        "volumes": [],
        "enableChap": true,
        "status": "active",
        "accountID": 5,
        "storageContainerID": "abcdef01-1234-5678-90ab-cdef01234567",
        "initiatorSecret": "initiatorsecret"
    }
  ]
}
}
```


New since version

9.6

ModifyAccount

You can use the `ModifyAccount` method to modify an existing account.

When you lock an account, any existing connections from that account are immediately terminated. When you change an account's CHAP settings, any existing connections remain active, and the new CHAP settings are used on subsequent connections or reconnections. To clear an account's attributes, specify {} for the attributes parameter.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
accountID	AccountID for the account to modify.	integer	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
enableChap	Specifies whether CHAP account credentials can be used by an initiator to access volumes.	boolean	None	No
initiatorSecret	The CHAP secret to use for the initiator. This secret must be 12-16 characters in length and should be impenetrable. The initiator CHAP secret must be unique and cannot be the same as the target CHAP secret.	string	None	No
status	Status for the account. Possible values: • active: Account is active and connections are allowed. • locked: Account is locked and connections are refused.	string	None	No

Name	Description	Type	Default value	Required
targetSecret	The CHAP secret to use for the target (mutual CHAP authentication). This secret must be 12-16 characters in length and should be impenetrable. The target CHAP secret must be unique and cannot be the same as the initiator CHAP secret.	string	None	No
username	Used to change the username associated with the account. (Must be 1 to 64 characters in length).	string	None	No

Return value

This method has the following return value:

Name	Description	Type
account	An object containing information about the modified account.	account

Request example

Requests for this method are similar to the following example. In this example, the attributes are cleared by specifying {} for them:

```
{
   "method": "ModifyAccount",
   "params": {
     "accountID"  : 25,
     "status"    : "locked",
     "attributes" : {}
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "account": {
        "storageContainerID": "abcdef01-1234-5678-90ab-cdef01234567",
        "username": "user1",
        "accountID": 1,
        "volumes": [
        ],
        "enableChap": true,
        "initiatorSecret": "txz123456q890",
        "attributes": {
        },
        "status": active",
        "targetSecret": "rxe123b567890"
    }
}
```

New since version

9.6

RemoveAccount

You can use the `RemoveAccount` method to remove an existing account. You must delete and purge all volumes associated with the account using `DeleteVolume` before you can remove the account. If volumes on the account are still pending deletion, you cannot use `RemoveAccount` to remove the account.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
accountID	The ID of the account to remove.	integer	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example.

```
{
   "method": "RemoveAccount",
   "params": {
     "accountID"  : 25
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id" : 1,
 "result" : { }
}
```

New since version

9.6

Find more information

DeleteVolume

Administrator API methods

You can use administrator API methods to create, modify, view, and remove storage cluster administrators and assign levels of access and privileges for those with access to a storage cluster.

- AddClusterAdmin
- GetCurrentClusterAdmin
- GetLoginBanner
- ListClusterAdmins
- ModifyClusterAdmin
- RemoveClusterAdmin
- SetLoginBanner

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddClusterAdmin

You can use the `AddClusterAdmin` method to add a new cluster admin account. A cluster admin can manage the cluster via the API and management tools. Cluster admins are completely separate and unrelated to standard tenant accounts.

Each cluster admin can be restricted to a subset of the API. You should use multiple cluster admin accounts for different users and applications. As a best practice, give each cluster admin the minimal permissions necessary; this reduces the potential impact of credential compromise.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	Controls which methods the cluster admin can use.	string array	None	Yes
acceptEula	Accept the End User License Agreement. Set to true to add a cluster administrator account to the system. If omitted or set to false, the method call fails.	boolean	None	Yes
attributes	List of name/value pairs in JSON object format.	JSON object	None	No
password	Password used to authenticate this cluster admin.	string	None	Yes
username	Unique username for this cluster admin. Must be between 1 and 1024 characters in length.	string	None	Yes

Return value

This method has the following return value:

Name	Description	Type

clusterAdminID	ClusterAdminID for the newly created cluster admin.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "AddClusterAdmin",
  "params": {
    "username": "joeadmin",
    "password": "68!5Aru268)$",
    "attributes": {},
    "acceptEula": true,
    "access": ["volumes", "reporting", "read"]
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id":1,
  "result" : {
    "clusterAdminID": 2
   }
}
```

New since version

9.6

Find more information

Access control

GetCurrentClusterAdmin

You can use the `GetCurrentClusterAdmin` method to return information for the current primary Cluster Admin. The primary Cluster Admin was created when the cluster was created.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
clusterAdmin	Information about the cluster admin.	clusterAdmin

Request example

Requests for this method are similar to the following example:

```
{
"method": "GetCurrentClusterAdmin",
"id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
"result": {
    "clusterAdmin":
        "access": [
           "administrator"
        ],
         "attributes": null,
         "authMethod": "Cluster"
         "clusterAdminID": 1,
         "username": "admin"
     }
   }
}
```

New since version

10.0

GetLoginBanner

You can use the `GetLoginBanner` method to get the currently active Terms of Use banner that users see when they log in to the Element web interface.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
banner	The current text of the Terms of Use banner. This value can contain text even when the banner is disabled.	string
enabled	The status of the Terms of Use banner. Possible values: • true: The Terms of Use banner is displayed upon web interface login. • false: The Terms of Use banner is not displayed upon web interface login.	boolean

Request example

Requests for this method are similar to the following example:

```
{
  "id": 3411,
  "method": "GetLoginBanner",
  "params": {}
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 3411,
  "result": {
    "loginBanner": {
      "banner": "Welcome to NetApp!",
      "enabled": false
    }
  }
}
```

New since version

10.0

ListClusterAdmins

You can use the `ListClusterAdmins` method to return the list of all cluster administrators for the cluster.

There can be several cluster administrator accounts with different levels of permissions. There can be only one primary cluster administrator in the system. The primary Cluster Admin is the administrator that was created when the cluster was created. LDAP administrators can also be created when setting up an LDAP system on the cluster.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
showHidden	Shows hidden cluster administrator users, such as SNMP admin.	boolean	None	No

Return value

This method has the following return value:

Name	Description	Type
clusterAdmins	Information about all cluster and LDAP administrators that exist for a cluster.	clusterAdmin array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListClusterAdmins",
   "params": {},
    "showHidden": true
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id":1,
"result":{
  "clusterAdmins":[
     {
       "access":[
           "administrator"
       ],
       "attributes":null,
       "authMethod":"Cluster",
       "clusterAdminID":1,
       "username":"admin"
   },
   {
       "access":[
           "read",
           "administrator"
       ],
       "attributes":{
       },
       "authMethod":"Ldap",
       "clusterAdminID":7,
       "username":"john.smith"
   },
   {
       "access":[
           "read",
           "administrator"
       ],
       "attributes":{},
       "authMethod":"Ldap",
       "clusterAdminID":6,
       "username":"cn=admin1
jones,ou=ptusers,c=prodtest,dc=solidfire,dc=net"
       }
     ]
   }
}
```


New since version

9.6

ModifyClusterAdmin

You can use the `ModifyClusterAdmin` method to change the settings for a cluster admin, LDAP cluster admin, or third-party Identity Provider (IdP) cluster admin. You cannot change access for the administrator cluster admin account.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	Controls which methods this cluster admin can use.	string array	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
clusterAdminID	ClusterAdminID for the cluster admin, LDAP cluster admin, or IdP cluster admin to modify.	integer	None	Yes
password	Password used to authenticate this cluster admin. This parameter does not apply to an LDAP or IdP cluster admin.	string	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ModifyClusterAdmin",
   "params": {
     "clusterAdminID"  : 2,
     "password"   : "7925Brc429a"
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id" : 1
 "result" : { }
}
```

New since version

9.6

Find more information

Access control

RemoveClusterAdmin

You can use the `RemoveClusterAdmin` method to remove a cluster admin, an LDAP cluster admin, or a third-party Identity Provider (IdP) cluster admin. You cannot remove the "admin" Cluster Admin account.

Parameter

When an IdP cluster admin is removed that has authenticated sessions associated with a third-party Identity Provider IdP, those sessions will either logout or possibly experience a loss of access rights within their current session. The access rights loss will depend on whether the removed IdP cluster admin matched one of multiple IdP cluster admins from a given user's SAML attributes. The remaining set of matching IdP cluster admins results in a reduced set of aggregate access rights. Other cluster admin user types are logged out when their cluster admins are removed.

This method has the following input parameter:

Name	Description	Type	Default value	Required
clusterAdminID	ClusterAdminID for the Cluster Admin to remove.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveClusterAdmin",
   "params": {
     "clusterAdminID"  : 2
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id" : 1
 "result" : { }
}
```

New since version

9.6

SetLoginBanner

You can use the `SetLoginBanner` method to configure the Terms of Use banner that users see when they log in to the Element web interface.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
banner	The desired text of the Terms of Use banner. The maximum length allowed is 4,096 characters.	string	None	No
enabled	The status of the Terms of Use banner. Possible values: • `true`: The Terms of Use banner is displayed upon web interface login. • `false`: The Terms of Use banner is not displayed upon web interface login.	boolean	None	No

Return values

This method has the following return values:

Name	Description	Type
banner	The current text of the Terms of Use banner. This value can contain text even when the banner is disabled.	string
enabled	The status of the Terms of Use banner. Possible values: • true: The Terms of Use banner is displayed upon web interface login. • false: The Terms of Use banner is not displayed upon web interface login.	boolean

Request example

Requests for this method are similar to the following example:

```
{
  "id": 3920,
  "method": "SetLoginBanner",
  "params": {
    "banner": "Welcome to NetApp!",
    "enabled": true
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 3920,
  "result": {
    "loginBanner": {
      "banner": "Welcome to NetApp!",
      "enabled": true
    }
  }
}
```

New since version

10.0

Cluster API methods

Element software cluster API methods enable you to manage the configuration and topology of the storage cluster and the nodes that belong to a storage cluster.

Some cluster API methods operate on nodes that are part of a cluster, or have been configured to join a cluster. You can add nodes to a new cluster or to an existing cluster. Nodes that are ready to be added to a cluster are in a "pending" state, which means they have been configured but not yet added to the cluster.

- AddNodes
- ClearClusterFaults
- CreateClusterInterfacePreference
- DeleteClusterInterfacePreference
- EnableFeature

- GetClusterCapacity
- GetClusterFullThreshold
- GetClusterHardwareInfo
- GetClusterInfo
- GetClusterInterfacePreference
- GetClusterMasterNodeID
- GetClusterStats
- GetClusterVersionInfo
- GetFeatureStatus
- GetLoginSessionInfo
- GetNodeHardwareInfo
- GetNodeStats
- ListActiveNodes
- ListAllNodes
- ListClusterFaults
- ListClusterInterfacePreferences
- ListEvents
- ListNodeStats
- ListISCSISessions
- ListServices
- ListPendingNodes
- ListPendingActiveNodes
- ModifyClusterFullThreshold
- ModifyClusterInterfacePreference
- RemoveNodes
- SetLoginSessionInfo
- Shutdown

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddNodes

You can use the `AddNodes` method to add one or more new nodes to a cluster.

When a node that is not configured starts up for the first time, you are prompted to configure the node. Once you configure the node, it is registered as a "pending node" with the cluster. Storage clusters running Element software automatically image a node to the version on the cluster. When you add a pending node, the method response includes an asyncHandle value that you can use with the `GetAsyncResult` method to query the

status of the automatic imaging process.

The process of adding a Fibre Channel node is the same as adding Element iSCSI storage nodes to a cluster. Fibre Channel nodes are registered in the system with a NodeID. When they become accessible, they are put in a "pending node" status. The `ListAllNodes` method will return the pendingNodeID for iSCSI nodes as well as any Fibre Channel nodes that are available to add to the cluster.

When you add a node to a cluster that you have configured for virtual networking, the system requires a sufficient number of virtual storage IP addresses to allocate a virtual IP to the new node. If there are no virtual IP addresses available for the new node, the `AddNode` operation fails. Use the `ModifyVirtualNetwork` method to add more storage IP addresses to your virtual network.

Once you add a node, any drives on the node are made available and you can add them using the `AddDrives` method to increase the storage capacity of the cluster.

It may take several seconds after adding a new node for it to start up and register its drives as available.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
autoInstall	If true, a return to factory image (RTFI) will be performed on the node upon adding. The default behavior is to perform RTFI. If the `cEnableAutoInstall` cluster constant is false, it takes priority over this parameter. If an upgrade is in progress, the RTFI process will not happen regardless of the value for this parameter.	boolean	None	No
pendingNodes	Pending NodeIDs for the nodes to be added. You can list all pending nodes using the ListPendingNodes method.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
autoInstall	Whether or not the added nodes are being returned to the factory image.	boolean
nodes	An array of objects mapping the previous "pendingNodeID" to the "nodeID". When you add a pending node that is running an incompatible software version, this array includes an asyncHandle value that you can use with the GetAsyncResult method to query the status of the automatic imaging process.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "AddNodes",
   "params": {
     "autoInstall" : true,
     "pendingNodes" : [1]
   },
   "id":1
}
```

Response example

This method returns a response similar to the following example:

```
{
  id: null,
  result: {
    autoInstall: true,
    nodes: [
      {
        activeNodeKey: "giAm2ep1hA",
        assignedNodeID: 6,
        asyncHandle: 3,
        cip: "10.10.5.106",
        mip: "192.168.133.106",
        pendingNodeID: 2,
        platformInfo: {
          chassisType: "R620",
          cpuModel: "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
          nodeMemoryGB: 72,
          nodeType: "SF3010"
        },
        sip: "10.10.5.106",
        softwareVersion: "9.0.0.1077"
      }
    ]
  }
}
```


New since version

9.6

Find more information

- AddDrives
- GetAsyncResult
- ListAllNodes
- ModifyVirtualNetwork

ClearClusterFaults

You can use the `ClearClusterFaults` method to clear information about both current and previously detected faults. Both resolved and unresolved faults can be cleared.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
faultTypes	Determines the types of faults to clear. Possible values: • current: Faults that are detected currently and have not been resolved. • resolved: Faults that were previously detected and resolved. • all: Both current and resolved faults. The fault status can be determined by the “resolved” field of the fault object.	string	resolved	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ClearClusterFaults",
   "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
    "result" : {}
}
```

New since version

9.6

CreateClusterInterfacePreference

The `CreateClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to create and store arbitrary information on the storage cluster. This method is for internal use.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference.	string	None	Yes
value	The value of the cluster interface preference.	string	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
        "method": "CreateClusterInterfacePreference",
        "params": {
            "name": "prefname",
            "value": "testvalue"
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {}
}
```

New since version

11.0

DeleteClusterInterfacePreference

The `DeleteClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to delete an existing cluster interface preference. This method is for internal use.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference to delete.	string	None	Yes

Return values

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
    "method": "DeleteClusterInterfacePreference",
    "params": {
        "name": "prefname"
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {}
}
```

New since version

11.0

EnableFeature

You can use the `EnableFeature` method to enable cluster features such as VVols that are disabled by default.

Parameter

This method has the following input parameter.

> For systems running Element software 11.x, enabling virtual volumes before or after setting protection domain monitoring causes the cluster protection domains feature to function only at node level.

Name	Description	Type	Default value	Required
feature	Enable a cluster feature. Possible values: • `fips`: Enable FIPS 140-2 certified encryption for HTTPS communications. • `FipsDrives`: Enable FIPS 140-2 drive support for the storage cluster. • `SnapMirror`: Enable the SnapMirror replication cluster feature. • `vvols`: Enable the Element software VVols cluster feature.	string	None	Yes

Return value

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableFeature",
    "params": {
        "feature" : "vvols"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {}
}
```

New since version

9.6

GetClusterCapacity

You can use the `GetClusterCapacity` to return high-level capacity measurements for an entire storage cluster. This method returns fields that you can use to calculate the efficiency rates shown in the Element web UI. You can use the efficiency calculations in scripts to return the efficiency rates for thin provisioning, deduplication, compression, and overall efficiency.

Efficiency calculations

Use the following equations to calculate thin provisioning, deduplication, and compression. These equations apply to Element 8.2 and later.

- thinProvisioningFactor = (nonZeroBlocks + zeroBlocks) / nonZeroBlocks
- deDuplicationFactor = (nonZeroBlocks + snapshotNonZeroBlocks) / uniqueBlocks
- compressionFactor = (uniqueBlocks * 4096) / (uniqueBlocksUsedSpace * 0.93)

Overall efficiency rate calculation

Use the following equation to calculate overall cluster efficiency using the results of the thin provisioning, deduplication, and compression efficiency calculations.

- efficiencyFactor = thinProvisioningFactor * deDuplicationFactor * compressionFactor

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
clusterCapacity	Capacity measurements for the storage cluster.	clusterCapacity

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetClusterCapacity",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "clusterCapacity": {
      "activeBlockSpace": 236015557096,
      "activeSessions": 20,
      "averageIOPS": 0,
      "clusterRecentIOSize": 0,
      "currentIOPS": 0,
      "maxIOPS": 150000,
      "maxOverProvisionableSpace": 259189767127040,
      "maxProvisionedSpace": 51837953425408,
      "maxUsedMetadataSpace": 404984011161,
      "maxUsedSpace": 12002762096640,
      "nonZeroBlocks": 310080350,
      "peakActiveSessions": 20,
      "peakIOPS": 0,
      "provisionedSpace": 1357931085824,
      "snapshotNonZeroBlocks": 0,
      "timestamp": "2016-10-17T21:24:36Z",
      "totalOps": 1027407650,
      "uniqueBlocks": 108180156,
      "uniqueBlocksUsedSpace": 244572686901,
      "usedMetadataSpace": 8745762816,
      "usedMetadataSpaceInSnapshots": 8745762816,
      "usedSpace": 244572686901,
      "zeroBlocks": 352971938
    }
  }
}
```

New since version

9.6

GetClusterFullThreshold

You can use the `GetClusterFullThreshold` method to view the stages set for cluster fullness levels. This method returns all fullness metrics for the cluster.

> When a cluster reaches the Error stage of block cluster fullness, the maximum IOPS on all volumes are reduced linearly to the volume minimum IOPS as the cluster approaches the Critical stage. This helps prevent the cluster from reaching the Critical stage of block cluster fullness.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
blockFullness	The current computed level of block fullness of the cluster. • stage1Happy: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage2Aware: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage3Low: Your system cannot provide redundant data protection from two non-simultaneous node failures. Corresponds to the **Warning** state in the web UI. You can configure this level in the web UI (by default, the system triggers this alert at a capacity of 3% below the Error state). • stage4Critical: The system is not capable of providing redundant data protection from a single node failure. No new volumes or clones can be created. Corresponds to the **Error** state in the Element UI. • stage5CompletelyConsumed: Completely consumed. The cluster is read-only and iSCSI connections are maintained, but all writes are suspended. Corresponds to the **Critical** state in the Element UI.	string
fullness	Reflects the highest level of fullness between "blockFullness" and "metadataFullness".	string
maxMetadataOverProvisionFactor	A value representative of the number of times metadata space can be over provisioned relative to the amount of space available. For example, if there was enough metadata space to store 100 TiB of volumes and this number was set to 5, then 500 TiB worth of volumes could be created.	integer

Name	Description	Type
metadataFullness	The current computed level of metadata fullness of the cluster. • stage1Happy: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage2Aware: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage3Low: Your system cannot provide redundant data protection from two non-simultaneous node failures. Corresponds to the **Warning** state in the web UI. You can configure this level in the web UI (by default, the system triggers this alert at a capacity of 3% below the Error state). • stage4Critical: The system is not capable of providing redundant data protection from a single node failure. No new volumes or clones can be created. Corresponds to the **Error** state in the Element UI. • stage5CompletelyConsumed: Completely consumed. The cluster is read-only and iSCSI connections are maintained, but all writes are suspended. Corresponds to the **Critical** state in the Element UI.	string
sliceReserveUsedThresholdPct	Error condition. A system alert is triggered if the reserved slice utilization is greater than this value.	integer
stage2AwareThreshold	Awareness condition. The value that is set for the stage 2 cluster threshold level.	integer
stage2BlockThresholdBytes	The number of bytes being used by the cluster at which a stage 2 condition will exist.	integer

Name	Description	Type
stage2MetadataThresholdBytes	The number of metadata bytes being used by the cluster at which a stage 2 fullness condition will exist.	
stage3BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 3 fullness condition will exist.	integer
stage3BlockThresholdPercent	The percent value set for stage 3. At this percent full, a warning is posted in the Alerts log.	integer
stage3LowThreshold	Error condition. The threshold at which a system alert is created due to low capacity on a cluster.	integer
stage3MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 3 fullness condition will exist.	integer
stage3MetadataThresholdPercent	The percent value set for stage3 of metadata fullness. At this percent full, a warning will be posted in the Alerts log.	integer
stage4BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 4 fullness condition will exist.	integer
stage4CriticalThreshold	Error condition. The threshold at which a system alert is created to warn about critically low capacity on a cluster.	integer
stage4MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 4 fullness condition will exist.	integer
stage5BlockThresholdBytes	The number of storage bytes used by the cluster at which a stage 5 fullness condition will exist.	integer
stage5MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 5 fullness condition will exist.	integer

Name	Description	Type
sumTotalClusterBytes	The physical capacity of the cluster, measured in bytes.	integer
sumTotalMetadataClusterBytes	The total amount of space that can be used to store metadata.	integer
sumUsedClusterBytes	The number of storage bytes used on the cluster.	integer
sumUsedMetadataClusterBytes	The amount of space used on volume drives to store metadata.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method" : "GetClusterFullThreshold",
   "params" : {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id":1,
  "result":{
    "blockFullness":"stage1Happy",
    "fullness":"stage3Low",
    "maxMetadataOverProvisionFactor":5,
    "metadataFullness":"stage3Low",
    "sliceReserveUsedThresholdPct":5,
    "stage2AwareThreshold":3,
    "stage2BlockThresholdBytes":2640607661261,
    "stage3BlockThresholdBytes":8281905846682,
    "stage3BlockThresholdPercent":5,
    "stage3LowThreshold":2,
    "stage4BlockThresholdBytes":8641988709581,
    "stage4CriticalThreshold":1,
    "stage5BlockThresholdBytes":12002762096640,
    "sumTotalClusterBytes":12002762096640,
    "sumTotalMetadataClusterBytes":404849531289,
    "sumUsedClusterBytes":45553617581,
    "sumUsedMetadataClusterBytes":31703113728
  }
}
```


New since version

9.6

Find more information

ModifyClusterFullThreshold

GetClusterHardwareInfo

You can use the `GetClusterHardwareInfo` method to retrieve the hardware status and information for all Fibre Channel nodes, iSCSI nodes and drives in the cluster. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
type	Include only one of the following types of hardware information in the response. Possible values: • drives: Lists only drive information in the response. • nodes: Lists only node information in the response. • all: Includes both drive and node information in the response. If this parameter is omitted, a type of all is assumed.	string	all	No

Return value

This method has the following return value:

Name	Description	Type
clusterHardwareInfo	Hardware information for all nodes and drives in the cluster. Each object in this output is labeled with the nodeID of the given node.	hardwareInfo

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetClusterHardwareInfo",
    "params": {
        "type": "all"
    },
    "id": 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

GetClusterHardwareInfo

GetClusterInfo

You can use the `GetClusterInfo` method to return configuration information about the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
clusterInfo	Cluster information.	clusterInfo

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetClusterInfo",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "clusterInfo": {
            "attributes": {},
            "defaultProtectionScheme": "doubleHelix",
            "enabledProtectionSchemes": [
                "doubleHelix"
            ],
            "encryptionAtRestState": "disabled",
            "ensemble": [
                "10.10.10.32",
                "10.10.10.34",
                "10.10.10.35",
                "10.10.10.36",
                "10.10.10.37"
            ],
            "mvip": "10.10.11.225",
            "mvipInterface": "team1G",
            "mvipNodeID": 3,
            "mvipVlanTag": "0",
            "name": "ClusterName",
            "repCount": 2,
            "softwareEncryptionAtRestState": "enabled",
            "supportedProtectionSchemes": [
                "doubleHelix"
            ],
            "svip": "10.10.10.111",
            "svipInterface": "team10G",
            "svipNodeID": 3,
            "svipVlanTag": "0",
            "uniqueID": "psmp",
            "uuid": "2f575d0c-36fe-406d-9d10-dbc1c306ade7"
        }
    }
}
```


New since version

9.6

GetClusterInterfacePreference

The `GetClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to get information about an existing cluster

interface preference. This method is for internal use.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference.	string	None	Yes

Return value

This method has the following return value:

Name	Description	Type
preference	The name and value of the requested cluster interface preference.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
    "method": "GetClusterInterfacePreference",
    "params": {
        "name": "prefname"
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "preference": {
            "name": "prefname",
            "value": "testvalue"
        }
    }
}
```

New since version

11.0

GetClusterMasterNodeID

You can use the `GetClusterMasterNodeID` method to retrieve the ID of the node that performs cluster-wide administration tasks and holds the storage virtual IP address (SVIP) and management virtual IP address (MVIP).

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
nodeID	ID of the master node.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetClusterMasterNodeID",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1
  "result": {
    "nodeID": 1
   }
}
```

New since version

9.6

GetClusterStats

You can use the `GetClusterStats` method to retrieve high-level activity measurements for the cluster. Values returned are cumulative from the creation of the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
clusterStats	Cluster activity information.	clusterStats

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetClusterStats",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "clusterStats": {
      "actualIOPS": 9376,
      "averageIOPSize": 4198,
      "clientQueueDepth": 8,
      "clusterUtilization": 0.09998933225870132,
      "latencyUSec": 52,
      "normalizedIOPS": 15000,
      "readBytes": 31949074432,
      "readBytesLastSample": 30883840,
      "readLatencyUSec": 27,
      "readLatencyUSecTotal": 182269319,
      "readOps": 1383161,
      "readOpsLastSample": 3770,
      "samplePeriodMsec": 500,
      "servicesCount": 3,
      "servicesTotal": 3,
      "timestamp": "2017-09-09T21:15:39.809332Z",
      "unalignedReads": 0,
      "unalignedWrites": 0,
      "writeBytes": 8002002944,
      "writeBytesLastSample": 7520256,
      "writeLatencyUSec": 156,
      "writeLatencyUSecTotal": 231848965,
      "writeOps": 346383,
      "writeOpsLastSample": 918
    }
  }
}
```


New since version

9.6

GetClusterVersionInfo

You can use the `GetClusterVersionInfo` method to retrieve information about the Element software version running on each node in the cluster. This method also returns information about nodes that are currently in the process of upgrading software.

Cluster version info object members

This method has the following object members:

Name	Description	Type
nodeID	ID of the node.	integer
nodeInternalRevision	Internal software version of the node.	string
nodeVersion	Software version of the node.	string

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
clusterAPIVersion	The current API version on the cluster.	string
clusterVersion	Version of Element software currently running on the cluster.	string
clusterVersionInfo	List of nodes in the cluster with version information for each node.	JSON object array
pendingClusterVersion	If present, this is the version that the cluster software is currently being upgraded or reverted to.	string

Name	Description	Type
softwareVersionInfo	The state of an upgrade. Object members: • currentVersion: The current software version on a node. • nodeID: ID of the node being upgraded from currentVersion to pendingVersion. This field is 0 (zero) if there is no upgrade in progress. • packageName: Name of the software package being installed. • pendingVersion: The version of the software being installed. • startTime: The date and time the installation was started, in UTC+0 format.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetClusterVersionInfo",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id": 1,
 "result": {
   "clusterAPIVersion": "6.0",
   "clusterVersion": "6.1382",
   "clusterVersionInfo": [
   {
      "nodeID": 1,
      "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
      "nodeVersion": "6.1382"
   },
   {
      "nodeID": 2,
      "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
      "nodeVersion": "6.1382"
   },
   {
      "nodeID": 3,
      "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
      "nodeVersion": "6.1382"
   },
   {
      "nodeID": 4,
      "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
      "nodeVersion": "6.1382"
   }
 ],
   "softwareVersionInfo": {
      "currentVersion": "6.1382",
      "nodeID": 0,
      "packageName": "",
      "pendingVersion": "6.1382",
      "startTime": ""
    }
  }
}
```

New since version

9.6

GetFeatureStatus

You can use the `GetFeatureStatus` method to retrieve the status of a cluster feature.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
feature	The status of a cluster feature. If no value is provided, the system returns a status of all features. Possible values: • Vvols: Retrieve status for the VVols cluster feature. • SnapMirror: Retrieve status for the SnapMirror replication cluster feature. • Fips: Retrieve status for the FIPS 140-2 encryption for HTTPS communication feature. • fipsDrives: Retrieve status for the FIPS 140-2 drive encryption feature.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
features	An array of feature objects indicating the feature name and its status. Object members: • feature: (string) The name of the feature. • enabled: (boolean) Whether the feature is enabled or not.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetFeatureStatus",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "features": [
            {
                "enabled": true,
                "feature": "Vvols"
            },
            {
                "enabled": true,
                "feature": "SnapMirror"
            },
            {
                "enabled": true,
                "feature": "Fips"
            },
            {
                "enabled": true,
                "feature": "FipsDrives"
            }
        ]
    }
}
```


New since version

9.6

GetLoginSessionInfo

You can use the `GetLoginSessionInfo` method to return the period of time a login authentication session is valid for both login shells and the TUI.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
loginSessionInfo	An object containing the authentication expiration period. Possible objects returned: • timeout: The time, in minutes, when this session will timeout and expire. Formatted in H:mm:ss. For example: 1:30:00, 20:00, 5:00. All leading zeros and colons are removed regardless of the format the timeout was entered.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetLoginSessionInfo",
    "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
  "result" : {
    "loginSessionInfo" : {
      "timeout" : "30:00"
    }
  }
}
```

New since version

9.6

GetNodeHardwareInfo

You can use the `GetNodeHardwareInfo` method to return all the hardware information and status for the node specified. This generally includes manufacturers, vendors,

versions, and other associated hardware identification information.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
nodeID	The ID of the node for which hardware information is being requested. Information about a Fibre Channel node is returned if a Fibre Channel node is specified.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
nodeHardwareInfo	Hardware information for the specified nodeID. Each object in this output is labeled with the nodeID of the given node.	hardwareInfo

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetNodeHardwareInfo",
   "params": {
     "nodeID": 1
   },
"id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

GetNodeHardwareInfo (output for Fibre Channel nodes)

GetNodeHardwareInfo (output for iSCSI)

GetNodeStats

You can use the `GetNodeStats` method to retrieve the high-level activity measurements for a single node.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
nodeID	Specifies the ID of the node for which statistics will be returned.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
nodeStats	Node activity information.	nodeStats

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetNodeStats",
   "params": {
     "nodeID": 5
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
     "nodeStats" : {
       "cBytesIn" : 9725856460404,
       "cBytesOut" : 16730049266858,
       "cpu" : 98,
       "mBytesIn" : 50808519,
       "mBytesOut" : 52040158,
       "networkUtilizationCluster" : 84,
       "networkUtilizationStorage" : 0,
       "sBytesIn" : 9725856460404,
       "sBytesOut" : 16730049266858,
       "timestamp" : "2012-05-16T19:14:37.167521Z",
       "usedMemory" : 41195708000
     }
   }
}
```

New since version

9.6

ListActiveNodes

You can use the `ListActiveNodes` method to return the list of currently active nodes that are in the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
nodes	List of active nodes in the cluster.	node array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListActiveNodes",
   "params": {},
   "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

ListActiveNodes

ListAllNodes

You can use the `ListAllNodes` method to list active and pending nodes in the cluster.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
nodes	List of objects describing active nodes in the cluster.	node
pendingActiveNodes	List of objects describing pending active nodes for the cluster.	pendingActiveNode array
pendingNodes	List of objects describing pending nodes for the cluster.	pendingNode array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListAllNodes",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "nodes": [
            {
                "associatedFServiceID": 0,
                "associatedMasterServiceID": 1,
                "attributes": {},
                "chassisName": "CT5TV12",
                "cip": "10.1.1.1",
                "cipi": "Bond10G",
                "fibreChannelTargetPortGroup": null,
                "mip": "10.1.1.1",
                "mipi": "Bond1G",
                "name": "NLABP0704",
                "nodeID": 1,
                "nodeSlot": "",
                "platformInfo": {
                    "chassisType": "R620",
                    "cpuModel": "Intel",
                    "nodeMemoryGB": 72,
                    "nodeType": "SF3010",
                    "platformConfigVersion": "0.0.0.0"
                },
                "sip": "10.1.1.1",
                "sipi": "Bond10G",
                "softwareVersion": "11.0",
                "uuid": "4C4C4544-0054",
                "virtualNetworks": []
            }
        ],
        "pendingActiveNodes": [],
        "pendingNodes": []
    }
}
```


New since version

9.6

ListClusterFaults

You can use the `ListClusterFaults` method to list information about any faults detected on the cluster. With this method, you can list both current faults as well as faults that have been resolved. The system caches faults every 30 seconds.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
bestPractices	Include faults triggered by sub-optimal system configuration. Possible values: • true • false	boolean	None	No
faultTypes	Determines the types of faults returned. Possible values: • current: List active, unresolved faults. • resolved: List faults that were previously detected and resolved. • all: List both current and resolved faults. You can see the fault status in the “resolved” member of the fault object.	string	all	No

Return value

This method has the following return value:

Name	Description	Type
faults	An object describing the requested cluster faults.	fault

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListClusterFaults",
    "params": {
      "faultTypes": "current",
      "bestPractices": true
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "faults": [
      {
        "blocksUpgrade": false,
        "clusterFaultID": 3,
        "code": "driveAvailable",
        "data": null,
        "date": "2024-04-03T22:22:56.660275Z",
        "details": "Node ID 1 has 6 available drive(s).",
        "driveID": 0,
        "driveIDs": [],
        "externalSource": "",
        "networkInterface": "",
        "nodeHardwareFaultID": 0,
        "nodeID": 1,
        "resolved": true,
        "resolvedDate": "2024-04-03T22:24:54.598693Z",
        "serviceID": 0,
        "severity": "warning",
        "type": "drive"
      },
      {
        "clusterFaultID": 9,
        "code": "disconnectedClusterPair",
        "data": null,
        "date": "2016-04-26T20:40:08.736597Z",
        "details": "One of the clusters in a pair may have become
misconfigured or disconnected.  Remove the local pairing and retry pairing
the clusters. Disconnected Cluster Pairs: []. Misconfigured Cluster Pairs:
[3]",
        "driveID": 0,
        "driveIDs": [],
        "nodeHardwareFaultID": 0,
        "nodeID": 0,
        "resolved": false,
        "resolvedDate": "",
        "serviceID": 0,
        "severity": "warning",
        "type": "cluster"
      }
    ]
  }
}
```

New since version

9.6

ListClusterInterfacePreferences

The `ListClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to list the existing cluster interface preferences stored on the system. This method is for internal use.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
preferences	A list of cluster interface objects currently stored on the storage cluster, each containing the name and value of the preference.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "ListClusterInterfacePreferences",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "preferences": [
            {
                "name": "prefname",
                "value": "testvalue"
            }
        ]
    }
}
```

New since version

11.0

ListEvents

You can use the `ListEvents` method to list events detected on the cluster, sorted from oldest to newest.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
driveID	Specifies that only events with this drive ID will be returned.	integer	0	No
endEventID	Identifies the end of a range of event IDs to return.	integer	(unlimited)	No
endPublishTime	Specifies that only events published earlier than this time will be returned.	string	0	No
endReportTime	Specifies that only events reported earlier than this time will be returned.	string	0	No

Name	Description	Type	Default value	Required
eventType	Specifies the type of events to return. See event for possible event types.	string	0	No
maxEvents	Specifies the maximum number of events to return.	integer	(unlimited)	No
nodeID	Specifies that only events with this node ID will be returned.	integer		
serviceID	Specifies that only events with this service ID will be returned.			
startEventID	Identifies the beginning of a range of events to return.	integer	0	No
startPublishTime	Specifies that only events published after this time will be returned.	string	0	No
startReportTime	Specifies that only events reported after this time will be returned.	string	0	No

Return value

This method has the following return value:

Name	Description	Type
events	List of events.	event array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListEvents",
   "params": {
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id":1,
  "result":{
    "events":[
       {
          "details":
            {
              "paramGCGeneration":1431550800,
              "paramServiceID":2
            },
          "driveID":0,
          "eventID":2131,
          "eventInfoType":"gcEvent",
          "message":"GC Cluster Coordination Complete",
          "nodeID":0,
          "serviceID":2,
          "severity":0,
          "timeOfPublish":"2015-05-13T21:00:02.361354Z",
          "timeOfReport":"2015-05-13T21:00:02.361269Z"
       },{
          "details":
               {

"eligibleBS":[5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,24,25,26,27,28,29,30
,31,40,41,42,43,44,45,46,47,52,53,54,55,56,57,58,59,60],
                  "generation":1431550800,
                  "participatingSS":[23,35,39,51]
               },
          "driveID":0,
          "eventID":2130,
          "eventInfoType":"gcEvent",
          "message":"GCStarted",
          "nodeID":0,
          "serviceID":2,
```

```
            "severity":0,
            "timeOfPublish":"2015-05-13T21:00:02.354128Z",
            "timeOfReport":"2015-05-13T21:00:02.353894Z"
         },{
            "details":"",
            "driveID":0,
            "eventID":2129,
            "eventInfoType":"tSEvent",
            "message":"return code:2 t:41286 tt:41286 qcc:1 qd:1 qc:1 vrc:1
tt:2 ct:Write etl:524288",
            "nodeID":0,
            "serviceID":0,
            "severity":0,
            "timeOfPublish":"2015-05-13T20:45:21.586483Z",
            "timeOfReport":"2015-05-13T20:45:21.586311Z"
         }
      ]
   }
}
```


New since version

9.6

ListNodeStats

You can use the `ListNodeStats` method to view the high-level activity measurements for all storage nodes in a storage cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
nodeStats	Storage node activity information.	nodeStats

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListNodeStats",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "nodeStats": {
       "nodes": [
         {
         "cBytesIn": 46480366124,
         "cBytesOut": 46601523187,
         "cpu": 0,
         "mBytesIn": 59934129,
         "mBytesOut": 41620976,
         "networkUtilizationCluster": 0,
         "networkUtilizationStorage": 0,
         "nodeID": 1,
         "sBytesIn": 46480366124,
         "sBytesOut": 46601523187,
         "timestamp": 1895558254814,
         "usedMemory": 31608135680
         }
       ]
     }
 }
```

New since version

9.6

ListISCSISessions

You can use the `ListISCSISessions` method to list iSCSI connection information for volumes in the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
sessions	Information about each iSCSI session.	session

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListISCSISessions",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
 {
  "id": 1,
  "result": {
    "sessions": [
      {
        "accountID": 1,
        "accountName": "account1",
        "authentication": {
            "authMethod": "CHAP",
            "chapAlgorithm": "SHA3_256",
            "chapUsername": "iqn.1994-05.com."redhat:1cf11f3eed3",
            "direction": "two-way"
        },
        "createTime": "2022-10-03T22:02:49.121723Z",
        "driveID": 23,
        "driveIDs": [23],
        "initiator": null,
        "initiatorIP": "10.1.1.1:37138",
        "initiatorName": "iqn.2010-01.net.solidfire.eng:c",
        "initiatorPortName": "iqn.2010-
01.net.solidfire.eng:c,i,0x23d860000",
        "initiatorSessionID": 9622126592,
        "msSinceLastIscsiPDU": 243,
        "msSinceLastScsiCommand": 141535021,
        "nodeID": 3,
        "serviceID": 6,
        "sessionID": 25769804943,
        "targetIP": "10.1.1.2:3260",
        "targetName": "iqn.2010-01.com.solidfire:a7sd.3",
        "targetPortName": "iqn.2010-01.com.solidfire:a7sd.3,t,0x1",
        "virtualNetworkID": 0,
        "volumeID": 3,
        "volumeInstance": 140327214758656
      }
     ...
    ]
  }
}
```


New since version

9.6

ListServices

You can use the `ListServices` method to list services information for nodes, drives, current software, and other services that are running on the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
services	Services that are running on drives and nodes.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListServices",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
"id": 1,
"result": {
    "services": [
        {
            "drive": {
                "assignedService": 22,
                "asyncResultIDs": [],
                "attributes": {},
                "capacity": 300069052416,
                "customerSliceFileCapacity": 0,
                "driveID": 5,
                "driveStatus": "assigned",
                "driveType": "block",
                "failCount": 0,
                "nodeID": 4,
```

```
            "reservedSliceFileCapacity": 0,
            "serial": "scsi-SATA_INTEL_SSDSC2",
            "slot": 3
        },
        "drives": [
            {
                "assignedService": 22,
                "asyncResultIDs": [],
                "attributes": {},
                "capacity": 300069052416,
                "customerSliceFileCapacity": 0,
                "driveID": 5,
                "driveStatus": "assigned",
                "driveType": "Block",
                "failCount": 0,
                "nodeID": 4,
                "reservedSliceFileCapacity": 0,
                "serial": "scsi-SATA_INTEL_SSDSC2",
                "slot": 3
            }
        ],
        "node": {
            "associatedFServiceID": 0,
            "associatedMasterServiceID": 1,
            "attributes": {},
            "cip": "10.117.63.18",
            "cipi": "Bond10G",
            "fibreChannelTargetPortGroup": null,
            "mip": "10.117.61.18",
            "mipi": "Bond1G",
            "name": "node4",
            "nodeID": 4,
            "nodeSlot": "",
            "platformInfo": {
                "chassisType": "R620",
                "cpuModel": "Intel(R) Xeon(R) CPU",
                "nodeMemoryGB": 72,
                "nodeType": "SF3010",
                "platformConfigVersion": "10.0"
            },
            "sip": "10.117.63.18",
            "sipi": "Bond10G",
            "softwareVersion": "10.0",
            "uuid": "4C4C4544-0053",
            "virtualNetworks": []
        },
```

```
            "service": {
                "associatedBV": 0,
                "associatedTS": 0,
                "associatedVS": 0,
                "asyncResultIDs": [
                    1
                ],
                "driveID": 5,
                "driveIDs": [
                    5
                ],
                "firstTimeStartup": true,
                "ipcPort": 4008,
                "iscsiPort": 0,
                "nodeID": 4,
                "serviceID": 22,
                "serviceType": "block",
                "startedDriveIDs": [],
                "status": "healthy"
            }
        }
    ]
}
```


New since version

9.6

ListPendingNodes

You can use the `ListPendingNodes` method to list the pending storage nodes in the system. Pending nodes are storage nodes that are running and configured to join the storage cluster but have not yet been added using the AddNodes API method.

IPv4 and IPv6 management addresses

Note that `ListPendingNodes` does not list pending nodes that have different address types for the management IP address (MIP) and management virtual IP address (MVIP). For example, if a pending node has an IPv6 MVIP and an IPv4 MIP, `ListPendingNodes` will not include the node as part of the result.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
pendingNodes	List of pending nodes in the cluster.	pendingNode array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListPendingNodes",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 3,
  "result": {
    "pendingNodes": [
      {
        "assignedNodeID": 0,
        "cip": "10.26.65.101",
        "cipi": "Bond10G",
        "compatible": true,
        "mip": "172.26.65.101",
        "mipi": "Bond1G",
        "name": "VWC-EN101",
        "pendingNodeID": 1,
        "platformInfo": {
          "chassisType": "R620",
          "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
          "nodeMemoryGB": 72,
          "nodeType": "SF3010"
        },
        "sip": "10.26.65.101",
        "sipi": "Bond10G",
        "softwareVersion": "9.0.0.1554",
        "uuid": "4C4C4544-0048-4410-8056-C7C04F395931"
      }
    ]
  }
}
```


New since version

9.6

Find more information

AddNodes

ListPendingActiveNodes

You can use the `ListPendingActiveNodes` method to list nodes in the cluster that are in the PendingActive state, between pending and active states. Nodes in this state are being returned to the factory image.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
pendingActiveNodes	List of objects detailing information about all PendingActive nodes in the system.	pendingActiveNode array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListPendingActiveNodes",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  id: null,
  result: {
    pendingActiveNodes: [
    {
      activeNodeKey: "5rPHP3lTAO",
      assignedNodeID: 5,
      asyncHandle: 2,
      cip: "10.10.5.106",
      mip: "192.168.133.106",
      pendingNodeID: 1,
      platformInfo: {
        chassisType: "R620",
        cpuModel: "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
        nodeMemoryGB: 72,
        nodeType: "SF3010"
      },
     sip: "10.10.5.106",
     softwareVersion: "9.0.0.1077"
    }
    ]
  }
}
```

New since version

9.6

ModifyClusterFullThreshold

You can use the `ModifyClusterFullThreshold` method to change the level at which the system generates an event when the storage cluster approaches a certain capacity utilization. You can use the threshold setting to indicate the acceptable amount of utilized block storage before the system generates a warning.

For example, if you want to be alerted when the system reaches 3% below the "Error" level block storage utilization, enter a value of "3" for the stage3BlockThresholdPercent parameter. If this level is reached, the system sends an alert to the Event Log in the Cluster Management Console.

Parameters

This method has the following input parameters:

You must select at least one parameter.

Name	Description	Type	Default value	Required
maxMetadataOverProvisionFactor	A value representative of the number of times metadata space can be over provisioned relative to the amount of space available. For example, if there was enough metadata space to store 100 TiB of volumes and this number was set to 5, then 500 TiB worth of volumes could be created.	integer	5	No
stage2AwareThreshold	The number of nodes of capacity remaining in the cluster before the system triggers a capacity notification.	integer	None	No
stage3BlockThresholdPercent	The percentage of block storage utilization below the "Error" threshold that causes the system to trigger a cluster "Warning" alert.	integer	None	No
stage3MetadataThresholdPercent	The percentage of metadata storage utilization below the “Error” threshold that causes the system to trigger a cluster “Warning” alert.	integer	None	No

Return values

This method has the following return values:

Name	Description	Type

blockFullness	The current computed level of block fullness of the cluster. • stage1Happy: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage2Aware: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage3Low: Your system cannot provide redundant data protection from two non-simultaneous node failures. Corresponds to the **Warning** state in the web UI. You can configure this level in the web UI (by default, the system triggers this alert at a capacity of 3% below the Error state). • stage4Critical: The system is not capable of providing redundant data protection from a single node failure. No new volumes or clones can be created. Corresponds to the **Error** state in the Element UI. • stage5CompletelyConsumed: Completely consumed. The cluster is read-only and iSCSI connections are maintained, but all writes are suspended. Corresponds to the **Critical** state in the Element UI.	string
fullness	Reflects the highest level of fullness between "blockFullness" and "metadataFullness".	string
maxMetadataOverProvisionFactor	A value representative of the number of times metadata space can be over provisioned relative to the amount of space available. For example, if there was enough metadata space to store 100 TiB of volumes and this number was set to 5, then 500 TiB worth of volumes could be created.	integer

metadataFullness	The current computed level of metadata fullness of the cluster. • stage1Happy: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage2Aware: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage3Low: Your system cannot provide redundant data protection from two non-simultaneous node failures. Corresponds to the **Warning** state in the web UI. You can configure this level in the web UI (by default, the system triggers this alert at a capacity of 3% below the Error state). • stage4Critical: The system is not capable of providing redundant data protection from a single node failure. No new volumes or clones can be created. Corresponds to the **Error** state in the Element UI. • stage5CompletelyConsumed: Completely consumed. The cluster is read-only and iSCSI connections are maintained, but all writes are suspended. Corresponds to the **Critical** state in the Element UI.	string
sliceReserveUsedThresholdPct	Error condition. A system alert is triggered if the reserved slice utilization is greater than the sliceReserveUsedThresholdPct value returned.	integer
stage2AwareThreshold	Awareness condition. The value that is set for "Stage 2" cluster threshold level.	integer
stage2BlockThresholdBytes	The number of bytes being used by the cluster at which a stage 2 fullness condition will exist.	integer

stage2MetadataThresholdBytes	The number of metadata bytes being used by the cluster at which a stage 2 fullness condition will exist.	
stage3BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 3 fullness condition will exist.	integer
stage3BlockThresholdPercent	The percent value set for stage 3. At this percent full, a warning is posted in the Alerts log.	integer
stage3LowThreshold	Error condition. The threshold at which a system alert is created due to low capacity on a cluster.	integer
stage3MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 3 fullness condition will exist.	
stage4BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 4 fullness condition will exist.	integer
stage4CriticalThreshold	Error condition. The threshold at which a system alert is created to warn about critically low capacity on a cluster.	integer
stage4MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 4 fullness condition will exist.	
stage5BlockThresholdBytes	The number of storage bytes used by the cluster at which a stage 5 fullness condition will exist.	integer
stage5MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 5 fullness condition will exist.	
sumTotalClusterBytes	The physical capacity of the cluster, measured in bytes.	integer
sumTotalMetadataClusterBytes	The total amount of space that can be used to store metadata.	integer

sumUsedClusterBytes	The number of storage bytes used on the cluster.	integer
sumUsedMetadataClusterBytes	The amount of space used on volume drives to store metadata.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method" : "ModifyClusterFullThreshold",
   "params" : {
              "stage3BlockThresholdPercent" : 3
              },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "blockFullness": "stage1Happy",
    "fullness": "stage3Low",
    "maxMetadataOverProvisionFactor": 5,
    "metadataFullness": "stage3Low",
    "sliceReserveUsedThresholdPct": 5,
    "stage2AwareThreshold": 3,
    "stage2BlockThresholdBytes": 2640607661261,
    "stage3BlockThresholdBytes": 8281905846682,
    "stage3BlockThresholdPercent": 3,
    "stage3LowThreshold": 2,
    "stage4BlockThresholdBytes": 8641988709581,
    "stage4CriticalThreshold": 1,
    "stage5BlockThresholdBytes": 12002762096640,
    "sumTotalClusterBytes": 12002762096640,
    "sumTotalMetadataClusterBytes": 404849531289,
    "sumUsedClusterBytes": 45553617581,
    "sumUsedMetadataClusterBytes": 31703113728
  }
}
```

New since version

9.6

ModifyClusterInterfacePreference

The `ModifyClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to change an existing cluster interface preference. This method is for internal use.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference to modify.	string	None	Yes
value	The new value of the cluster interface preference.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
    "method": "ModifyClusterInterfacePreference",
    "params": {
    "name": "testname",
    "value": "newvalue"
},
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {}
}
```

New since version

11.0

RemoveNodes

You can use `RemoveNodes` to remove one or more nodes that should no longer participate in the cluster.

Before removing a node, you must remove all drives the node contains using the `RemoveDrives` method. You cannot remove a node until the `RemoveDrives` process has completed and all data has been migrated away from the node. After you remove a node, it registers itself as a pending node. You can add the node again or shut it down (shutting the node down removes it from the pending node list).

Cluster master node removal

If you use `RemoveNodes` to remove the cluster master node, the method might time out before returning a response. If the method call fails to remove the node, make the method call again. Note that if you are removing the cluster master node along with other nodes, you should use a separate call to remove the cluster master node by itself.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
ignoreEnsembleToleranceChange	Ignore changes to the ensemble's node failure tolerance when removing nodes. If the storage cluster uses data protection schemes that tolerate multiple node failures and removing the nodes would decrease the node failure tolerance of the ensemble, the node removal normally fails with an error. You can set this parameter to true to disable the ensemble tolerance check so that the node removal succeeds.	boolean	false	No
nodes	List of NodeIDs for the nodes to be removed.	integer array	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveNodes",
   "params": {
     "nodes"  : [3,4,5]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1
  "result" : {},
}
```

New since version

9.6

SetLoginSessionInfo

You can use the `SetLoginSessionInfo` method to set the period of time that a login authentication for a session is valid. After the login period elapses without activity on the system, the authentication expires. New login credentials are required for continued access to the cluster after the login period has elapsed.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
timeout	Cluster authentication expiration period. Formatted in HH:mm:ss. For example: 01:30:00, 00:90:00, and 00:00:5400 can all be used to equal a 90 minute timeout period. The minimum timeout value is 1 minute. When a value is not provided, or is set to zero, the login session has no timeout value.	string	30 minutes	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetLoginSessionInfo",
  "params": {
    "timeout" : "01:30:00"
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

9.6

Shutdown

You can use the `Shutdown` method to restart or shutdown the nodes in a cluster. You can shut down a single node, multiple nodes, or all of the nodes in the cluster using this method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
nodes	List of NodeIDs for the nodes to be restarted or shut down.	integer array	None	Yes

Name	Description	Type	Default value	Required
option	Action to take for the cluster. Possible values: • restart: Restarts the cluster. • halt: Performs a full power-off.	string	restart	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "Shutdown",
   "params": {
     "nodes": [
       2,
       3,
       4
     ],
     "option": "halt"
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
     "failed": [],
     "successful": [
          6
      [
   }
}
```

New since version

9.6

Cluster creation API Methods

You can use these API methods to create a storage cluster. All of these methods need to be used against the API endpoint on a single node.

- CheckProposedCluster
- CreateCluster
- GetBootstrapConfig

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

CheckProposedCluster

You can use the `CheckProposedCluster` method to test a set of storage nodes before creating a storage cluster with them to identify possible errors or faults that would occur from the attempt, such as unbalanced mixed node capabilities or node types that are not supported for two-node storage clusters.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
nodes	A list of storage IP addresses of the initial set of storage nodes making up the storage cluster.	string array	None	Yes
force	Set to true to run on all storage nodes in the storage cluster.	boolean	None	No

Return values

This method has the following return values:

Name	Description	Type

proposedClusterValid	Indicates whether or not the proposed storage nodes would make up a valid storage cluster. Possible values: • true • false	boolean
proposedClusterErrors	Errors that would occur if a storage cluster was created using the proposed storage nodes.	string array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "CheckProposedCluster",
    "params": {
    "nodes": [
        "192.168.1.11",
        "192.168.1.12",
        "192.168.1.13",
        "192.168.1.14"
    ]
},
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "proposedClusterValid": true,
        "proposedClusterErrors": [ ]
    }
}
```

New since version

11.0

CreateCluster

You can use the `CreateCluster` method to initialize the node in a cluster that has ownership of the "mvip" and "svip" addresses. Each new cluster is initialized using the management IP (MIP) of the first node in the cluster. This method also automatically adds all the nodes being configured into the cluster. You only need to use this method once each time a new cluster is initialized.

After you log in to the master node for the cluster and run the GetBootStrapConfig method to get the IP addresses for the rest of the nodes that you want to include in the cluster, you can run the CreateCluster method against the master node for the cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
acceptEula	Indicate your acceptance of the End User License Agreement when creating this cluster. To accept the EULA, set this parameter to true.	boolean	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
enableSoftwareEncryptionAtRest	Enable this parameter to use software-based encryption at rest. Defaults to false on all clusters. After software encryption at rest is enabled, it cannot be disabled on the cluster.	boolean	true	No
mvip	Floating (virtual) IP address for the cluster on the management network.	string	None	Yes

Name	Description	Type	Default value	Required
nodes	CIP/SIP addresses of the initial set of nodes making up the cluster. This node's IP must be in the list.	string array	None	Yes
orderNumber	Alphanumeric sales order number. Required on software-based platforms.	string	None	No (hardware-based platforms) Yes (software-based platforms)
password	Initial password for the cluster admin account.	string	None	Yes
serialNumber	Nine-digit alphanumeric Serial Number. May be required on software-based platforms.	string	None	No (hardware-based platforms) Yes (software-based platforms)
svip	Floating (virtual) IP address for the cluster on the storage (iSCSI) network.	string	None	Yes
username	User name for the cluster admin.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CreateCluster",
  "params": {
    "acceptEula": true,
    "mvip": "10.0.3.1",
    "svip": "10.0.4.1",
    "username": "Admin1",
    "password": "9R7ka4rEPa2uREtE",
    "attributes": {
      "clusteraccountnumber": "axdf323456"
    },
    "nodes": [
      "10.0.2.1",
      "10.0.2.2",
      "10.0.2.3",
      "10.0.2.4"
    ]
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id" : 1,
"result" : {}
}
```

New since version

9.6

Find more information

- GetBootstrapConfig
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

GetBootstrapConfig

You can use the `GetBootstrapConfig` method to get cluster and node information from the bootstrap configuration file. Use this API method on an individual node before it has been joined with a cluster. The information this method returns is used in the cluster

configuration interface when you create a cluster.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
clusterName	Name of the cluster.	string
mvip	Cluster MVIP address. Empty if the node is not part of a cluster.	string
nodeName	Name of the node.	string
nodes	List of information about each node that is actively waiting to join the cluster. Possible values: • chassisType: (string) Hardware platform of the node. • cip: (string) Cluster IP address of the node. • compatible: (boolean) Indicates if the node is compatible with the node the API call was executed against. • hostname: (string) Host name of the node. • mip: (string) The IPv4 management IP address of the node. • mipV6: (string) The IPv6 management IP address of the node. • nodeType: (string)Model name of the node. • version: (string)Version of software currently installed on the node.	JSON object array
svip	Cluster SVIP address. Null if the node is not part of a cluster.	string

Name	Description	Type
version	Version of Element software currently installed on the node that was called by this API method.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetBootstrapConfig",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id":1,
    "result":{
        "clusterName":"testname",
        "nodeName":"testnode",
        "svip": "10.117.1.5",
        "mvip": "10.117.1.6",
        "nodes":[
            {
                "chassisType":"R630",
                "cip":"10.117.115.16",
                "compatible":true,
                "hostname":"NLABP1132",
                "mip":"10.117.114.16",
                "mipV6":"fd20:8b1e:b256:45a::16",
                "nodeType":"SF2405",
                "role":"Storage",
                "version":"11.0"
            },
            {
                "chassisType":"R630",
                "cip":"10.117.115.17",
                "compatible":true,
                "hostname":"NLABP1133",
                "mip":"10.117.114.17",
                "mipV6":"fd20:8b1e:b256:45a::17",
                "nodeType":"SF2405",
                "role":"Storage",
                "version":"11.0"
            },
            {
                "chassisType":"R630",
                "cip":"10.117.115.18",
                "compatible":true,
                "hostname":"NLABP1134",
                "mip":"10.117.114.18",
                "mipV6":"fd20:8b1e:b256:45a::18",
                "nodeType":"SF2405",
                "role":"Storage",
                "version":"11.0"
            }
        ],
        "version":"11.0"
    }
}
```

New since version

9.6

Find more information

CreateCluster

Drive API methods

You can use drive API methods to add and manage drives that are available to a storage cluster. When you add a storage node to the storage cluster or install new drives in an existing storage node, the drives are available to be added to the storage cluster.

- AddDrives
- GetDriveHardwareInfo
- GetDriveStats
- ListDrives
- ListDriveStats
- RemoveDrives
- SecureEraseDrives

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddDrives

You can use the `AddDrives` method to add one or more available drives to the cluster, enabling the drives to host a portion of the data for the cluster.

When you add a storage node to the cluster or install new drives in an existing node, the new drives are marked as available and must be added via AddDrives before they can be utilized. Use the ListDrives method to display drives that are available to be added. When you add a drive, the system automatically determines the type of drive it should be.

The method is asynchronous and returns as soon as the processes for rebalancing the drives in the cluster are started. However, it might take more time for the data in the cluster to be rebalanced using the newly added drives; the rebalancing continues even after the AddDrives method call is complete. You can use the GetAsyncResult method to query the method's returned asyncHandle. After the AddDrives method returns, you can use the ListSyncJobs method to see the progress of the rebalancing of data with the new drives.

When you add multiple drives, it is more efficient to add them in a single AddDrives method call rather than multiple individual methods with a single drive each. This reduces the amount of data balancing that must occur to stabilize the storage load on the cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
drives	Information about each drive to be added to the cluster. Possible values: • driveID: The ID of the drive to add (integer). • type: The type of drive to add (string). Valid values are "slice", "block", or "volume". If omitted, the system assigns the correct type.	JSON object array	None	Yes (type is optional)

Return value

This method has the following return value:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "AddDrives",
  "params": {
    "drives": [
      {
        "driveID": 1,
        "type": "slice"
      },
      {
        "driveID": 2,
        "type": "block"
      },
      {
        "driveID": 3,
        "type": "block"
      }
    ]
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result" : {
    "asyncHandle": 1
  }
}
```

New since version

9.6

Find more information

- GetAsyncResult
- ListDrives
- ListSyncJobs

GetDriveHardwareInfo

You can use the `GetDriveHardwareInfo` method to get all the hardware information

for the given drive. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
driveID	ID of the drive for the request.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
result	Returned hardware information for the specified driveID.	hardwareInfo

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetDriveHardwareInfo",
   "params": {
     "driveID": 5
   },
   "id" : 100
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 100,
   "result" : {
     "driveHardwareInfo" : {
       "description" : "ATA Drive",
       "dev" : "8:80",
       "devpath" :
"/devices/pci0000:40/0000:40:01.0/0000:41:00.0/host6/port-6:0/expander-
6:0/port-6:0:4/end_device-6:0:4/target6:0:4/6:0:4:0/block/sdf",
       "driveSecurityAtMaximum" : false,
       "driveSecurityFrozen" : false
       "driveSecurityLocked" : false,
       "logicalname" : "/dev/sdf",
       "product" : "INTEL SSDSA2CW300G3",
       "securityFeatureEnabled" : false,
       "securityFeatureSupported" : true,
       "serial" : "CVPR121400NT300EGN",
       "size" : "300069052416",
       "uuid" : "7e1fd5b9-5acc-8991-e2ac-c48f813a3884",
       "version" : "4PC10362"
     }
   }
}
```


New since version

9.6

Find more information

ListDrives

GetDriveStats

You can use the `GetDriveStats` method to get high-level activity measurements for a single drive. Values are cumulative from the addition of the drive to the cluster. Some values are specific to block drives. Statistical data is returned for either block or metadata drive types when you run this method.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
driveID	ID of the drive for the request.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
driveStats	Drive activity information for the specified driveID.	driveStats

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetDriveStats",
   "params": {
     "driveID": 3
   },
   "id" : 1
}
```

Response example (block drive)

This method returns a response similar to the following example for a block drive:

```
{
  "id": 1,
  "result": {
    "driveStats": {
      "driveID": 10,
      "failedDieCount": 0,
      "lifeRemainingPercent": 99,
      "lifetimeReadBytes": 26471661830144,
      "lifetimeWriteBytes": 13863852441600,
      "powerOnHours": 33684,
      "readBytes": 10600432105,
      "readOps": 5101025,
      "reallocatedSectors": 0,
      "reserveCapacityPercent": 100,
      "timestamp": "2016-10-17T20:23:45.456834Z",
      "totalCapacity": 300069052416,
      "usedCapacity": 6112226545,
      "usedMemory": 114503680,
      "writeBytes": 53559500896,
      "writeOps": 25773919
    }
  }
}
```


Response example (volume metadata drive)

This method returns a response similar to the following example for a volume metadata drive:

```
{
  "id": 1,
  "result": {
    "driveStats": {
      "activeSessions": 8,
      "driveID": 12,
      "failedDieCount": 0,
      "lifeRemainingPercent": 100,
      "lifetimeReadBytes": 2308544921600,
      "lifetimeWriteBytes": 1120986464256,
      "powerOnHours": 16316,
      "readBytes": 1060152152064,
      "readOps": 258826209,
      "reallocatedSectors": 0,
      "reserveCapacityPercent": 100,
      "timestamp": "2016-10-17T20:34:52.456130Z",
      "totalCapacity": 134994670387,
      "usedCapacity": null,
      "usedMemory": 22173577216,
      "writeBytes": 353346510848,
      "writeOps": 86266238
    }
  }
}
```


New since version

9.6

Find more information

ListDrives

ListDrives

You can use the `ListDrives` method to list the drives that exist in the active nodes of the cluster. This method returns drives that have been added as volume metadata or block drives as well as drives that have not been added and are available.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
drives	List of drives in the cluster.	drive array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListDrives",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
     "drives" : [
       {
         "attributes" : {},
         "capacity" : 299917139968,
         "driveID" : 35,
         "nodeID" : 5,
         "serial" : "scsi-SATA_INTEL_SSDSA2CW6CVPR141502R3600FGN-part2",
         "slot" : 0,
         "status" : "active",
         "type" : "volume"
       },
       {
         "attributes" : {},
         "capacity" : 600127266816,
         "driveID" : 36,
         "nodeID" : 5,
         "serial" : "scsi-SATA_INTEL_SSDSA2CW6CVPR1415037R600FGN",
         "slot" : 6,
         "status" : "active",
         "type" : "block"
       }
     }
   ]
}
```


New since version

9.6

ListDriveStats

You can use the `ListDriveStats` method to list high-level activity measurements for multiple drives in the cluster. By default, this method returns statistics for all drives in the cluster, and these measurements are cumulative from the addition of the drive to the cluster. Some values this method returns are specific to block drives, and some are specific to metadata drives.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
drives	List of drive IDs (driveID) for which to return drive statistics. If you omit this parameter, measurements for all drives are returned.	integer array	None	No

Return values

This method has the following return values:

Name	Description	Type
driveStats	List of drive activity information for each drive.	driveStats array
errors	This list contains the driveID and associated error message. It is always present, and empty if there are no errors.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "ListDriveStats",
  "params": {
    "drives":[22,23]
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "driveStats": [
      {
        "driveID": 22,
        "failedDieCount": 0,
        "lifeRemainingPercent": 84,
        "lifetimeReadBytes": 30171004403712,
        "lifetimeWriteBytes": 103464755527680,
        "powerOnHours": 17736,
        "readBytes": 14656542,
         "readOps": 3624,
        "reallocatedSectors": 0,
        "reserveCapacityPercent": 100,
        "timestamp": "2016-03-01T00:19:24.782735Z",
        "totalCapacity": 300069052416,
        "usedCapacity": 1783735635,
        "usedMemory": 879165440,
        "writeBytes": 2462169894,
        "writeOps": 608802
      }
    ],
    "errors": [
      {
        "driveID": 23,
        "exception": {
          "message": "xStatCheckpointDoesNotExist",
          "name": "xStatCheckpointDoesNotExist"
        }
      }
    ]
  }
}
```


New since version

9.6

Find more information

GetDriveStats

RemoveDrives

You can use the `RemoveDrives` method to proactively remove drives that are part of the

cluster. You might use this method when reducing cluster capacity or preparing to replace drives nearing the end of their service life. `RemoveDrives` creates a third copy of the block data on the other nodes in the cluster and waits for syncing to complete before moving the drives to the "Available" list. Drives in the "Available" list are completely removed from the system and have no running services or active data.

`RemoveDrives` is an asynchronous method. Depending on the total capacity of the drives being removed, it might take several minutes to migrate all of the data.

When removing multiple drives, use a single `RemoveDrives` method call rather than multiple individual methods with a single drive each. This reduces the amount of data balancing that must occur to evenly stabilize the storage load on the cluster.

You can also remove drives with a "failed" status using `RemoveDrives`. When you remove a drive with a "failed" status, the drive is not returned to an "available" or “active” status. The drive is unavailable for use in the cluster.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
drives	List of driveIDs to remove from the cluster.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveDrives",
   "params": {
      "drives"  : [3, 4, 5]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result" : {
     "asyncHandle": 1
   }
}
```

New since version

9.6

Find more information

- GetAsyncResult
- ListDrives

SecureEraseDrives

You can use the `SecureEraseDrives` method to remove any residual data from drives that have a status of "available". You might use this method when replacing a drive nearing the end of its service life that contained sensitive data. This method uses a Security Erase Unit command to write a predetermined pattern to the drive and resets the encryption key on the drive. This asynchronous method might take several minutes to complete.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
drives	List of drive IDs to secure erase.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "SecureEraseDrives",
   "params": {
     "drives"  : [3, 4, 5]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1
   "result" : {
     "asyncHandle" : 1
   }
}
```

New since version

9.6

Find more information

- GetAsyncResult
- ListDrives

Fibre Channel API methods

You can use Fibre Channel API methods to add, modify, or remove Fibre Channel node members of a storage cluster.

- GetVolumeAccessGroupLunAssignments
- ListFibreChannelPortInfo
- ListFibreChannelSessions
- ListNodeFibreChannelPortInfo
- ModifyVolumeAccessGroupLunAssignments

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

GetVolumeAccessGroupLunAssignments

You can use the `GetVolumeAccessGroupLunAssignments` method to retrieve details on LUN mappings of a specified volume access group.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeAccessGroupID	A unique volume access group ID used to return information.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumeAccessGroupLunAssignments	A list of all physical Fibre Channel ports, or a port for a single node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetVolumeAccessGroupLunAssignments",
   "params": {
      "volumeAccessGroupID": 5
    },
  "id" : 1
   }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
    "volumeAccessGroupLunAssignments" : {
       "volumeAccessGroupID" : 5,
       "lunAssignments" : [
          {"volumeID" : 5, "lun" : 0},
          {"volumeID" : 6, "lun" : 1},
          {"volumeID" : 7, "lun" : 2},
          {"volumeID" : 8, "lun" : 3}
       ],
       "deletedLunAssignments" : [
           {"volumeID" : 44, "lun" : 44}
       ]
    }
  }
}
```

New since version

9.6

ListFibreChannelPortInfo

You can use the `ListFibreChannelPortInfo` method to list information about the Fibre Channel ports.

This API method is intended for use on individual nodes; a userid and password are required for access to individual Fibre Channel nodes. However, this method can be used on the cluster if the force parameter is set to true. When used on the cluster, all Fibre Channel interfaces are listed.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
force	Set to true to run on all nodes in the cluster.	boolean	None	No

Return value

This method has the following return value:

Name	Description	Type

fibreChannelPorts	A list of all physical Fibre Channel ports, or a port for a single node.	fibreChannelPort array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListFibreChannelPortInfo",
    "params": {},
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "fibreChannelPortInfo": {
      "5": {
        "result": {
          "fibreChannelPorts": [
            {
              "firmware": "7.04.00 (d0d5)",
              "hbaPort": 1,
              "model": "QLE2672",
              "nPortID": "0xc70084",
              "pciSlot": 3,
              "serial": "BFE1335E03500",
              "speed": "8 Gbit",
              "state": "Online",
              "switchWwn": "20:01:00:2a:6a:98:a3:41",
              "wwnn": "5f:47:ac:c8:3c:e4:95:00",
              "wwpn": "5f:47:ac:c0:3c:e4:95:0a"
            },
            {
              "firmware": "7.04.00 (d0d5)",
              "hbaPort": 2,
              "model": "QLE2672",
              "nPortID": "0x0600a4",
              "pciSlot": 3,
              "serial": "BFE1335E03500",
              "speed": "8 Gbit",
```

```
            "state": "Online",
            "switchWwn": "20:01:00:2a:6a:9c:71:01",
            "wwnn": "5f:47:ac:c8:3c:e4:95:00",
            "wwpn": "5f:47:ac:c0:3c:e4:95:0b"
          },
          {
            "firmware": "7.04.00 (d0d5)",
            "hbaPort": 1,
            "model": "QLE2672",
            "nPortID": "0xc70044",
            "pciSlot": 2,
            "serial": "BFE1335E04029",
            "speed": "8 Gbit",
            "state": "Online",
            "switchWwn": "20:01:00:2a:6a:98:a3:41",
            "wwnn": "5f:47:ac:c8:3c:e4:95:00",
            "wwpn": "5f:47:ac:c0:3c:e4:95:08"
          },
          {
            "firmware": "7.04.00 (d0d5)",
            "hbaPort": 2,
            "model": "QLE2672",
            "nPortID": "0x060044",
            "pciSlot": 2,
            "serial": "BFE1335E04029",
            "speed": "8 Gbit",
            "state": "Online",
            "switchWwn": "20:01:00:2a:6a:9c:71:01",
            "wwnn": "5f:47:ac:c8:3c:e4:95:00",
            "wwpn": "5f:47:ac:c0:3c:e4:95:09"
          }
        ]
      }
    },
    "6": {
      "result": {
          "fibreChannelPorts": [
        {
           "firmware": "7.04.00 (d0d5)",
           "hbaPort": 1,
           "model": "QLE2672",
           "nPortID": "0x060084",
           "pciSlot": 3,
           "serial": "BFE1335E04217",
           "speed": "8 Gbit",
           "state": "Online",
```

```
          "switchWwn": "20:01:00:2a:6a:9c:71:01",
          "wwnn": "5f:47:ac:c8:3c:e4:95:00",
          "wwpn": "5f:47:ac:c0:3c:e4:95:02"
        },
        {
          "firmware": "7.04.00 (d0d5)",
          "hbaPort": 2,
          "model": "QLE2672",
          "nPortID": "0xc700a4",
          "pciSlot": 3,
          "serial": "BFE1335E04217",
          "speed": "8 Gbit",
          "state": "Online",
          "switchWwn": "20:01:00:2a:6a:98:a3:41",
          "wwnn": "5f:47:ac:c8:3c:e4:95:00",
          "wwpn": "5f:47:ac:c0:3c:e4:95:03"
        },
        {
          "firmware": "7.04.00 (d0d5)",
          "hbaPort": 1,
          "model": "QLE2672",
          "nPortID": "0xc70064",
          "pciSlot": 2,
          "serial": "BFE1341E09515",
          "speed": "8 Gbit",
          "state": "Online",
          "switchWwn": "20:01:00:2a:6a:98:a3:41",
          "wwnn": "5f:47:ac:c8:3c:e4:95:00",
          "wwpn": "5f:47:ac:c0:3c:e4:95:00"
        },
        {
          "firmware": "7.04.00 (d0d5)",
          "hbaPort": 2,
          "model": "QLE2672",
          "nPortID": "0x060064",
          "pciSlot": 2,
          "serial": "BFE1341E09515",
          "speed": "8 Gbit",
          "state": "Online",
          "switchWwn": "20:01:00:2a:6a:9c:71:01",
          "wwnn": "5f:47:ac:c8:3c:e4:95:00",
          "wwpn": "5f:47:ac:c0:3c:e4:95:01"
        }
      ]
    }
  }
```

```
    }
  }
}
```

New since version

9.6

ListFibreChannelSessions

You can use the `ListFibreChannelSessions` method to list information about the Fibre Channel sessions on a cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
sessions	A list of objects describing active Fibre Channel sessions on the cluster.	session array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListFibreChannelSessions",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
     "sessions" : [
    {
       "initiatorWWPN" : "21:00:00:0e:1e:14:af:40",
       "nodeID" : 5,
       "serviceID" : 21,
       "targetWWPN": "5f:47:ac:c0:00:00:00:10",
       "volumeAccessGroupID": 7
    },
    {
       "initiatorWWPN" : "21:00:00:0e:1e:14:af:40",
       "nodeID" : 1,
       "serviceID" : 22,
       "targetWWPN": "5f:47:ac:c0:00:00:00:11",
       "volumeAccessGroupID": 7
    }
    ]
  }
}
```

New since version

9.6

ListNodeFibreChannelPortInfo

You can use the `ListNodeFibreChannelPortInfo` method to list information about the Fibre Channel ports on a node.

This API method is intended for use on individual nodes; a userid and password are required for access to individual Fibre Channel nodes. When used on the cluster, all Fibre Channel interfaces are listed.

Parameter

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
fibreChannelPorts	A list of all physical Fibre Channel ports, or a port for a single node.	fibreChannelPort array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListNodeFibreChannelPortInfo",
   "params": {
       "nodeID": 5,
       "force": true
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "fibreChannelPorts": [
      {
        "firmware": "7.04.00 (d0d5)",
        "hbaPort": 1,
        "model": "QLE2672",
        "nPortID": "0xc7002c",
        "pciSlot": 3,
        "serial": "BFE1335E03500",
        "speed": "8 Gbit",
        "state": "Online",
        "switchWwn": "20:01:00:2a:6a:98:a3:41",
        "wwnn": "5f:47:ac:c8:35:54:02:00",
        "wwpn": "5f:47:ac:c0:35:54:02:02"
      },
      {
         "firmware": "7.04.00 (d0d5)",
         "hbaPort": 2,
         "model": "QLE2672",
         "nPortID": "0x06002d",
         "pciSlot": 3,
         "serial": "BFE1335E03500",
         "speed": "8 Gbit",
         "state": "Online",
         "switchWwn": "20:01:00:2a:6a:9c:71:01",
         "wwnn": "5f:47:ac:c8:35:54:02:00",
         "wwpn": "5f:47:ac:c0:35:54:02:03"
```

```
      },
      {
         "firmware": "7.04.00 (d0d5)",
         "hbaPort": 1,
         "model": "QLE2672",
         "nPortID": "0xc7002a",
         "pciSlot": 2,
         "serial": "BFE1335E04029",
         "speed": "8 Gbit",
         "state": "Online",
         "switchWwn": "20:01:00:2a:6a:98:a3:41",
         "wwnn": "5f:47:ac:c8:35:54:02:00",
         "wwpn": "5f:47:ac:c0:35:54:02:00"
     },
     {
         "firmware": "7.04.00 (d0d5)",
         "hbaPort": 2,
         "model": "QLE2672",
         "nPortID": "0x06002a",
         "pciSlot": 2,
         "serial": "BFE1335E04029",
         "speed": "8 Gbit",
         "state": "Online",
         "switchWwn": "20:01:00:2a:6a:9c:71:01",
         "wwnn": "5f:47:ac:c8:35:54:02:00",
         "wwpn": "5f:47:ac:c0:35:54:02:01"
      }
    ]
  }
}
```

New since version

9.6

ModifyVolumeAccessGroupLunAssignments

You can use the `ModifyVolumeAccessGroupLunAssignments` method to define custom LUN assignments for specific volumes.

This method changes only LUN values set on the lunAssignments parameter in the volume access group. All other LUN assignments remain unchanged.

LUN assignment values must be unique for volumes in a volume access group. You cannot define duplicate LUN values within a volume access group. However, you can use the same LUN values again in different volume access groups.

Valid LUN values are 0 through 16383. The system generates an exception if you pass a LUN value outside of this range. None of the specified LUN assignments are modified if there is an exception.

CAUTION:

If you change a LUN assignment for a volume with active I/O, the I/O can be disrupted. You should change the server configuration before changing volume LUN assignments.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeAccessGroupID	Unique volume access group ID for which the LUN assignments will be modified.	integer	None	Yes
lunAssignments	The volume IDs with new assigned LUN values.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumeAccessGroupLunAssignments	An object containing details of the modified volume access group LUN assignments.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ModifyVolumeAccessGroupLunAssignments",
   "params": {
       "volumeAccessGroupID" : 218,
       "lunAssignments" : [
           {"volumeID" : 832, "lun" : 0},
           {"volumeID" : 834, "lun" : 1}
        ]
    },
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeAccessGroupLunAssignments": {
      "deletedLunAssignments": [],
      "lunAssignments": [
        {
          "lun": 0,
          "volumeID": 832
        },
        {
          "lun": 1,
          "volumeID": 834
        }
      ],
      "volumeAccessGroupID": 218
    }
  }
}
```

New since version

9.6

Initiator API methods

Initiator methods enable you to add, remove, view, and modify iSCSI initiator objects, which handle communication between the storage system and external storage clients.

- CreateInitiators
- DeleteInitiators
- ListInitiators
- ModifyInitiators

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

CreateInitiators

You can use `CreateInitiators` to create multiple new initiator IQNs or World Wide Port Names (WWPNs) and optionally assign them aliases and attributes. When you use `CreateInitiators` to create new initiators, you can also add them to volume access groups.

If the operation fails to create one of the initiators provided in the parameter, the method returns an error and does not create any initiators (no partial completion is possible).

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
initiators	A list of objects containing characteristics of each new initiator. Objects: • `alias`: (Optional) The friendly name to assign to this initiator. (string) • `attributes`: (Optional) A set of JSON attributes to assign to this initiator. (JSON object) • `chapUsername`: (Optional) The unique CHAP username for this initiator. Defaults to the initiator name (IQN) if not specified during creation and requireChap is true. (string) • `initiatorSecret`: (Optional) The CHAP secret used to authenticate the initiator. Defaults to a randomly generated secret if not specified during creation and requireChap is true. (string) • `name`: (Required) The name of the initiator (IQN or WWPN) to create. (string) • `requireChap`: (Optional) True if CHAP is required during session login for	JSON object array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
initiators	List of objects describing the newly created initiators.	initiator array

Error

This method can return the following error:

Name	Description
xInitiatorExists	Returned if the initiator name you chose already exists.

Request example

Requests for this method are similar to the following example:

```
{
  "id": 3291,
  "method": "CreateInitiators",
  "params": {
    "initiators": [
      {
        "name": "iqn.1993-08.org.debian:01:288170452",
        "alias": "example1"
      },
      {
        "name": "iqn.1993-08.org.debian:01:297817012",
        "alias": "example2"
      }
    ]
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 3291,
  "result": {
    "initiators": [
      {
        "alias": "example1",
        "attributes": {},
        "initiatorID": 145,
        "initiatorName": "iqn.1993-08.org.debian:01:288170452",
        "volumeAccessGroups": []
      },
      {
        "alias": "example2",
        "attributes": {},
        "initiatorID": 146,
        "initiatorName": "iqn.1993-08.org.debian:01:297817012",
        "volumeAccessGroups": []
      }
    ]
  }
}
```


New since version

9.6

Find more information

ListInitiators

DeleteInitiators

You can use `DeleteInitiators` to delete one or more initiators from the system (and from any associated volumes or volume access groups).

If `DeleteInitiators` fails to delete one of the initiators provided in the parameter, the system returns an error and does not delete any initiators (no partial completion is possible).

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
initiators	An array of IDs of initiators to delete.	integer array	None	Yes

Return values

This method has no return values.

Error

This method can return the following error:

Name	Description
xInitiatorDoesNotExist	Returned if the initiator name you choose does not exist.

Request example

Requests for this method are similar to the following example:

```
{
  "id": 5101,
  "method": "DeleteInitiators",
  "params": {
    "initiators": [
      145,
      147
    ]
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 5101,
  "result": {}
}
```

New since version

9.6

ListInitiators

You can use the `ListInitiators` method to get the list of initiator IQNs or World Wide Port Names (WWPNs).

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
initiators	A list of initiator IDs to retrieve. You can supply this parameter or the startInitiatorID parameter, but not both.	integer array	None	No
startInitiatorID	The initiator ID at which to begin the listing. You can supply this parameter or the initiators parameter, but not both.	integer	0	No
limit	The maximum number of initiator objects to return.	integer	(unlimited)	No

Return value

This method has the following return value:

Name	Description	Type
initiators	List of the initiator information.	initiator array

Exceptions

This method can have the following exception:

Name	Description
xInvalidParameter	Thrown if you include both the startInitiatorID and the initiators parameters in the same method call.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListInitiators",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "initiators": [
      {
        "alias": "",
        "attributes": {},
        "initiatorID": 2,
        "initiatorName": "iqn.1993-08.org.debian:01:c84ffd71216",
        "volumeAccessGroups": [
          1
        ]
      }
    ]
  }
}
```

New since version

9.6

ModifyInitiators

You can use the `ModifyInitiators` method to change the attributes of one or more existing initiators.

You cannot change the name of an existing initiator. If you need to change the name of an initiator, delete it first with the DeleteInitiators method and create a new one with the CreateInitiators method.

If ModifyInitiators fails to change one of the initiators provided in the parameter, the method returns an error and does not modify any initiators (no partial completion is possible).

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required

initiators	A list of objects containing characteristics of each initiator to modify. Possible objects: • `alias`: (Optional) A new friendly name to assign to the initiator. (string) • `attributes`: (Optional) A new set of JSON attributes to assign to the initiator. (JSON object) • `chapUsername`: (Optional) A new unique CHAP username for this initiator. (string) • `forceDuringUpgrade`: Complete the initiator modification during an upgrade. • `initiatorID`: (Required) The ID of the initiator to modify. (integer) • `initiatorSecret`: (Optional) A new CHAP secret used to authenticate the initiator. (string) • `requireChap`: (Optional) True if CHAP is required for this initiator. (boolean) • `targetSecret`: (Optional) A new CHAP secret used to authenticate the	JSON object array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
initiators	List of objects describing the newly modified initiators.	initiator array

Request example

Requests for this method are similar to the following example:

```
{
  "id": 6683,
  "method": "ModifyInitiators",
  "params": {
    "initiators": [
      {
        "initiatorID": 2,
        "alias": "alias1",
        "volumeAccessGroupID": null
      },
      {
        "initiatorID": 3,
        "alias": "alias2",
        "volumeAccessGroupID": 1
      }
    ]
  }
}
```

Response example

This method returns a response similar to the following example:

target (when using mutual CHAP authentication)

list of virtual network identifiers associated with this initiator. If

different volume access group, it is removed from the old volume access group. If this key is present but null, the initiator is removed from its current volume access group, but not placed in any new volume access group. (integer)

```
{
  "id": 6683,
  "result": {
    "initiators": [
      {
        "alias": "alias1",
        "attributes": {},
        "initiatorID": 2,
        "initiatorName": "iqn.1993-08.org.debian:01:395543635",
        "volumeAccessGroups": []
      },
      {
        "alias": "alias2",
        "attributes": {},
        "initiatorID": 3,
        "initiatorName": "iqn.1993-08.org.debian:01:935573135",
        "volumeAccessGroups": [
          1
        ]
      }
    ]
  }
}
```


New since version

9.6

Find more information

- CreateInitiators
- DeleteInitiators

LDAP API methods

You can use the Lightweight Directory Access Protocol (LDAP) to authenticate access to Element storage. The LDAP API methods described in this section enable you to configure LDAP access to the storage cluster.

- AddLdapClusterAdmin
- EnableLdapAuthentication
- DisableLdapAuthentication
- GetLdapConfiguration
- TestLdapAuthentication

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddLdapClusterAdmin

You can use the `AddLdapClusterAdmin` to add a new LDAP cluster administrator user. An LDAP cluster administrator can manage the cluster using the API and management tools. LDAP cluster admin accounts are completely separate and unrelated to standard tenant accounts.

Parameters

You can also use this method to add an LDAP group that has been defined in Active Directory®. The access level that is given to the group is passed to the individual users in the LDAP group.

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	Controls which methods this cluster admin can use.	string array	None	Yes
acceptEula	Accept the End User License Agreement. Set to true to add a cluster administrator account to the system. If omitted or set to false, the method call fails.	boolean	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
username	The distinguished user name for the new LDAP cluster admin.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "AddLdapClusterAdmin",
  "params": {"username":"cn=mike
jones,ou=ptusers,dc=prodtest,dc=solidfire,dc=net",
    "access": ["administrator", "read"
    ]
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id": 1,
 "result": {}
}
```

New since version

9.6

Find more information

Access control

EnableLdapAuthentication

You can use the `EnableLdapAuthentication` method to configure an LDAP directory connection for LDAP authentication to a cluster. Users that are members of the LDAP directory can then log in to the storage system using their LDAP credentials.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
authType	Identifies which user authentication method to use. Possible values: • `DirectBind` • `SearchAndBind`	string	SearchAndBind	No
groupSearchBaseDN	The base DN of the tree to start the group subtree search.	string	None	No
groupSearchType	Controls the default group search filter used. Possible values: • NoGroups: No group support. • ActiveDirectory: Nested membership of all of a user's active directory groups. • MemberDN: MemberDN style groups (single level).	string	ActiveDirectory	No
serverURIs	A comma-separated list of LDAP or LDAPS server URIs. You can add a custom port to the end of an LDAP or LDAPS URI by using a colon followed by the port number. For example, the URI "ldap://1.2.3.4" uses the default port and the URI "ldaps://1.2.3.4:123" uses the custom port 123.	string array	None	Yes

Name	Description	Type	Default value	Required
userSearchBaseDN	The base DN of the tree to start the subtree search. This parameter is required when using an authType of SearchAndBind.	string	None	No
searchBindDN	A fully qualified DN to log in with to perform an LDAP search for the user. The DN requires read access to the LDAP directory. This parameter is required when using an authType of SearchAndBind.	string	None	Yes
searchBindPassword	The password for the searchBindDN account used for searching. This parameter is required when using an authType of SearchAndBind.	string	None	Yes

Name	Description	Type	Default value	Required
userSearchFilter	The LDAP search filter to use when querying the LDAP server. The string should have the placeholder text "%USERNAME%" which is replaced with the username of the authenticating user. For example, (&(objectClass=person)(sAMAccountName=%USERNAME%)) will use the sAMAccountName field in Active Directory to match the username entered at cluster login. This parameter is required when using an authType of SearchAndBind.	string	None	Yes
userDNTemplate	A string template used to define a pattern for constructing a full user distinguished name (DN). The string should have the placeholder text "%USERNAME%" which is replaced with the username of the authenticating user. This parameter is required when using an authType of DirectBind.	string	None	Yes

Name	Description	Type	Default value	Required
groupSearchCustomFilter	For use with the CustomFilter search type, an LDAP filter to use to return the DNs of a user's groups. The string can have placeholder text of %USERNAME% and %USERDN% to be replaced with their username and full userDN as needed.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableLdapAuthentication",
  "params": {
     "authType": "SearchAndBind",
     "groupSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
     "groupSearchType": "ActiveDirectory",
     "searchBindDN": "SFReadOnly@prodtest.solidfire.net",
     "searchBindPassword": "zsw@#edcASD12",
     "sslCert": "",
     "userSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
     "userSearchFilter":
"(&(objectClass=person)(sAMAccountName=%USERNAME%))",
     "serverURIs":[
           "ldaps://111.22.333.444",
           "ldap://555.66.777.888"
           ]
       },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
"result": {
  }
}
```

New since version

9.6

DisableLdapAuthentication

You can use the `DisableLdapAuthentication` method to disable LDAP authentication and remove all LDAP configuration settings. This method does not remove any configured cluster admin accounts for users or groups. After LDAP authentication has been disabled, cluster admins that are configured to use LDAP authentication can no longer access the cluster.

Parameters

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableLdapAuthentication",
  "params": {},
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
"result": {}
}
```

New since version

9.6

GetLdapConfiguration

You can use the `GetLdapConfiguration` method to get the currently active LDAP configuration on the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value.

Name	Description	Type
ldapConfiguration	List of the current LDAP configuration settings. This API call does not return the plain text of the search account password. **Note:** If LDAP authentication is currently disabled, all the returned settings are empty with the exception of "authType", and "groupSearchType" which are set to "SearchAndBind" and "ActiveDirectory" respectively.	ldapConfiguration

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetLdapConfiguration",
  "params": {},
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "ldapConfiguration": {
        "authType": "SearchAndBind",
        "enabled": true,
        "groupSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
        "groupSearchCustomFilter": "",
        "groupSearchType": "ActiveDirectory",
        "searchBindDN": "SFReadOnly@prodtest.solidfire.net",
        "serverURIs": [
           "ldaps://111.22.333.444",
           "ldap://555.66.777.888"
            ],
        "userDNTemplate": "",
        "userSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
        "userSearchFilter":
"(&(objectClass=person)(sAMAccountName=%USERNAME%))"
     }
   }
}
```

New since version

9.6

TestLdapAuthentication

You can use the `TestLdapAuthentication` method to validate the currently enabled LDAP authentication settings. If the configuration is correct, the API call returns the group membership of the tested user.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
username	The username to be tested.	string	None	Yes
password	The password for the username to be tested.	string	None	Yes

Name	Description	Type	Default value	Required
ldapConfiguration	An ldapConfiguration object to be tested. If you provide this parameter, the system tests the provided configuration even if LDAP authentication is currently disabled.	ldapConfiguration	None	No

Return values

This method has the following return values:

Name	Description	Type
groups	List of LDAP groups that include the tested user as a member.	array
userDN	The tested user's full LDAP distinguished name.	string

Request example

Requests for this method are similar to the following example:

```
{
"method": "TestLdapAuthentication",
"params": {"username":"admin1",
           "password":"admin1PASS"
      },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id": 1,
 "result": {
    "groups": [
        "CN=StorageMgmt,OU=PTUsers,DC=prodtest,DC=solidfire,DC=net"
      ],
        "userDN": "CN=Admin1
Jones,OU=PTUsers,DC=prodtest,DC=solidfire,DC=net"
  }
}
```

New since version

9.6

Multi-factor authentication API methods

You can use multi-factor authentication (MFA) to manage user sessions using a third-party Identity Provider (IdP) via the Security Assertion Markup Language (SAML).

- AddIdpClusterAdmin
- CreateIdpConfiguration
- DeleteAuthSession
- DeleteAuthSessionsByClusterAdmin
- DeleteAuthSessionsByUsername
- DeleteIdpConfiguration
- DisableIdpAuthentication
- EnableIdpAuthentication
- GetIdpAuthenticationState
- ListActiveAuthSessions
- ListIdpConfigurations
- UpdateIdpConfiguration

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddIdpClusterAdmin

You can use the `AddIpdClusterAdmin` method to add a cluster administrator user authenticated by a third-party Identity Provider (IdP). IdP cluster admin accounts are configured based on SAML attribute-value information provided within the IdP's SAML

assertion associated with the user. If a user successfully authenticates with the IdP and has SAML attribute statements within the SAML assertion matching multiple IdP cluster admin accounts, the user will have the combined access level of those matching IdP cluster admin accounts.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	Controls which methods this IdP cluster admin can use.	string array	None	Yes
acceptEula	Accept the End User License Agreement. Set to true to add a cluster administrator account to the system. If omitted or set to false, the method call fails.	boolean	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
username	A SAML attribute-value mapping to an IdP cluster admin (for example, email=test@example.com). This can be defined using a specific SAML subject using `NameID` or as an entry in the SAML attribute statement, such as `eduPersonAffiliation`.	string	None	Yes

Return values

This method has the following return value:

Name	Description	Type

clusterAdminID	Unique identifier for the newly created cluster admin.	integer

Request example

Requests for this method are similar to the following example:

```
{
    "method": "AddIdpClusterAdmin",
    "params": {
        "username": "email=test@example.com",
        "acceptEula": true,
        "access": ["administrator"]
    }
}
```

Response example

This method returns a response similar to the following example:

```
{
    "result": {
        "clusterAdminID": 13
    }
}
```

New since version

12.0

CreateIdpConfiguration

You can use the `CreateIpdConfiguration` method to create a potential trust relationship for authentication using a third-party Identity Provider (IdP) for the cluster. A SAML Service Provider certificate is required for IdP communication. This certificate is generated as required, and returned by this API call.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
idpMetadata	IdP metadata to store.	string	None	Yes

Name	Description	Type	Default value	Required
idpName	Name used to identify an IdP provider for SAML 2.0 single sign-on.	string	None	Yes

Return values

This method has the following return value:

Name	Description	Type
idpConfigInfo	Information about the third-party Identity Provider (IdP) configuration.	idpConfigInfo

Request example

Requests for this method are similar to the following example:

```
{
    "method": "CreateIdpConfiguration",
    "params": {
        "idpMetadata": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>
            <EntityDescriptor
              xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata\"
              xmlns:ds=\"http://www.w3.org/2000/09/xmldsig#\"
              xmlns:shibmd=\"urn:mace:shibboleth:metadata:1.0\"
              xmlns:xml=\"http://www.w3.org/XML/1998/namespace\"
              ...</Organization>
            </EntityDescriptor>",
        "idpName": "https://provider.name.url.com"
    },
}
```

Response example

This method returns a response similar to the following example:

```
{
    "result": {
        "idpConfigInfo": {
        "enabled": false,
        "idpConfigurationID": "f983c602-12f9-4c67-b214-bf505185cfed",
        "idpMetadata": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n
        <EntityDescriptor
xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata\"\r\n
        xmlns:ds=\"http://www.w3.org/2000/09/xmldsig#\"\r\n
        xmlns:shibmd=\"urn:mace:shibboleth:metadata:1.0\"\r\n
        xmlns:xml=\"http://www.w3.org/XML/1998/namespace\"\r\n
        ... </Organization>\r\n
        </EntityDescriptor>",
        "idpName": "https://privider.name.url.com",
        "serviceProviderCertificate": "-----BEGIN CERTIFICATE-----\n
        MIID...SlBHi\n
        -----END CERTIFICATE-----\n",
        "spMetadataUrl": "https://10.193.100.100/auth/ui/saml2"
        }
    }
}
```


New since version

12.0

DeleteAuthSession

You can use the `DeleteAuthSession` method to delete an individual user authentication session. If the calling user is not in the ClusterAdmins / Administrator AccessGroup, only the authentication session belonging to the calling user can be deleted.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
sessionID	Unique identifier for the auth session to be deleted.	UUID	None	Yes

Return values

This method has the following return value:

Name	Description	Type
session	Session information for the delete auth session.	authSessionInfo

Request example

Requests for this method are similar to the following example:

```
{
    "method": "DeleteAuthSession",
    "params": {
        "sessionID": "a862a8bb-2c5b-4774-a592-2148e2304713"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "session": {
            "accessGroupList": [
                "administrator"
            ],
            "authMethod": "Cluster",
            "clusterAdminIDs": [
                1
            ],
            "finalTimeout": "2020-04-09T17:51:30Z",
            "idpConfigVersion": 0,
            "lastAccessTimeout": "2020-04-06T18:21:33Z",
            "sessionCreationTime": "2020-04-06T17:51:30Z",
            "sessionID": "a862a8bb-2c5b-4774-a592-2148e2304713",
            "username": "admin"
        }
    }
}
```

New since version

12.0

DeleteAuthSessionsByClusterAdmin

You can use the `DeleteAuthSessionsByClusterAdmin` method to delete all authentication sessions associated with the specified `ClusterAdminID`. If the specified ClusterAdminID maps to a group of users, all authentication sessions for all members of that group will be deleted. To view a list of sessions for possible deletion, use the ListAuthSessionsByClusterAdmin method with the `ClusterAdminID` parameter.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
clusterAdminID	Unique identifier for the cluster admin.	integer	None	Yes

Return values

This method has the following return value:

Name	Description	Type
sessions	Session information for the deleted authentication sessions.	authSessionInfo

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DeleteAuthSessionsByClusterAdmin",
  "params": {
    "clusterAdminID": 1
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
"sessions": [
      {
        "accessGroupList": [
          "administrator"
        ],
        "authMethod": "Cluster",
        "clusterAdminIDs": [
          1
        ],
        "finalTimeout": "2020-03-14T19:21:24Z",
        "idpConfigVersion": 0,
        "lastAccessTimeout": "2020-03-11T19:51:24Z",
        "sessionCreationTime": "2020-03-11T19:21:24Z",
        "sessionID": "b12bfc64-f233-44df-8b9f-6fb6c011abf7",
        "username": "admin"
      }
    ]
}
```


New since version

12.0

DeleteAuthSessionsByUsername

You can use the `DeleteAuthSessionsByUsername` method to delete all authentication sessions for a given user(s). A caller not in AccessGroup ClusterAdmins/Administrator can only delete their own sessions. A caller with ClusterAdmins/Administrator privileges can delete sessions belonging to any user. To see the list of sessions that could be deleted, use `ListAuthSessionsByUsername` with the same parameters. To view a list of sessions for possible deletion, use the `ListAuthSessionsByUsername` method with the same parameter.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
authMethod	Authentication method of the user sessions to be deleted. Only a caller in the ClusterAdmins/Administrator AccessGroup can provide this parameter. Possible values are: • **authMethod=Cluster** specifies the ClusterAdmin username. • **authMethod=Ldap** specifies the user's LDAP DN. • **authMethod=Idp** specifies either the user's IdP UUID or NameID. If the IdP is not configured to return either option, this specifies a random UUID issued when the session was created.	authMethod	None	No
username	Unique identifier for the user.	string	None	No

Return values

This method has the following return value:

Name	Description	Type
sessions	Session information for the deleted authentication sessions.	authSessionInfo

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DeleteAuthSessionsByUsername",
  "params": {
    "authMethod": "Cluster",
    "username": "admin"
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
"sessions": [
      {
        "accessGroupList": [
          "administrator"
        ],
        "authMethod": "Cluster",
        "clusterAdminIDs": [
          1
        ],
        "finalTimeout": "2020-03-14T19:21:24Z",
        "idpConfigVersion": 0,
        "lastAccessTimeout": "2020-03-11T19:51:24Z",
        "sessionCreationTime": "2020-03-11T19:21:24Z",
        "sessionID": "b12bfc64-f233-44df-8b9f-6fb6c011abf7",
        "username": "admin"
      }
    ]
}
```

New since version

12.0

DeleteIdpConfiguration

You can use the `DeleteIdpConfiguration` method to delete an existing configuration of a third-party IdP for the cluster. Deleting the last IdP configuration removes the SAML Service Provider certificate from the cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
idpConfigurationID	UUID for the third-party IdP configuration.	UUID	None	No
idpName	Name used to identify and retrieve an IdP provider for SAML 2.0 single sign-on.	string	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DeleteIdpConfiguration",
  "params": {
    "idpConfigurationID": "f983c602-12f9-4c67-b214-bf505185cfed",
    "idpName": "https://provider.name.url.com"
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "result":{}
}
```

New since version

12.0

DisableIdpAuthentication

You can use the `DisableIdpAuthentication` method to disable support for

authentication using third-party IdPs for the cluster. Once disabled, users authenticated by third party IdPs are no longer able to access the cluster and any active authenticated sessions are invalidated/disconnected. LDAP and cluster admins are able to access the cluster via supported UIs.

Parameters

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableIdpAuthentication",
  "params": {}
}
```

Response example

This method returns a response similar to the following example:

```
{
"result": {}
}
```

New since version

12.0

EnableIdpAuthentication

You can use the `EnableIdpAuthentication` method to enable support for authentication using third-party IdPs for the cluster. Once IdP authentication is enabled, LDAP and cluster admins are no longer able to access the cluster via supported UIs and any active authenticated sessions are invalidated/disconnected. Only users authenticated by third party IdPs are able to access the cluster via supported UIs.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
idpConfigurationID	UUID for the third-party IdP configuration. If only one IdP configuration exists, then the default is to enable that configuration. If you have only a single IdpConfiguration, you need not provide the idpConfigurationID parameter.	UUID	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableIdpAuthentication",
  "params": {
    "idpConfigurationID": "f983c602-12f9-4c67-b214-bf505185cfed",
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
"result": {}
}
```

New since version

12.0

GetIdpAuthenticationState

You can use the `GetIdpAuthenticationState` method to return information regarding the state of authentication using third-party IdPs.

Parameters

This method has no input parameters.

Return values

This method has the following return value:

Name	Description	Type
enabled	Indicates whether third-party IdP authentication is enabled.	boolean

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetIdpAuthenticationState"
}
```

Response example

This method returns a response similar to the following example:

```
{
  "result": {"enabled": true}
}
```

New since version

12.0

ListActiveAuthSessions

You can use the `ListActiveAuthSessions` method to list all of the active authenticated sessions. Only users with Administrative access rights can call this method.

Parameters

This method has no input parameters.

Return values

This method has the following return value:

Name	Description	Type
sessions	Session information for the authentication sessions.	authSessionInfo

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListActiveAuthSessions"
}
```

Response example

This method returns a response similar to the following example:

```
{
"sessions": [
      {
        "accessGroupList": [
          "administrator"
        ],
        "authMethod": "Cluster",
        "clusterAdminIDs": [
          1
        ],
        "finalTimeout": "2020-03-14T19:21:24Z",
        "idpConfigVersion": 0,
        "lastAccessTimeout": "2020-03-11T19:51:24Z",
        "sessionCreationTime": "2020-03-11T19:21:24Z",
        "sessionID": "b12bfc64-f233-44df-8b9f-6fb6c011abf7",
        "username": "admin"
      }
    ]
}
```

New since version

12.0

ListIdpConfigurations

You can use the `ListIdpConfigurations` method to list configurations for third-party IdPs. Optionally, you can provide either the `enabledOnly` flag to retrieve the currently

enabled IdP configuration or an IdP metadata UUID or IdP name to query information for a specific IdP configuration.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
enabledOnly	Filters the result to return the currently enabled IdP configuration.	boolean	None	No
idpConfigurationID	UUID for the third-party IdP configuration.	UUID	None	No
idpName	Retrieves IdP configuration information for a specific IdP name.	string	None	No

Return values

This method has the following return value:

Name	Description	Type
idpConfigInfos	Information on the third-party IdP configuration(s).	idpConfigInfo array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListIdpConfigurations",
  "params": {}
}
```

Response example

This method returns a response similar to the following example:

```
{
    "result": {
        "idpConfigInfo": {
        "enabled": true,
        "idpConfigurationID": "f983c602-12f9-4c67-b214-bf505185cfed",
        "idpMetadata": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n
        <EntityDescriptor
xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata\"\r\n
        xmlns:ds=\"http://www.w3.org/2000/09/xmldsig#\"\r\n
        xmlns:shibmd=\"urn:mace:shibboleth:metadata:1.0\"\r\n
        xmlns:xml=\"http://www.w3.org/XML/1998/namespace\"\r\n
        ...</Organization>\r\n
        </EntityDescriptor>",
        "idpName": "https://privider.name.url.com",
        "serviceProviderCertificate": "-----BEGIN CERTIFICATE-----\n
        MI...BHi\n
        -----END CERTIFICATE-----\n",
        "spMetadataUrl": "https://10.193.100.100/auth/ui/saml2"
        }
    }
}
```


New since version

12.0

UpdateIdpConfiguration

You can use the `UpdateIdpConfiguration` method to update an existing configuration with a third-party IdP for the cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
generateNewCertificate	When specified as true, a new SAML key and certificate is generated and replaces the existing pair. Note: Replacing the existing certificate will disrupt the established trust between the cluster and the IdP until the cluster's Service Provider metadata is reloaded at the IdP. If not provided or set to false, the SAML certificate and key remains unchanged.	boolean	None	No
idpConfigurationID	UUID for the third-party IdP configuration.	UUID	None	No
idpMetadata	IdP metadata for configuration and integration details for SAML 2.0 single sign-on.	string	None	No
idpName	Name used to identify and retrieve an IdP provider for SAML 2.0 single sign-on.	string	None	No
newIdpName	If specified, this name replaces the old IdP name.	string	None	No

Return values

This method has the following return value:

Name	Description	Type
idpConfigInfo	Information around the third-party IdP configuration.	idpConfigInfo

Request example

Requests for this method are similar to the following example:

```
{
"method": "UpdateIdpConfiguration",
"params": {
        "idpConfigurationID": "f983c602-12f9-4c67-b214-bf505185cfed",
        "generateNewCertificate": true
    }
}
```

Response example

This method returns a response similar to the following example:

```
{
    "result": {
        "idpConfigInfo": {
        "enabled": true,
        "idpConfigurationID": "f983c602-12f9-4c67-b214-bf505185cfed",
        "idpMetadata": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\r\n
        <EntityDescriptor
xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata\"\r\n
        xmlns:ds=\"http://www.w3.org/2000/09/xmldsig#\"\r\n
        xmlns:shibmd=\"urn:mace:shibboleth:metadata:1.0\"\r\n
        xmlns:xml=\"http://www.w3.org/XML/1998/namespace\"\r\n
        ...</Organization>\r\n
        </EntityDescriptor>",
        "idpName": "https://privider.name.url.com",
        "serviceProviderCertificate": "-----BEGIN CERTIFICATE-----\n
        MI...BHi\n
        -----END CERTIFICATE-----\n",
        "spMetadataUrl": "https://10.193.100.100/auth/ui/saml2"
        }
    }
}
```

New since version

12.0

Session authentication API methods

You can use session-based authentication to manage user sessions.

- ListAuthSessionsByClusterAdmin
- ListAuthSessionsByUsername

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

ListAuthSessionsByClusterAdmin

You can use the `ListAuthSessionsByClusterAdmin` method to list all auth sessions associated with the specified `ClusterAdminID`. If the specified `ClusterAdminID` maps to a group of users, all auth sessions for all members of that group will be listed.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
clusterAdminID	Unique identifier for the cluster admin.	integer	None	Yes

Return values

This method has the following return value:

Name	Description	Type
sessions	List of session information for the auth sessions.	authSessionInfo

Request example

Requests for this method are similar to the following example:

```
{
    "method": "ListAuthSessionsByClusterAdmin",
    "clusterAdminID": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"sessions": [
      {
        "accessGroupList": [
          "administrator"
        ],
        "authMethod": "Cluster",
        "clusterAdminIDs": [
          1
        ],
        "finalTimeout": "2020-03-14T19:21:24Z",
        "idpConfigVersion": 0,
        "lastAccessTimeout": "2020-03-11T19:51:24Z",
        "sessionCreationTime": "2020-03-11T19:21:24Z",
        "sessionID": "b12bfc64-f233-44df-8b9f-6fb6c011abf7",
        "username": "admin"
      }
    ]
}
```


New since version

12.0

ListAuthSessionsByUsername

You can use the `ListAuthSessionsByUsername` method to list all auth sessions for the specified user. A caller not in AccessGroup ClusterAdmins / Administrator privileges may only list their own sessions. A caller with ClusterAdmins / Administrator privileges may list sessions belonging to any user.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
authMethod	Authentication method of the user sessions to be listed. Only a caller in the ClusterAdmins/Administrator AccessGroup can provide this parameter. Possible values are: • **authMethod=Cluster** specifies the ClusterAdmin username. • **authMethod=Ldap** specifies the user's LDAP DN. • **authMethod=Idp** specifies either the user's IdP UUID or NameID. If the IdP is not configured to return either option, this specifies a random UUID issued when the session was created.	authMethod	None	Yes
username	Unique identifier for the user.	string	None	Yes

Return values

This method has the following return value:

Name	Description	Type
sessions	List of session information for the auth sessions.	authSessionInfo

Request example

Requests for this method are similar to the following example:

```
{
    "method": "ListAuthSessionsByUsername",
    "authMethod": "Cluster",
    "username": "admin"
}
```

Response example

This method returns a response similar to the following example:

```
{
"sessions": [
      {
        "accessGroupList": [
          "administrator"
        ],
        "authMethod": "Cluster",
        "clusterAdminIDs": [
          1
        ],
        "finalTimeout": "2020-03-14T19:21:24Z",
        "idpConfigVersion": 0,
        "lastAccessTimeout": "2020-03-11T19:51:24Z",
        "sessionCreationTime": "2020-03-11T19:21:24Z",
        "sessionID": "b12bfc64-f233-44df-8b9f-6fb6c011abf7",
        "username": "admin"
      }
    ]
}
```

New since version

12.0

Node API methods

You can use node API methods to configure individual nodes. These methods operate on single nodes that need to be configured, are configured but not yet participating in a cluster, or are actively participating in a cluster. Node API methods enable you to view and modify settings for individual nodes and the cluster network used to communicate with the node. You must run these methods against individual nodes; you cannot run per-

node API methods against the address of the cluster.

- CheckPingOnVlan
- CheckProposedNodeAdditions
- CreateClusterSupportBundle
- CreateSupportBundle
- DeleteAllSupportBundles
- DisableMaintenanceMode
- DisableSsh
- EnableMaintenanceMode
- EnableSsh
- GetClusterConfig
- GetClusterState
- GetConfig
- GetDriveConfig
- GetHardwareConfig
- GetHardwareInfo
- GetIpmiConfig
- GetIpmiInfo
- GetNetworkConfig
- GetNetworkInterface
- GetNodeActiveTlsCiphers
- GetNodeFipsDrivesReport
- GetNodeSSLCertificate
- GetNodeSupportedTlsCiphers
- GetPendingOperation
- GetSshInfo
- ListDriveHardware
- ListNetworkInterfaces
- ListTests
- ListUtilities
- RemoveNodeSSLCertificate
- ResetDrives
- ResetNode
- ResetNodeSupplementalTlsCiphers
- RestartNetworking
- RestartServices
- SetClusterConfig

- SetConfig
- SetNetworkConfig
- SetNodeSSLCertificate
- SetNodeSupplementalTlsCiphers
- Shutdown
- TestConnectEnsemble
- TestConnectMvip
- TestConnectSvip
- TestDrives
- TestHardwareConfig
- TestLocateCluster
- TestLocalConnectivity
- TestNetworkConfig
- TestPing
- TestRemoteConnectivity

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

CheckPingOnVlan

You can use the `CheckPingOnVlan` method to test network connectivity on a temporary VLAN when performing pre-deployment network validation. `CheckPingOnVlan` creates a temporary VLAN interface, sends ICMP packets to all nodes in the storage cluster using the VLAN interface, and then removes the interface.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
attempts	Specifies the number of times the system should repeat the test ping.	integer	5	No
hosts	Specifies a comma-separated list of addresses or hostnames of devices to ping.	string	The nodes in the cluster	No

Name	Description	Type	Default value	Required
interface	The existing (base) interface from which the pings should be sent. Possible values: • Bond10G: Send pings from the Bond10G interface. • Bond1G: Send pings from the Bond1G interface.	string	None	Yes
packetSize	Specifies the number of bytes to send in the ICMP packet that is sent to each IP. The number of bytes must be less than the maximum MTU specified in the network configuration.	integer	None	No
pingTimeoutMsec	Specifies the number of milliseconds to wait for each individual ping response.	integer	500 ms	No
prohibitFragmentation	Enables the DF (Do not Fragment) flag for the ICMP packets.	boolean	false	No
sourceAddressV4	The source IPv4 address to use in the ICMP ping packets.	string	None	Yes
sourceAddressV6	The source IPv6 address to use in the ICMP ping packets.	string	None	Yes

Name	Description	Type	Default value	Required
totalTimeoutSec	Specifies the time in seconds the ping should wait for a system response before issuing the next ping attempt or ending the process.	integer	5	No
virtualNetworkTag	The VLAN ID to use when sending the ping packets.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
result	List of each IP the node was able to communicate with and ping response statistics.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
    "method": "CheckPingOnVlan",
    "params": {
        "interface": "Bond10G",
        "virtualNetworkTag": 4001,
        "sourceAddressV4": "192.168.41.4",
        "hosts": "192.168.41.2"
},
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "192.168.41.2": {
      "individualResponseCodes": [
        "Success",
        "Success",
        "Success",
        "Success",
        "Success"
      ],
      "individualResponseTimes": [
        "00:00:00.000373",
        "00:00:00.000098",
        "00:00:00.000097",
        "00:00:00.000074",
        "00:00:00.000075"
      ],
      "individualStatus": [
        true,
        true,
        true,
        true,
        true
      ],
      "interface": "Bond10G",
      "responseTime": "00:00:00.000143",
      "sourceAddressV4": "192.168.41.4",
      "successful": true,
      "virtualNetworkTag": 4001
    }
  }
}
```


New since version

11.1

CheckProposedNodeAdditions

You can use the `CheckProposedNodeAdditions` method to test a set of storage nodes to see if you can add them to a storage cluster without errors or best practice violations.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
nodes	A list of storage IP addresses of storage nodes that are ready to be added to a storage cluster.	string array	None	Yes

Return values

This method has the following return values:

Name	Description	Type
proposedClusterValid	Indicates whether or not the proposed storage nodes would make up a valid storage cluster. Possible values: • true • false	boolean

proposedClusterErrors	Errors that would occur if a storage cluster was created using the proposed storage nodes. Possible error codes: • `nodesNoCapacity`: Nodes did not have any useable capacity. • `nodesTooLarge`: Nodes constitute too large a portion of cluster capacity for the active protection scheme. • `nodesConnectFailed`: Could not connect to nodes to query hardware configuration. • `nodesQueryFailed`: Could not query nodes for hardware configuration. • `nodesClusterMember`: IP addresses for nodes are already in use in the cluster. • `nonFipsNodeCapable`: Unable to add a non-FIPS capable node to the storage cluster while the FIPS 140-2 drive encryption feature is enabled. • `nonFipsDrivesCapable`: Unable to add a node with non-FIPS-capable drives to the cluster while the FIPS 140-2 drive encryption feature is enabled.	string array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "CheckProposedNodeAdditions",
    "params": {
    "nodes": [
        "192.168.1.11",
        "192.168.1.12",
        "192.168.1.13",
        "192.168.1.14"
    ]
},
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "proposedClusterValid": true,
        "proposedClusterErrors": [ ]
    }
}
```

New since version

11.0

CreateClusterSupportBundle

You can use the `CreateClusterSupportBundle` on the management node to gather support bundles from all nodes in a cluster. The individual node support bundles are compressed as tar.gz files. The cluster support bundle is a tar file containing the node support bundles. You can only run this method on a management node; it does not work when run on a storage node.

Parameters

You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has the following input parameters:

Name	Description	Type	Default value	Required
allowIncomplete	Allows the script to continue to run if bundles cannot be gathered from one or more of the nodes.	boolean	None	No
bundleName	Unique name for each support bundle created. If no name is provided, then "supportbundle" and the node name are used as the file name	string	None	No
mvip	The MVIP of the cluster. Bundles are gathered from all nodes in the cluster. This parameter is required if the nodes parameter is not specified.	string	None	Yes
nodes	The IP addresses of the nodes from which to gather bundles. Use either nodes or mvip, but not both, to specify the nodes from which to gather bundles. This parameter is required if mvip is not specified.	string array	None	Yes
password	The cluster admin password. **Note:** This password is visible as text when entered.	string	None	Yes
username	The cluster admin user name.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
"method": "CreateClusterSupportBundle",
"params": {
          "bundlename": "clusterbundle",
          "mvip": "132.119.120.100"
     }
   },
 "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id":1,
  "result":{
   "details":{
      "bundleName":"clusterbundle",
      "extraArgs":"",
      "files":[
          "/tmp/supportbundles/clusterbundle.cl-4SD5.tar"
      ],
      "output":"timeout -s KILL 1790s
/usr/local/bin/sfclustersupportbundle --quiet --name=\"clusterbundle\"
--target-directory=\"/tmp/solidfire-dtemp.MM7f0m\" --user=\"admin\"
--pass=\"admin\" --mvip=132.119.120.100"
       },
       "duration":"00:00:24.938127",
       "result":"Passed"
    }
}
```

New since version

9.6

CreateSupportBundle

You can use `CreateSupportBundle` to create a support bundle file under the node's directory. After creation, the bundle is stored on the node as a tar file (gz compression option is available via the extraArgs parameter.)

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
bundleName	Unique name for the support bundle. If no name is provided, then "supportbundle" and the node name are used as the file name.	string	None	No
extraArgs	Use '--compress gz' to create the support bundle as a tar.gz file.	string	None	No
timeoutSec	The number of seconds the support bundle script runs.	integer	1500	No

Return values

This method has the following return values:

Name	Description	Type

details	The details of the support bundle. Possible values: • bundleName: The name specified in the CreateSupportBundleAPI method. If no name was specified, "supportbundle" is used. • extraArgs: The arguments passed with this method. • files: A list of the support bundle files that the system created. • output: The command line output from the script that created the support bundle. • timeoutSec: The number of seconds the support bundle script runs before stopping. • url: URL to the support bundle created.	JSON object
duration	The time used to create the support bundle in the format: HH:MM:SS.ssssss.	string
result	The success or failure of the support bundle operation.	string

Request example

Requests for this method are similar to the following example:

```
{
 "method": "CreateSupportBundle",
"params": {
  "extraArgs": "--compress gz"
 },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
"result": {
  "details": {
    "bundleName": "supportbundle",
    "extraArgs": "--compress gz",
    "files": [
         "supportbundle.nodehostname.tar.gz"
     ],
     "output": "timeout -s KILL 1500s /sf/scripts/sfsupportbundle --quiet
--compress gz /tmp/solidfire-dtemp.1L6bdX/supportbundle<br><br>Moved
'/tmp/solidfire-dtemp.1L6bdX/supportbundle.nodehostname.tar.gz' to
/tmp/supportbundles",
      "timeoutSec": 1500,
      "url": [

"https://nodeIP:442/config/supportbundles/supportbundle.nodehostname.tar.g
z"
      ]
    },
    "duration": "00:00:43.101627",
    "result": "Passed"
  }
}
```


New since version

9.6

DeleteAllSupportBundles

You can use the `DeleteAllSupportBundles` method to delete all support bundles generated with the `CreateSupportBundle` API method.

Parameters

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteAllSupportBundles",
   "params": {}
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
 }
}
```

New since version

9.6

DisableMaintenanceMode

You can use the `DisableMaintenanceMode` method to take a storage node out of maintenance mode. You should only disable maintenance mode after you have completed maintenance and the node is online.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
nodes	List of storage node IDs to take out of maintenance mode.	integer array	None	Yes

Return values

This method has the following return values:

Name	Description	Type

asyncHandle	You can use the GetAsyncResult method to retrieve this asyncHandle and determine when the maintenance mode transition is complete.	integer
currentMode	The current maintenance mode state of the node. Possible values: • Disabled: No maintenance has been requested. • FailedToRecover: The node failed to recover from maintenance mode. • Unexpected: The node was found to be offline, but was in the Disabled mode. • RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. • PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. • ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)

requestedMode	The requested maintenance mode state of the node. Possible values: • Disabled: No maintenance has been requested. • FailedToRecover: The node failed to recover from maintenance mode. • Unexpected: The node was found to be offline, but was in the Disabled mode. • RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. • PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. • ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableMaintenanceMode",
  "params": {
    "nodes": [6]
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
        "requestedMode": "Disabled",
        "asyncHandle": 1,
        "currentMode": "Enabled"
     }
}
```

New since version

12.2

Find more information

NetApp HCI storage maintenance mode concepts

DisableSsh

You can use the `DisableSsh` method to disable the SSH service for a single storage node. This method does not affect the cluster-wide SSH service timeout duration.

Parameter

This method has no input parameter.

Return value

This method has the following return value:

Name	Description	Type
enabled	The status of the SSH service for this node.	boolean

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableSsh",
  "params": {
      },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {"enabled": false}
}
```

EnableMaintenanceMode

You can use the `EnableMaintenanceMode` method to prepare a storage node for maintenance. Maintenance scenarios include any task that requires the node to be powered off or restarted.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
forceWithUnresolvedFaults	Force maintenance mode to be enabled for this node even with blocking cluster faults present.	boolean	False	No
nodes	The list of node IDs to put in maintenance mode. Only one node at a time is supported.	integer array	None	Yes
perMinutePrimarySwapLimit	The number of primary slices to swap per minute. If not specified, all primary slices will be swapped at once.	integer	None	No

Name	Description	Type	Default value	Required
timeout	Specifies how long maintenance mode should remain enabled before it is automatically disabled. Formatted as a time string (for example, HH:mm:ss). If not specified, maintenance mode will remain enabled until explicitly disabled.	string	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	You can use the GetAsyncResult method to retrieve this asyncHandle and determine when the maintenance mode transition is complete.	integer
currentMode	The current maintenance mode state of the node. Possible values: • Disabled: No maintenance has been requested. • FailedToRecover: The node failed to recover from maintenance mode. • RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. • PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. • ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)

requestedMode	The requested maintenance mode state of the node. Possible values: • Disabled: No maintenance has been requested. • FailedToRecover: The node failed to recover from maintenance mode. • RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. • PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. • ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableMaintenanceMode",
  "params": {
    "forceWithUnresolvedFaults": False,
    "nodes": [6],
    "perMinutePrimarySwapLimit" : 40,
    "timeout" : "01:00:05"
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
        "requestedMode": "ReadyForMaintenance",
        "asyncHandle": 1,
        "currentMode": "Disabled"
     }
}
```

New since version

12.2

Find more information

NetApp HCI storage maintenance mode concepts

EnableSsh

You can use the `EnableSsh` method to enable the Secure Shell (SSH) service for a single node. This method does not affect the cluster-wide SSH timeout duration, and does not exempt the node from having SSH disabled by the global SSH timeout.

Parameter

This method has no input parameter.

Return value

This method has the following return value:

Name	Description	Type
enabled	The status of the SSH service for this node.	boolean

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableSsh",
  "params": {
     },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {"enabled": true}
}
```

GetClusterConfig

You can use the `GetClusterConfig` API method to return information about the cluster configuration the node uses to communicate with its cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
cluster	Cluster configuration information the node uses to communicate with the cluster.	cluster

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetClusterConfig",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "cluster": {
      "cipi": "Bond10G",
      "cluster": "ClusterName",
      "ensemble": [
        "1:10.30.65.139",
        "2:10.30.65.140",
        "3:10.30.65.141"
      ],
      "fipsDriveConfiguration": true,
      "mipi": "Bond1G",
      "name": "xxx-en142",
      "nodeID": 4,
      "pendingNodeID": 0,
      "role": "Storage",
      "sipi": "Bond10G",
      "state": "Active",
      "version": "9.1.0"
    }
  }
}
```

New since version

9.6

GetClusterState

You can use the `GetClusterState` API method to indicate if a node is part of a cluster or not.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
cluster	Name of the cluster.	string
state	• Available: Node has not been configured with a cluster name. • Pending: Node is pending for a specific named cluster and can be added. • Active: Node is an active member of a cluster and may not be added to another cluster.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetClusterState",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" :
      "cluster" :  "Cluster101"
      "state" : "Active"
}
```

New since version

9.6

GetConfig

You can use the `GetConfig` API method to get all configuration information for a node. This API method includes the same information available in both the `GetClusterConfig` and `GetNetworkConfig` API methods.

Parameters

This method has no input parameters.

Return values

This method has the following return value:

Name	Description	Type
config	The configuration details of the cluster. This object contains: • cluster: Cluster information that identifies how the storage node communicates with the storage cluster it is associated with. • network (all interfaces): Network connection types and current settings for each network interface of the node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetConfig",
   "params": {},
   "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

- GetClusterConfig
- GetNetworkConfig
- GetConfig

GetDriveConfig

You can use the `GetDriveConfig` method to get drive information for expected slice

and block drive counts as well as the number of slices and block drives that are currently connected to the node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
driveConfig	Information on the drives that are connected to the node.	drive

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetDriveConfig",
   "params": {},
   "id" : 1
}
```

Response example

Responses for this method are similar to the following example. Due to length, the response contains information for one drive of one storage node only.

```
{
    "id": 1,
    "result": {
            "driveConfig": {
                    "drives": [
                        {
                            "canonicalName": "sda",
                            "connected": true,
                            "dev": 2052,
                            "devPath": "/dev/sdimm0p4",
                            "driveType": "Slice",
                            "name": "scsi-SATA_VRFSD3400GNCVMT205581853-
part4",
                            "path": "/dev/sda4",
                            "pathLink": "/dev/sdimm0p4",
                            "product": "VRFSD3400GNCVMTKS1",
                            "scsiCompatId": "scsi-
SATA_VRFSD3400GNCVMT205581853-part4",
                            "scsiState": "Running",
                            "securityAtMaximum": false,
                            "securityEnabled": false,
                            "securityFrozen": true,
                            "securityLocked": false,
                            "securitySupported": true,
                            "serial": "205581853",
                            "size": 299988156416,
                            "slot": -1,
                            "uuid": "9d4b198b-5ff9-4f7c-04fc-
3bc4e2f38974",
                            "vendor": "Viking",
                            "version": "612ABBF0"
                        }
                    ],
                    "numBlockActual": 10,
                    "numBlockExpected": 10,
                    "numSliceActual": 1,
                    "numSliceExpected": 1,
                    "numTotalActual": 11,
                    "numTotalExpected": 11
            }
    }
}
```

GetHardwareConfig

You can use the `GetHardwareConfig` method to get the hardware configuration information for a node. This configuration data is intended for internal use. To get a more useful live system hardware component inventory, use the `GetHardwareInfo` method instead.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
hardwareConfig	List of hardware information and current settings.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetHardwareConfig",
   "params": {},
   "id" : 1
}
```

Response example

Responses for this method are similar to the following example.

```
{
    "id": 1,
    "result": {
        "hardwareConfig": {
            "biosRevision": "1.0",
            "biosVendor": [
                "NetApp",
                "SolidFire"
            ],
            "biosVersion": "1.1.2",
            "blockDriveSizeBytes": 300069052416,
            "blockDrives": [
                "/dev/slot0",
```

```
            "/dev/slot1",
            "/dev/slot2",
            "/dev/slot3",
            "/dev/slot4",
            "/dev/slot5",
            "/dev/slot6",
            "/dev/slot7",
            "/dev/slot8",
            "/dev/slot9"
        ],
        "blockServiceFormat": "Standard",
        "bmcFirmwareRevision": "1.6",
        "bmcIpmiVersion": "2.0",
        "chassisType": "R620",
        "cpuCores": 6,
        "cpuCoresEnabled": 6,
        "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
        "cpuThreads": 12,
        "driveSizeBytesInternal": 400088457216,
        "fibreChannelFirmwareRevision": "",
        "fibreChannelModel": "",
        "fibreChannelPorts": {},
        "idracVersion": "1.06.06",
        "ignoreFirmware": [],
        "memoryGB": 72,
        "memoryMhz": 1333,
        "networkDriver": [
            "bnx2x"
        ],
        "nicPortMap": {
            "PortA": "eth2",
            "PortB": "eth3",
            "PortC": "eth0",
            "PortD": "eth1"
        },
        "nodeType": "SF3010",
        "numCpu": 2,
        "numDrives": 10,
        "numDrivesInternal": 1,
        "nvramTempMonitorEnable": false,
        "rootDrive": "/dev/sdimm0",
        "scsiBusExternalDriver": "mpt3sas",
        "scsiBusInternalDriver": "ahci",
        "sliceDriveSizeBytes": 299988156416,
        "sliceDrives": [
            "/dev/sdimm0p4"
```

```
            ],
            "slotOffset": 0,
            "solidfireDefaults": {
                "bufferCacheGB": 12,
                "configuredIops": 50000,
                "cpuDmaLatency": -1,
                "driveWriteThroughputMBPerSleep": 10,
                "maxDriveWriteThroughputMBPerSec": 175,
                "maxIncomingSliceSyncs": 10,
                "postCallbackThreadCount": 8,
                "sCacheFileCapacity": 100000000,
                "sliceFileLogFileCapacity": 5000000000
            }
        }
    }
}
```

New since version

9.6

GetHardwareInfo

You can use the `GetHardwareInfo` method to get live hardware information and status for a single node. Hardware information generally includes manufacturers, vendors, versions, drives, and other associated identification information.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
force	Set this "force" parameter to true to run on all nodes in the cluster.	boolean	false	No

Return value

This method has the following return value:

Name	Description	Type
hardwareInfo	Hardware information for the node.	hardwareInfo

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetHardwareInfo",
   "params": {
   },
"id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "hardwareInfo": {
      "bus": {
        "core_DMI:0200": {
          "description": "Motherboard",
          "physid": "0",
          "product": "0A47AA",
          "serial": "..AB123456C12354.",
          "version": "C07"
        }
      },
      "driveHardware": [
        {
          "canonicalName": "sdh",
          "connected": true,
          "dev": 2160,
          "devPath": "/dev/disk/by-path/pci-0000:41:00.0-sas-
0x500056b37789abf0-lun-0",
          "driveEncryptionCapability": "fips",
          "driveType": "Block",
          "lifeRemainingPercent": 92,
          "lifetimeReadBytes": 175436696911872,
          "lifetimeWriteBytes": 81941097349120,
          "name": "scsi-SATA_INTEL_SSDSC2BB3BTWL12345686300AAA",
          "path": "/dev/sdh",
          "pathLink": "/dev/disk/by-path/pci-0000:41:00.0-sas-
0x500056b37789abf0-lun-0",
          "powerOnHours": 17246,
          "product": "INTEL SSDAA2AA300A4",
```

```
          "reallocatedSectors": 0,
          "reserveCapacityPercent": 100,
          "scsiCompatId": "scsi-SATA_INTEL_SSDSC2BB3BTWL12345686300AAA",
          "scsiState": "Running",
          "securityAtMaximum": false,
          "securityEnabled": false,
          "securityFrozen": false,
          "securityLocked": false,
          "securitySupported": true,
          "serial": "AAAA33710886300AAA",
          "size": 300069052416,
          "slot": 1,
          "smartSsdWriteCapable": false,
          "uuid": "aea178b9-c336-6bab-a61d-87b615e8120c",
          "vendor": "Intel",
          "version": "D2010370"
        },
      ...
      ]
    }
  }
}
```


New since version

9.6

GetIpmiConfig

You can use the `GetIpmiConfig` method to retrieve hardware sensor information from sensors that are in your node.

Parameter

This method has the following input parameter:

Name	Description	Type
chassisType	Used to display information for each node chassis type. Possible values: • all: returns sensor information for each chassis type. • {chassis type}: returns sensor information for a specified chassis type.	string

Return values

This method has the following return values:

Name	Description	Type
sensorName	Name of the sensor that has been found.	string
uniqueSensorID	Unique identifier for the sensor.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetIpmiConfig",
  "params": {
     "chassisType"; "all"
     },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id": 1,
   "result": {
     "nodes": [
         {
           "nodeID": 1,
           "result": {
             "ipmiConfig": {
                "C220M4": [
                    {
                     "sensorName": "Fan1A RPM",
                     "uniqueSensorID": "29.1:0xf"
                    },
                    {
                      "sensorName": "Fan1B RPM",
                      "uniqueSensorID": "29.1:0x10"
                    },
                    {
                      "sensorName": "Fan2A RPM",
```

```
      "uniqueSensorID": "29.2:0x11"
    },
    {
      "sensorName": "Fan2B RPM",
      "uniqueSensorID": "29.2:0x12"
    },
    {
      "sensorName": "Fan3A RPM",
      "uniqueSensorID": "29.3:0x13"
    },
    {
      "sensorName": "Fan3B RPM",
      "uniqueSensorID": "29.3:0x14"
    },
    {
      "sensorName": "Fan4A RPM",
      "uniqueSensorID": "29.4:0x15"
    },
    {
      "sensorName": "Fan4B RPM",
      "uniqueSensorID": "29.4:0x16"
    },
    {
      "sensorName": "Fan5A RPM",
      "uniqueSensorID": "29.5:0x17"
    },
    {
      "sensorName": "Fan5B RPM",
      "uniqueSensorID": "29.5:0x18"
    },
    {
      "sensorName": "Fan6A RPM",
      "uniqueSensorID": "29.6:0x19"
    },
    {
      "sensorName": "Fan6B RPM",
      "uniqueSensorID": "29.6:0x1a"
    },
    {
      "sensorName": "Exhaust Temp",
      "uniqueSensorID": "7.1:0x1"
    },
    {
      "sensorName": "Inlet Temp",
      "uniqueSensorID": "7.1:0x4"
    },
```

```
            {
              "sensorName": "PS1",
              "uniqueSensorID": "10.1:0x26"
            },
            {
              "sensorName": "PS2",
              "uniqueSensorID": "10.2:0x2c"
            }
        ],
        "R620": [
            {
              "sensorName": "Fan1A RPM",
              "uniqueSensorID": "7.1:0x30"
            },
            {
              "sensorName": "Fan1B RPM",
              "uniqueSensorID": "7.1:0x31"
            },
            {
              "sensorName": "Fan2A RPM",
              "uniqueSensorID": "7.1:0x32"
            },
            {
              "sensorName": "Fan2B RPM",
              "uniqueSensorID": "7.1:0x33"
            },
            {
              "sensorName": "Fan3A RPM",
              "uniqueSensorID": "7.1:0x34"
            },
            {
              "sensorName": "Fan3B RPM",
              "uniqueSensorID": "7.1:0x35"
            },
            {
              "sensorName": "Fan4A RPM",
              "uniqueSensorID": "7.1:0x36"
            },
            {
              "sensorName": "Fan4B RPM",
              "uniqueSensorID": "7.1:0x37"
            },
            {
              "sensorName": "Fan5A RPM",
              "uniqueSensorID": "7.1:0x38"
            },
```

```
                    {
                      "sensorName": "Fan5B RPM",
                      "uniqueSensorID": "7.1:0x39"
                    },
                    {
                      "sensorName": "Fan6A RPM",
                      "uniqueSensorID": "7.1:0x3a"
                    },
                    {
                      "sensorName": "Fan6B RPM",
                      "uniqueSensorID": "7.1:0x3b"
                    },
                    {
                      "sensorName": "Fan7A RPM",
                      "uniqueSensorID": "7.1:0x3c"
                    },
                    {
                      "sensorName": "Fan7B RPM",
                      "uniqueSensorID": "7.1:0x3d"
                    },
                    {
                      "sensorName": "Exhaust Temp",
                      "uniqueSensorID": "7.1:0x1"
                    },
                    {
                      "sensorName": "Inlet Temp",
                      "uniqueSensorID": "7.1:0x4"
                    },
                    {
                      "sensorName": "PS1",
                      "uniqueSensorID": "10.1:0x62"
                    },
                    {
                      "sensorName": "PS2",
                      "uniqueSensorID": "10.2:0x63"
                    }
                ],
            }
```


New since version

9.6

GetIpmiInfo

You can use the `GetIpmiInfo` method to display a detailed reporting of sensors

(objects) for node fans, intake and exhaust temperatures, and power supplies that are monitored by the system.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
sensors	Detailed information from each sensor within a node.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetIpmiInfo",
  "params": {},
  "id" : 1
}
```

Response example

Due to the length of the returned response for this API method, portions of the response have been intentionally eliminated from this document. What is included are the portions of the hardware information that the system monitors in order to ensure the node is running at optimum performance.

```
{
  "id": 1,
    "result": {
      "ipmiInfo": {
          "sensors": [
            {
              "entityID": "7.1 (System Board)",
              "sensorID": "0x72",
              "sensorName": "SEL",
              "sensorType": "Event Logging Disabled",
              "uniqueSensorID": "7.1:0x72"
            },
            {
              "assertionsEnabled": [ "General Chassis intrusion" ],
              "deassertionsEnabled": [ "General Chassis intrusion" ],
```

```
              "entityID": "7.1 (System Board)", "sensorID": "0x73",
              "sensorName": "Intrusion",
              "sensorType": "Physical Security",
              "uniqueSensorID": "7.1:0x73"
            },
            {THIS ENTIRE SECTION IS REPEATED FOR EACH FAN IN THE SYSTEM
              "assertionEvents": [],
              "assertionsEnabled": [],
              "deassertionsEnabled": [],
              "entityID": "7.1 (System Board)",
              "eventMessageControl": "Per-threshold",
              "lowerCritical": "720.000",
              "lowerNonCritical": "840.000",
              "maximumSensorRange": "Unspecified",
              "minimumSensorRange": "Unspecified",
              "negativeHysteresis": "600.000",
              "nominalReading": "10080.000",
              "normalMaximum": "23640.000",
              "normalMinimum": "16680.000",
              "positiveHysteresis": "600.000",
              "readableThresholds": "lcr lnc",
              "sensorID": "0x30",
              "sensorName": "Fan1A RPM",
              "sensorReading": "4440 (+/- 120) RPM",
              "sensorType": "Fan",
              "settableThresholds": "",
              "status": "ok",
              "thresholdReadMask": "lcr lnc",
              "uniqueSensorID": "7.1:0x30"
            },
            .
            .
            .
            {THIS ENTIRE SECTION IS REPEATED FOR THE EXHAUST TEMPERATURE
OF EACH NODE
              "assertionEvents": [],
              "assertionsEnabled": [],
              "entityID": "7.1 (System Board)",
              "eventMessageControl": "Per-threshold",
              "lowerCritical": "3.000",
              "lowerNonCritical": "8.000",
              "maximumSensorRange": "Unspecified",
              "minimumSensorRange": "Unspecified",
              "negativeHysteresis": "1.000",
              "nominalReading": "23.000",
              "normalMaximum": "69.000",
```

```
              "normalMinimum": "11.000",
              "positiveHysteresis": "1.000",
              "readableThresholds": "lcr lnc unc ucr",
              "sensorID": "0x1",
              "sensorName": "Exhaust Temp",
              "sensorReading": "44 (+/- 1) degrees C",
              "sensorType": "Temperature",
              "settableThresholds": "",
              "status": "ok",
              "uniqueSensorID": "7.1:0x1",
              "upperCritical": "75.000",
              "upperNonCritical": "70.000"
            },
            {THIS ENTIRE SECTION IS REPEATED FOR THE INLET TEMPERATURE OF
EACH NODE
              "assertionEvents": [],
              "assertionsEnabled": [],
              "deassertionsEnabled": [],
              "entityID": "7.1 (System Board)",
              "eventMessageControl": "Per-threshold",
              "lowerCritical": "-7.000",
              "lowerNonCritical": "3.000",
              "maximumSensorRange": "Unspecified",
              "minimumSensorRange": "Unspecified",
              "negativeHysteresis": "1.000",
              "nominalReading": "23.000",
              "normalMaximum": "69.000",
              "normalMinimum": "11.000",
              "positiveHysteresis": "1.000",
              "readableThresholds": "lcr lnc unc ucr",
              "sensorID": "0x4",
              "sensorName": "Inlet Temp",
              "sensorReading": "20 (+/- 1) degrees C",
              "sensorType": "Temperature",
              "settableThresholds": "lcr lnc unc ucr",
              "status": "ok",
              "thresholdReadMask": "lcr lnc unc ucr",
              "uniqueSensorID": "7.1:0x4",
              "upperCritical": "47.000",
              "upperNonCritical": "42.000"
            },
            {THIS ENTIRE SECTION IS REPEATED FOR EACH POWER SUPPLY ON EACH
NODE
              "assertionEvents": [],
              "assertionsEnabled": [],
              "entityID": "10.2 (Power Supply)",
```

```
                "eventMessageControl": "Per-threshold",
"maximumSensorRange": "Unspecified",
                "minimumSensorRange": "Unspecified",
                "negativeHysteresis": "Unspecified",
                "nominalReading": "0.000",
                "normalMaximum": "0.000",
                "positiveHysteresis": "Unspecified",
                "readableThresholds": "No Thresholds",
                "sensorID": "0x6d",
                "sensorName": "Voltage 2",
                "sensorReading": "118 (+/- 0) Volts",
                "sensorType": "Voltage",
                "settableThresholds": "No Thresholds", "status": "ok",
"uniqueSensorID": "10.2:0x6d"
              },
                .
                .
                .
              }
            ]
          }
        }
      }
```


New since version

9.6

GetNetworkConfig

You can use the `GetNetworkConfig` method to display the network configuration information for a node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
network	Network connection types and current settings for each network interface of the node.	network (all interfaces)

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetNetworkConfig",
   "params": {},
   "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

GetNetworkConfig

GetNetworkInterface

You can use the `GetNetworkInterface` method to get information about a network interface on a node.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
interface	The name of the interface to get information about for each node. Possible values: • Bond1G • Bond10G	string	None	No
force	Set this parameter to true to run on all nodes in the cluster.	boolean	false	No

Return value

This method has the following return value:

Name	Description	Type
nodes	An array of objects describing the interface for each storage node in the storage cluster. Each object within the array contains the following items: • nodeID: (integer) The ID of the storage node in the storage cluster the interface information applies to. • result: (networkInterface) Interface configuration information for this storage node.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "GetNetworkInterface",
    "params": {
        "interface": "Bond1G",
        "force": true
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "nodes": [
            {
                "nodeID": 1,
                "result": {
                    "interface": {
                        "address": "10.117.64.32",
                        "addressV6": "::",
                        "broadcast": "10.117.79.255",
                        "macAddress": "90:b1:1c:42:e0:1e",
                        "mtu": 1500,
```

```
                    "name": "Bond1G",
                    "namespace": false,
                    "netmask": "255.255.240.0",
                    "status": "UpAndRunning",
                    "type": "BondMaster",
                    "virtualNetworkTag": 0
                }
            }
        },
        {
            "nodeID": 2,
            "result": {
                "interface": {
                    "address": "10.117.64.35",
                    "addressV6": "::",
                    "broadcast": "10.117.79.255",
                    "macAddress": "d4:ae:52:7a:ae:23",
                    "mtu": 1500,
                    "name": "Bond1G",
                    "namespace": false,
                    "netmask": "255.255.240.0",
                    "status": "UpAndRunning",
                    "type": "BondMaster",
                    "virtualNetworkTag": 0
                }
            }
        },
        {
            "nodeID": 3,
            "result": {
                "interface": {
                    "address": "10.117.64.39",
                    "addressV6": "::",
                    "broadcast": "10.117.79.255",
                    "macAddress": "c8:1f:66:f0:9d:17",
                    "mtu": 1500,
                    "name": "Bond1G",
                    "namespace": false,
                    "netmask": "255.255.240.0",
                    "status": "UpAndRunning",
                    "type": "BondMaster",
                    "virtualNetworkTag": 0
                }
            }
        },
        {
```

```
                "nodeID": 4,
                "result": {
                    "interface": {
                        "address": "10.117.64.107",
                        "addressV6": "::",
                        "broadcast": "10.117.79.255",
                        "macAddress": "b8:ca:3a:f5:24:f8",
                        "mtu": 1500,
                        "name": "Bond1G",
                        "namespace": false,
                        "netmask": "255.255.240.0",
                        "status": "UpAndRunning",
                        "type": "BondMaster",
                        "virtualNetworkTag": 0
                    }
                }
            }
        ]
    }
}
```


New since version

9.6

GetNodeActiveTlsCiphers

You can use the `GetNodeActiveTlsCiphers` method on a single node to get a list of the TLS ciphers that are currently accepted on this node. You can use this method on management and storage nodes.

Parameter

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
mandatoryCiphers	List of mandatory TLS cipher suites for the node. These are ciphers which are always active on the node.	string
supplementalCiphers	List of supplemental TLS cipher suites for the node.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetNodeActiveTlsCiphers",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
      "id" : 1,
      "result" : {
        "mandatoryCiphers": [
            "DHE-RSA-AES256-SHA256",
            "DHE-RSA-AES256-GCM-SHA384",
            "ECDHE-RSA-AES256-SHA384",
            "ECDHE-RSA-AES256-GCM-SHA384"
        ],
        "supplementalCiphers": [
            "DHE-RSA-AES128-SHA256",
            "DHE-RSA-AES128-GCM-SHA256",
            "ECDHE-RSA-AES128-SHA256",
            "ECDHE-RSA-AES128-GCM-SHA256"
        ]
    }
}
```

GetNodeFipsDrivesReport

You can use the `GetNodeFipsDrivesReport` method to check the FIPS 140-2 drive encryption capability status of a single node in the storage cluster. You must run this method against an individual storage node.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
fipsDrives	A JSON object containing the status of FIPS 140-2 feature support for this node. Possible values: • None: Node is not FIPS capable. • Partial: Node is FIPS capable but not all drives in the node are FIPS drives. • Ready: Node is FIPS capable and all drives in the node are FIPS drives (or no drives are present).	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetNodeFipsDrivesReport",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "fipsDrives": "None"
    }
}
```

New since version

11.5

GetNodeSSLCertificate

You can use the `GetNodeSSLCertificate` method to retrieve the SSL certificate that is currently active on the management node.

Parameters

You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
certificate	The full PEM-encoded text of the certificate.	string
details	The decoded information of the certificate.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "GetNodeSSLCertificate",
    "params" : {},
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD\nVQQGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G\nA1UEChMYV2hhdC
BIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1
UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGF
doYXQgSGFwcGVucyBpbiBWZWdh\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdm
VnYXNzdGF5c2luLnZl\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28f
```

```
nLKQNWEWMR\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FH
P4\nt6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nmrgp
GZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\n7dg9cyXu4afXcgKy14Ji
NBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrn
DJyTsdnEbaGBiaSBhjCBgzELMAkG\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxW
ZWdhcywgQmFieSExITAf\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqG
SIb3DQEJARYe\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYD
VR0T\nBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe\nMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N\niebEgAaE5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl+DwkxnS1cx1v
ERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDg
l0QoQDWNDoTeRBsz80nwiouA==\n-----END CERTIFICATE-----\n",
        "details": {
            "issuer":
"/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org",
            "modulus":
"F14FB6F1F9CB290356116311E9A91E0CAB9E852A52EFDA1D2C68A0235F2A94257F0146396
4B8EAB138C1BD325546FE38CA809380DAF1DFA53B1473F8B7A3FF4A2D1A62BE28BF1979C03
A44337432CB924F07B25E94E07A003EDF9A24F078FDB41D162966F63E533ECB6041429AB82
9199405DE239221C047B4B284E75F3A2554FA8F9760EB28D41903B7E76CA573D1D71DC9FA9
5BFE3CA5D0399535467471A430026212DC99A8CB1FB38FF61AE162AAFB64AA4C05FB6D7D05
DF01C77D79D99479CCF1F113E4DFFD03E2BA952EDD83D7325EEE1A7D77202B2D78262341BE
A6C18E1809B44EFAC80CBAAD31EED313378E376471BF58F2688DCF117E002ABE8AD6B",
            "notAfter": "2027-03-06T22:50:26Z",
            "notBefore": "2017-03-08T22:50:26Z",
            "serial": "CC1B221598E37FF3",
            "sha1Fingerprint":
"1D:70:7A:6F:18:8A:CD:29:50:C7:95:B1:DD:5E:63:21:F4:FA:6E:21",
            "subject":
"/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org"
        }
    }
}
```


GetNodeSupportedTlsCiphers

You can use the `GetNodeSupportedTlsCiphers` method on a single node to get a list of the TLS ciphers that are currently supported on this node. You can use this method on management and storage nodes.

Parameter

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
mandatoryCiphers	List of mandatory TLS cipher suites for the node. These are ciphers which are always active on the node.	string
defaultSupplementalCiphers	List of default supplemental TLS cipher suites for the node. The supplemental ciphers are restored to this list when you run the ResetNodeSupplementalTlsCiphers API method.	string
supportedSupplementalCiphers	List of available supplemental TLS cipher suites which you can configure with the SetNodeSupplementalTlsCiphers API method.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetNodeSupportedTlsCiphers",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
    "defaultSupplementalCiphers": [
        "DHE-RSA-AES128-SHA256",
        "DHE-RSA-AES128-GCM-SHA256",
        "ECDHE-RSA-AES128-SHA256",
        "ECDHE-RSA-AES128-GCM-SHA256"
    ],
    "mandatoryCiphers": [
        "DHE-RSA-AES256-SHA256",
        "DHE-RSA-AES256-GCM-SHA384",
        "ECDHE-RSA-AES256-SHA384",
        "ECDHE-RSA-AES256-GCM-SHA384"
    ],
    "supportedSupplementalCiphers": [
        "DHE-RSA-AES128-SHA256",
        "DHE-RSA-AES128-GCM-SHA256",
        "ECDHE-RSA-AES128-SHA256",
        "ECDHE-RSA-AES128-GCM-SHA256",
        "DHE-RSA-AES256-SHA",
        "ECDHE-RSA-AES256-SHA",
        "DHE-RSA-CAMELLIA256-SHA",
        "DHE-RSA-AES128-SHA",
        "ECDHE-RSA-AES128-SHA",
        "DHE-RSA-CAMELLIA128-SHA"
    ]
}
}
```


GetPatchInfo

You can use the `GetPatchInfo` method to get information about Element software patches installed on a storage node.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`force`	Force the method to run on all nodes in the storage cluster. You only need this when you issue the API to a cluster IP address instead of a single node. Possible values: • `true` • `false`	boolean	`false`	No

Return values

This method has the following return values:

Name	Description	Type
patches	Object containing information about the patches installed on this node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetPatchInfo",
   "params": {
      "force": false,
      },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "patches": {
          "SUST936": {
          "date": "Wed 09 Dec 2020 10:41:59 PM UTC",
          "description": "BMC fixes",
          "newFiles": [
              "None"
          ],
          "patchedFiles": [
              "Patched_file_1.bin",
              "Patched_file_2.dat",
              "Patched_file_3.tgz"
          ]
          }
        }
    }
}
```


New since version

12.3

GetPendingOperation

You can use the `GetPendingOperation` method to detect an operation on a node that is currently in progress. This method can also be used to report back when an operation has completed.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
pending	Possible values: • true: The operation is still in progress. • false: The operation is no longer in progress.	boolean
operation	Name of operation that is in progress or has completed.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetPendingOperation",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "pendingOperation" : {
         "pending" : "true",
         "operation" : "TestDrivesInternal",
      }
   }
}
```

New since version

9.6

GetSshInfo

You can use the `GetSshInfo` method to query the status of the SSH service on a single node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
result	The status of the SSH service for this node.	boolean

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "GetSshInfo",
    "params" : {},
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "enabled": false
    }
}
```

ListDriveHardware

You can use the `ListDriveHardware` method to list all the drives connected to a node. When used on individual nodes, this method returns drive hardware information. When used on the cluster master node MVIP, this method returns information for all drives on all nodes.

Parameters

> The "securitySupported": true line of the method response does not imply that the drives are capable of encryption; only that the security status can be queried. If you have a node type with a model number ending in "-NE", commands to enable security features on these drives will fail.

This method has the following parameter:

Name	Description	Type	Default value	Required
force	Set to true to run this method on all nodes.	boolean	None	No

Return value

This method has the following return value:

Name	Description	Type
driveHardware	Returned drive hardware information for the node.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
"method": "ListDriveHardware",
"params": {},
"id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "driveHardware": [
      {
        "canonicalName": "sda",
        "connected": true,
        "dev": 2048,
        "devPath": "/dev/slot0",
        "driveEncryptionCapability": "fips",
        "driveType": "Slice",
        "lifeRemainingPercent": 98,
        "lifetimeReadBytes": 0,
        "lifetimeWriteBytes": 14012129542144,
        "name": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
        "path": "/dev/sda",
        "pathLink": "/dev/slot0",
        "powerOnHours": 15489,
        "product": "SAMSUNG MZ7GE240HMGR-00003",
        "reallocatedSectors": 0,
        "reserveCapacityPercent": 100,
        "scsiCompatId": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
        "scsiState": "Running",
        "securityAtMaximum": false,
        "securityEnabled": true,
        "securityFrozen": false,
        "securityLocked": false,
        "securitySupported": true,
        "serial": "S1M9NWAG501251",
        "size": 240057409536,
        "slot": 0,
        "uncorrectableErrors": 0,
        "uuid": "789aa05d-e49b-ff4f-f821-f60eed8e43bd",
        "vendor": "Samsung",
        "version": "EXT1303Q"
      }
    ]
}
```


New since version

9.6

Find more information

EnableEncryptionAtRest

ListNetworkInterfaces

You can use the `ListNetworkInterfaces` method to list information about each network interface on a node. This API method is intended for use on individual nodes; user ID and password authentication is required for access to individual nodes. However, you can use this method on the cluster if the parameter force is given the value true in the method call. When the parameter is used on the cluster, all interfaces are listed.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
force	Possible values: • true: Information about all network interfaces in the cluster is returned. • false: No information is returned.	boolean	None	No

Return value

This method has the following return value:

Name	Description	Type
interfaces	A list of configuration information for each network interface of the storage node (or entire storage cluster, if force = `true`).	networkInterface array

Request example

Requests for this method are similar to the following example:

```
{
"method": "ListNetworkInterfaces",
"params": {},
"id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "nodes": [
            {
                "nodeID": 1,
                "result": {
                    "interfaces": [
                        {
                            "address": "10.117.80.32",
                            "addressV6": "::",
                            "broadcast": "10.117.95.255",
                            "macAddress": "90:b1:1c:42:e0:1a",
                            "mtu": 9000,
                            "name": "Bond10G",
                            "namespace": false,
                            "netmask": "255.255.240.0",
                            "status": "UpAndRunning",
                            "type": "BondMaster",
                            "virtualNetworkTag": 0
                        },
                        {
                            "address": "10.117.64.32",
                            "addressV6": "::",
                            "broadcast": "10.117.79.255",
                            "macAddress": "90:b1:1c:42:e0:1e",
                            "mtu": 1500,
                            "name": "Bond1G",
                            "namespace": false,
                            "netmask": "255.255.240.0",
                            "status": "UpAndRunning",
                            "type": "BondMaster",
                            "virtualNetworkTag": 0
                        },
                        {
```

```
                            "address": "0.0.0.0",
                            "addressV6": "::",
                            "broadcast": "0.0.0.0",
                            "macAddress": "90:b1:1c:42:e0:1a",
                            "mtu": 9000,
                            "name": "eth0",
                            "namespace": false,
                            "netmask": "0.0.0.0",
                            "status": "UpAndRunning",
                            "type": "BondSlave",
                            "virtualNetworkTag": 0
                        },
                        {
                            "address": "127.0.0.1",
                            "addressV6": "::",
                            "broadcast": "0.0.0.0",
                            "macAddress": "00:00:00:00:00:00",
                            "mtu": 0,
                            "name": "lo",
                            "namespace": false,
                            "netmask": "0.0.0.0",
                            "status": "UpAndRunning",
                            "type": "Loopback",
                            "virtualNetworkTag": 0
                        }
                    ]
                }
            }
        ]
    }
}
```


New since version

9.6

ListNetworkInterfaceStats

You can use the `ListNetworkInterfaceStats` method to list statistics such as the number of dropped packets and various types of errors for each network interface on a node. This API method is intended for use on individual nodes; user ID and password authentication is required for access to individual nodes. However, you can use this method on the cluster if the parameter force is given the value true in the method call. When the parameter is used on the cluster, network statistics for all interfaces are listed.

Parameter

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
networkInterfaceStats	A list of network statistics information, such as the number of dropped packets and various types of network errors, for each network interface of a storage node.	networkInterfaceStats array

Request example

Requests for this method are similar to the following example:

```
{
"method": "ListNetworkInterfaceStats",
"params": {},
"id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "networkInterfaceStats": [
        {
            "rxErrors": 1,
            "rxPackets": 1,
            "txErrors": 1,
            "rxDropped": 1,
            "txCarrierErrors": 1,
            "rxOverErrors": 1,
            "rxMissedErrors": 1,
            "txPackets": 1,
            "name": "if_name",
            "rxLengthErrors": 1,
            "collisions": 1,
            "rxFifoErrors": 1,
            "txBytes": 1,
            "rxBytes": 1,
            "rxFrameErrors": 1,
            "rxCrcErrors": 1,
            "txFifoErrors": 1
        }
    ]
}
```


New since version

12.3

ListTests

You can use the `ListTests` method to list the tests that are available to run on a node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
tests	List of tests that can be performed on the node.	string array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListTests",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
   "tests": [
             "TestConnectEnsemble",
             "TestConnectMvip",
             "TestConnectSvip",
             "TestDrives",
             "TestHardwareConfig",
             "TestLocateCluster",
             "TestPing",
             "TestLocalConnectivity",
             "TestRemoteConnectivity",
             "TestNetworkConfig"
           ]
      }
}
```

New since version

9.6

ListUtilities

You can use the `ListUtilities` method to list the operations that are available to run on a node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
utilities	List of utilities currently available to run on the node.	string array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListUtilities",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "utilities": [
      "ResetDrives",
      "ResetNode",
      "RestartNetworking",
      "RestartServices",
      "CreateSupportBundle",
      "DeleteAllSupportBundles",
      "CreateClusterSupportBundle"
    ]
  }
}
```

New since version

9.6

RemoveNodeSSLCertificate

You can use the `RemoveNodeSSLCertificate` method to remove the user SSL certificate and private key for the management node. After the certificate and private key

are removed, the management node is configured to use the default certificate and private key.

Parameters

You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "RemoveNodeSSLCertificate",
    "params" : {},
    "id" : 3
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id" : 3,
    "result" : {}
}
```

ResetDrives

You can use the `ResetDrives` method to proactively initialize drives and remove all data currently residing on a drive. The drive can then be reused in an existing node or used in an upgraded node.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
drives	List of device names (not driveIDs) to reset.	string	None	Yes
force	Set to true to reset the drive.	boolean	None	Yes

Return value

This method has the following return value:

Name	Description	Type
details	Details of drives that are being reset.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ResetDrives",
   "params": {
      "drives" : "slot3",
      "force" : true
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
   "details": {
    "drives": [
    {
      "drive": "slot3",
      "returnCode": 0,
      "stderr": " * Unlocking /dev/slot9 .[ ok ]\ * Setting master
password /dev/slot9 .[ ok ]\ * Secure erasing /dev/slot9 (hdparm)
[tries=0/1] .........................[ ok ]",
      "stdout": ""
    }
   ]
  },
  "duration": "00:00:28.501269",
  "result": "Passed"
  }
}
```


New since version

9.6

ResetNode

You can use the `ResetNode` method to reset a node to the factory settings. All data, packages (software upgrades, etc), configurations, and log files are deleted from the node when you call this method. However, network settings for the node are preserved during this operation. Nodes that are participating in a cluster cannot be reset to the factory settings.

Parameters

The ResetNode API can only be used on nodes that are in an "Available" state. It cannot be used on nodes that are "Active" in a cluster, or in a "Pending" state.

CAUTION:

This method clears any customer data that is on the node.

This method has the following input parameters:

Name	Description	Type	Default value	Required
build	Used to specify the URL to a remote Element software image to which the node will be reset.	URL	None	No
force	Set to true to reset the node.	boolean	None	Yes
options	Used to enter specifications for running the reset operations. Details are be provided by NetApp Support, if required.	JSON object	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ResetNode",
   "params": {
      "build" : "file:///sf/rtfi/image/filesystem.squashfs",
      "force" : true
              },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": null,
  "result": {
    "rtfiInfo": {
      "build": "file:///sf/rtfi/image/filesystem.squashfs",
      "generation": "9",
      "options": {
```

```
        "edebug": "",
        "sf_auto": "0",
        "sf_bond_mode": "ActivePassive",
        "sf_check_hardware": "0",
        "sf_disable_otpw": "0",
        "sf_fa_host": "",
        "sf_hostname": "SF-FA18",
        "sf_inplace": "1",
        "sf_inplace_die_action": "kexec",
        "sf_inplace_safe": "0",
        "sf_keep_cluster_config": "0",
        "sf_keep_data": "0",
        "sf_keep_hostname": "0",
        "sf_keep_network_config": "0",
        "sf_keep_paths": "\"/var/log/hardware.xml\"",
        "sf_max_archives": "5",
        "sf_nvram_size": "",
        "sf_oldroot": "",
        "sf_postinst_erase_root_drive": "0",
        "sf_root_drive": "",
        "sf_rtfi_cleanup_state": "",
        "sf_secure_erase": "1",
        "sf_secure_erase_retries": "5",
        "sf_slice_size": "",
        "sf_ssh_key": "1",
        "sf_ssh_root": "1",
        "sf_start_rtfi": "1",
        "sf_status_httpserver": "1",
        "sf_status_httpserver_stop_delay": "5m",
        "sf_status_inject_failure": "",
        "sf_status_json": "0",
        "sf_support_host": "sfsupport.solidfire.com",
        "sf_test_hardware": "0",
        "sf_upgrade": "0",
        "sf_upgrade_firmware": "0",
        "sf_upload_logs_url": ""
      },
       "statusUrlAll": "http://192.168.130.20/status/all.json",
       "statusUrlCurrent": "http://192.168.130.20/status/current.json"
     }
   }
}
```


New since version

9.6

ResetNodeSupplementalTlsCiphers

You can use the `ResetNodeSupplementalTlsCiphers` method to restore the list of supplemental TLS ciphers to the default. You can use this command on management nodes.

Parameter

You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ResetNodeSupplementalTlsCiphers",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

RestartNetworking

You can use the `RestartNetworking` method to restart the networking services on a node.

CAUTION:

This method restarts all networking services on a node, causing temporary loss of networking connectivity.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
force	Set to true to restart networking services on a node.	boolean	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RestartNetworking",
   "params": {
              "force" : true
             },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{ "id" : 1,
  "result" : {}
}
```

New since version

9.6

RestartServices

You can use the `RestartServices` method to restart the services on a node.

Parameters

CAUTION:

This method causes temporary node services interruption.

This method has the following input parameters:

Name	Description	Type	Default value	Required
force	Set to true to restart services on a node.	boolean	None	Yes
service	Service name to be restarted.	string	None	No
action	Action to perform on the service (start, stop, restart).	string	None	No

Return values

This method has the following return values:

Name	Description	Type
details	The output of the service restart procedure, including errors (if any).	JSON object
duration	The time, in seconds, it took to restart services to the node.	string
result	Results of the restart.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RestartServices",
   "params": {
              "force" : true
              "action" : restart,
             }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "details": "solidfire stop/waiting\nsolidfire start/running, process
7284\n",
    "duration": "00:00:02.541594",
    "result": "Passed"
  }
}
```


New since version

9.6

SetClusterConfig

You can use the `SetClusterConfig` method to set the configuration that a node uses to communicate with the cluster it is associated with. To display the current cluster interface settings for a node, run the `GetClusterConfig` API method.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
cluster	Configuration attributes that should be changed during this method call. Only the fields you want changed need to be added to this method as members in this parameter.	cluster	None	No

Return value

This method has the following return value:

Name	Description	Type
cluster	Configuration information the node uses to communicate with the cluster.	cluster

Request example

Requests for this method are similar to the following example:

```
{
   "method": "SetClusterConfig",
   "params": {
      "cluster": {
         "name": "myhost",
         "mipi": "Bond10G"
      },
    "id" : 1
   }
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "cluster" : {
         "cipi" : "Bond10G",
         "cluster" : "QoS",
         "ensemble" : [
            "1:10.10.5.42",
            "2:10.10.5.43",
            "3:10.10.5.44",
            "4:10.10.5.46",
            "5:10.10.5.47"
         ],
          "hostname" : "myhost",
          "mipi" : "Bond10G",
          "nodeID" : 1,
          "sipi" : "Bond10G",
          "state" : "Active"
      }
   }
}
```

New since version

9.6

SetConfig

You can use the `SetConfig` method to set the network and cluster information for the node. This method includes the same settings in a single API method that are available using both `SetClusterConfig` and `SetNetworkConfig` methods. Only the fields you want changed need to be included with this method.

Parameter

CAUTION:

Changing the bond-mode on a node can cause a temporary loss of network connectivity.

This method has the following input parameters:

Name	Description	Type	Default value	Required
cluster	Cluster information that identifies how the storage node communicates with the storage cluster it is associated with.	cluster	None	No
network	Network connection types and current settings for each network interface of the node.	network (all interfaces)	None	No

Return value

This method has the following return value:

Name	Description	Type
config	The new and current configuration for the node. This object contains: • cluster: Cluster information that identifies how the storage node communicates with the storage cluster it is associated with. • network (all interfaces): Network connection types and current settings for each network interface of the node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
    "method": "SetConfig",
    "params": {
        "cluster": {
            "name": "MyHostname"
        },
        "network": {
            "Bond10G": {
                "bond-mode": "ALB"
            }
        }
    }
}
```

Response example

The response from this method is the same as the return for the GetConfig method. All fields for the object display and updated values are seen when SetConfig is used.

New since version

9.6

Find more information

- SetClusterConfig
- SetNetworkConfig
- GetConfig

SetNetworkConfig

You can use the `SetNetworkConfig` method to set the network configuration for a node. To display the current network settings for a node, run the `GetNetworkConfig` API method.

Parameter

CAUTION:

Changing the bond-mode on a node can cause a temporary loss of network connectivity.

This method has the following input parameter:

Name	Description	Type	Default value	Required
network	An object containing node network settings to modify. You only need to add the fields you want changed to this method as attributes in this parameter.	network (all interfaces)	None	No

Return value

This method has the following return value:

Name	Description	Type
network	The new and current network configuration for the node.	network (all interfaces)

Request example

Requests for this method are similar to the following example:

```
{
    "method": "SetNetworkConfig",
        "params": {
            "network": {
                "Bond10G": {
                    "bond-mode": "ALB"
                },
                "Bond1G": {
                    "netmask": "255.255.224.0"
                },
                "eth0": {
                    "method": "bond"
                },
                "lo": {
                    "method": "loopback"
                }
            }
        }
}
```

Response example

The response from this method is the same as the response from the GetNetworkConfig method. The method

displays all members for each object and includes the new values for any changed members.

New since version

9.6

Find more information

- GetNetworkConfig
- GetNetworkConfig

SetNodeSSLCertificate

You can use the `SetNodeSSLCertificate` method to set a user SSL certificate and private key for the management node.

After using the API, you must reboot the management node.

Parameters

You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has the following input parameters:

Name	Description	Type	Default value	Required
certificate	The PEM-encoded text version of the certificate.**Note:** When setting a node or cluster certificate, the certificate must include the extendedKeyUsage extension for serverAuth. This extension allows the certificate to be used without error on common operating systems and browsers. If the extension is not present, the API will reject the certificate as invalid.	string	None	Yes

Name	Description	Type	Default value	Required
privateKey	The PEM-encoded text version of the private key.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "SetNodeSSLCertificate",
    "params" : {
        "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEA8U+28fnLKQNWEWMR6akeDKuehSpS79odLGigI18qlCV/AUY5\nZLjqsT
jBvTJVRv44yoCTgNrx36U7FHP4t6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP
20HRYpZvY+Uz7LYEFCmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg\n6yjUGQO352ylc9HXHcn6lb
/jyl0DmVNUZ0caQwAmIS3Jmoyx+zj/Ya4WKq+2SqTA\nX7bX0F3wHHfXnZlHnM8fET5N/9A+K6
lS7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE\n76yAy6rTHu0xM3jjdkcb9Y8miNzxF+ACq+itaw
IDAQABAoIBAH1jlIZr6/sltqVW\nO0qVC/49dyNu+KWVSq92ti9rFe7hBPueh9gklh78hP9Qli
tLkir3YK4GFsTFUMux\n7z1NRCxA/4LrmLSkAjW2kRXDfVl2bwZq0ua9NefGw92O8D2OZvbuOx
k7Put2p6se\nfgNzSjf2SI5DIX3UMe5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7w
nq\nt+Am5/Vu1LI6rGiG6oHEW0oGSuHl1esIyXXa2hqkU+1+iF2iGRMTiXac4C8d11NU\nWGIR
CXFJAmsAQ+hQm7pmtsKdEqumj/PIoGXf0BoFVEWaIJIMEgnfuLZp8IelJQXn\nSFJbk2ECgYEA
+d5ooU4thZXylWHUZqomaxyzOruA1T53UeH69HiFTrLjvfwuaiqj\nlHzPlhms6hxexwz1dzAp
gog/NOM+2bAc0rn0dqvtV4doejtlDZKRqrNCf/cuN2QX\njaCJClCWau3sEHCckLOhWeY4HaPS
oWq0GKLmKkKDChB4nWUYg3gSWQkCgYEA9zuN\nHW8GPS+yjixeKXmkKO0x/vvxzR+J5HH5znaI
Hss48THyhzXpLr+v30Hy2h0yAlBS\nny5Ja6wsomb0mVe4NxVtVawg2E9vVvTa1UC+TNmFBBuL
RPfjcnjDerrSuQ5lYY+M\nC9MJtXGfhp//G0bzwsRzZxOBsUJb15tppaZIs9MCgYAJricpkKjM
0xlZ1jdvXsos\nPilnbho4qLngrzuUuxKXEPEnzBxUOqCpwQgdzZLYYw788TCVVIVXLEYem2s0
7dDA\nDTo+WrzQNkvC6IgqtXH1RgqegIoG1VbgQsbsYmDhdaQ+os4+AOeQXw3vgAhJ/qNJ\njQ
4Ttw3ylt7FYkRH26ACWQKBgQC74Zmf4JuRLAo5WSZFxpcmMvtnlvdutqUH4kXA\nzPssy6t+QE
La1fFbAXkZ5Pg1ITK752aiaX6KQNG6qRsA3VS1J6drD9/2AofOQU17\n+jOkGzmmoXf49Zj3iS
akwg0ZbQNGXNxEsCAUr0BYAobPp9/fB4PbtUs99fvtocFr\njS562QKBgCb+JMDP5q7jpUuspj
0obd/ZS+MsomE+gFAMBJ71KFQ7KuoNezNFO+ZE\n3rnR8AqAm4VMzqRahs2PWNe2Hl4J4hKu96
qNpNHbsW1NjXdAL9P7oqQIrhGLVdhX\nInDXvTgXMdMoet4BKnftelrXFKHgGqXJoczq4JWzGS
IHNgvkrH60\n-----END RSA PRIVATE KEY-----\n",
        "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD\nVQQGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G\nA1UEChMYV2hhdC
BIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1
UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGF
```

```
doYXQgSGFwcGVucyBpbiBWZWdh\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdm
VnYXNzdGF5c2luLnZl\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28f
nLKQNWEWMR\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FH
P4\nt6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nmrgp
GZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\n7dg9cyXu4afXcgKy14Ji
NBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrn
DJyTsdnEbaGBiaSBhjCBgzELMAkG\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxW
ZWdhcywgQmFieSExITAf\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqG
SIb3DQEJARYe\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYD
VR0T\nBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe\nMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N\niebEgAaE5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl+DwkxnS1cx1v
ERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDg
l0QoQDWNDoTeRBsz80nwiouA==\n-----END CERTIFICATE-----\n"
    },
    "id" : 2
}
```


Response example

This method returns a response similar to the following example:


```
{
    "id" : 2,
    "result" : {}
}
```


SetNodeSupplementalTlsCiphers

You can use the `SetNodeSupplementalTlsCiphers` method to specify the list of supplemental TLS ciphers. You can use this command on management nodes.

Parameter

You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has the following input parameter:

Name	Description	Type	Default value	Required
supplementalCiphers	The supplemental cipher suite names using the OpenSSL naming scheme. Use of cipher suite names is case-insensitive.	string	None	Yes

Return values

This method has the following return values:

Name	Description	Type
mandatoryCiphers	List of mandatory TLS cipher suites for the node. These are ciphers which are always active on the node.	string
supplementalCiphers	List of supplemental TLS cipher suites for the node.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "SetNodeSupplementalTlsCiphers",
   "params": {
        "supplementalCiphers": [
            "DHE-RSA-AES128-SHA256",
            "DHE-RSA-AES128-GCM-SHA256",
            "ECDHE-RSA-AES128-SHA256",
            "ECDHE-RSA-AES128-GCM-SHA256"
        ]
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
        "mandatoryCiphers": [
            "DHE-RSA-AES256-SHA256",
            "DHE-RSA-AES256-GCM-SHA384",
            "ECDHE-RSA-AES256-SHA384",
            "ECDHE-RSA-AES256-GCM-SHA384"
        ],
        "supplementalCiphers": [
            "DHE-RSA-AES128-SHA256",
            "DHE-RSA-AES128-GCM-SHA256",
            "ECDHE-RSA-AES128-SHA256",
            "ECDHE-RSA-AES128-GCM-SHA256"
        ]
    }
}
```

Shutdown

You can use the `Shutdown` method to restart or shutdown the nodes in a cluster. You can shut down a single node, multiple nodes, or all of the nodes in the cluster using this method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
nodes	List of NodeIDs for the nodes to be restarted or shut down.	integer array	None	Yes
option	Action to take for the cluster. Possible values: • restart: Restarts the cluster. • halt: Performs a full power-off.	string	restart	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "Shutdown",
   "params": {
     "nodes": [
       2,
       3,
       4
     ],
     "option": "halt"
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
     "failed": [],
     "successful": [
          6
      [
   }
}
```

New since version

9.6

TestConnectEnsemble

You can use the `TestConnectEnsemble` method to verify connectivity with a specified database ensemble. By default it uses the ensemble for the cluster the node is associated with. Alternatively you can provide a different ensemble to test connectivity.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
ensemble	A comma-separated list of ensemble node cluster IP addresses for connectivity testing.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
details	Objects returned: • `nodes`: (object) A list of each ensemble node in the test and the results of the tests. • `duration`: (string) The time required to run the test. • `result`: (string) The results of the entire test.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "TestConnectEnsemble",
  "params": {},
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
      "details": {
         "nodes":  {
            "1:10.10.20.70": "Passed",
            "2:10.10.20.71": "Passed",
            "3:10.10.20.72": "Passed",
            "4:10.10.20.73": "Passed",
            "5:10.10.20.74": "Passed"
        }
      },
        "duration": "00:00:00:756072",
        "result": "Passed"
   }
}
```


New since version

9.6

TestConnectMvip

You can use the `TestConnectMvip` method to test the management connection to the storage cluster. The test pings the MVIP and executes a simple API method to verify connectivity.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
mvip	You can pass this value to test the management connection of a different MVIP. You do not need to use this value when testing the connection to the target cluster.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
details	Information about the test operation (JSON object): • `connected`: Indicates if the test could connect to the MVIP (boolean) • `mvip`: The MVIP tested against (string) • `pingBytes`: Details of the ping tests with 56 bytes and 1500 bytes (object) ◦ `56`: Results of the 56 Byte ping test (JSON object): ▪ `individualResponseTimes`: List of response times from each ensemble node (string array) ▪ `individualStatus`: List of ping status from each ensemble node (boolean array) ▪ `responseTime`: Average ping response time (string) ▪ `successful`: Indicates if the ping test was successful (boolean) ◦ `1500`: Results of the 1500 byte ping test (JSON object): ▪ `individualResponseTimes`: List of response times from each ensemble node (string array) ▪ `individualStatus`: List of ping status from each ensemble node (boolean array) ▪ `responseTime`: Average ping response time (string) ▪ `successful`: Whether the ping test was successful (boolean) `duration`: Length of time required to run the test (string)	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "TestConnectMvip",
   "params": {
       "mvip" : "172.27.62.50"
       },
    "id":1
}
```

- `result`: Result of the test as a whole (string)

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "details": {
      "connected": true,
      "mvip": "172.27.62.50",
      "pingBytes": {
        "1500": {
          "individualResponseTimes": [
            "00:00:00.000250",
            "00:00:00.000206",
            "00:00:00.000200",
            "00:00:00.000199",
            "00:00:00.000199"
         ],
          "individualStatus": [
             true,
             true,
             true,
             true,
             true
         ],
          "responseTime": "00:00:00.000211",
          "successful": true
       },
       "56": {
          "individualResponseTimes": [
            "00:00:00.000217",
            "00:00:00.000122",
            "00:00:00.000117",
```

```
            "00:00:00.000119",
            "00:00:00.000121"
         ],
         "individualStatus": [
            true,
            true,
            true,
            true,
            true
         ],
         "responseTime": "00:00:00.000139",
         "successful": true
        }
      }
    },
    "duration": "00:00:00.271244",
    "result": "Passed"
  }
}
```


New since version

9.6

TestConnectSvip

You can use the `TestConnectSvip` method to test the storage connection to the storage cluster. The test pings the SVIP using ICMP packets, and when successful, connects as an iSCSI initiator.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
svip	You can pass this value to test the management connection of a different SVIP. You do not need to use this value when testing the connection to the target cluster.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
details	Information about the test operation (JSON object): • `connected`: Indicates if the test could connect to the SVIP (boolean) • `svip`: The SVIP tested against (string) • `pingBytes`: Details of the ping tests with 56 bytes and 9000 bytes (object) ◦ `56`: Results of the 56 byte ping test (JSON object): ▪ `individualResponseTimes`: List of response times from each ensemble node (string array) ▪ `individualStatus`: List of ping status from each ensemble node (boolean array) ▪ `responseTime`: Average ping response time (string) ▪ `successful`: Indicates if the ping test was successful (boolean) ◦ `9000`: Results of the 9000 Byte ping test (JSON object): ▪ `individualResponseTimes`: List of response times from each ensemble node (string array) ▪ `individualStatus`: List of ping status from each ensemble node (boolean array) ▪ `responseTime`: Average ping response time (string) ▪ `successful`: Indicates if the ping test was successful (boolean) `duration`: Length of time required to run the test (string)	string

Request example

Requests for this method are similar to the following example:

- result: Result of the test as a whole (string)

```
{
   "method": "TestConnectSvip",
   "params": {
      "svip" : "172.27.62.50"
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "details": {
      "connected": true,
      "pingBytes": {
           "56": {
               "individualResponseTimes": [
                   "00:00:00.000152",
                   "00:00:00.000132",
                   "00:00:00.000119",
                   "00:00:00.000114",
                   "00:00:00.000112"
              ],
              "individualStatus": [
                  true,
                  true,
                  true,
                  true,
                  true
              ],
              "responseTime": "00:00:00.000126",
              "successful": true
           },
          "9000": {
                "individualResponseTimes": [
                    "00:00:00.000295",
                    "00:00:00.000257",
                    "00:00:00.000172",
                    "00:00:00.000172",
```

```
                    "00:00:00.000267"
                ],
                "individualStatus": [
                    true,
                    true,
                    true,
                    true,
                    true
                ],
                "responseTime": "00:00:00.000233",
                "successful": true
            }
          },
          "svip": "172.27.62.50"
        },
        "duration": "00:00:00.421907",
        "result": "Passed"
    }
}
```


New since version

9.6

TestDrives

You can use the `TestDrives` method to run a hardware validation on all drives on the node. This method detects hardware failures on the drives and reports any in the results of the validation tests.

Parameters

You can only use the `TestDrives` method on nodes that are not "Active" in a cluster.

This test takes approximately 10 minutes.

This method has the following input parameters:

Name	Description	Type	Default value	Required
force	Set to true to test the drives on the node.	boolean	None	Yes
minutes	Specifies the number of minutes for the test to run.	integer	10	No

Return value

This method has the following return value:

Name	Description	Type
details	Information about the test operation success or failure.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "TestDrives",
   "params": {
              "force": true,
              "minutes" : 10
             },
   "id" : 1
}
```

Response example

This method returns a table containing test results for each drive in the node.

New since version

9.6

TestHardwareConfig

You can use the `TestHardwareConfig` method to perform hardware tests on a node. Test options include verifying hardware configurations, firmware versions, and that all drives are present.

Parameters

These test are not intended to detect hardware failures.

This method has the following input parameters:

Name	Description	Type	Default value	Required
clean	Starts the hardware configuration test with a clean cache. Possible values: • true: Deletes the cached test results file and reruns the tests. • false: Retrieves a cached test results.	boolean	false	No
force	The force parameter must be included in this method to successfully reset the node.	boolean	None	Yes

Return value

This method has the following return value:

Name	Description	Type
details	Hardware configuration details.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "TestHardwareConfig",
   "params": {
      "force": true
     },
   "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

TestHardwareConfig

TestLocateCluster

You can use the `TestLocateCluster` method to validate that the node can locate the cluster specified in the cluster configuration. The output validates that the cluster has been created and lists the nodes in the cluster ensemble.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
details	Information about the test operation success or failure.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "TestLocateCluster",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
      "details": {
         "complete": true,
         "ensemble": {
            "nodes": [
               {
                 "IP": "10.10.5.94",
                 "nodeID": 1
               },
               {
                 "IP": "10.10.5.107",
                 "nodeID": 2
               },
               {
                 "IP": "10.10.5.108",
                 "nodeID": 3
               }
            ]
         },
         "version": "5.749"
      },
      "duration": "0.0384478sec",
      "result": "Passed"
   }
}
```


New since version

9.6

TestLocalConnectivity

You can use the `TestLocalConnectivity` method to ping the Cluster IP (CIP) of each node in an active cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
details	Individual ping response times for each node in the local, active cluster.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "TestLocalConnectivity",
   "params": {},
   "id": 1
}
```

Response example

Requests for this method are similar to the following example:

```
{
   "id": null,
   "result": {
      "details": {
                  "10.26.86.17": {
                     individualResponseTimes: [
                          "00:00:00.006868",
                          "00:00:00.005933",
                          "00:00:00.006655",
                          "00:00:00.006584",
                          "00:00:00.006334"
                     ],
                     individualStatus: [
                           true,
                           true,
                           true,
                           true,
                           true
                     ],
                     responseTime: "00:00:00.006475",
                     successful: true
                        },
                        "10.26.86.18": {
                     individualResponseTimes: [
                          "00:00:00.006201",
```

```
                    "00:00:00.006187",
                    "00:00:00.005990",
                    "00:00:00.006029",
                    "00:00:00.005917"],
                individualStatus: [
                     true,
                     true,
                     true,
                     true,
                     true
                ],
                "responseTime": "00:00:00.006065",
                "successful": true
    },
                  "10.26.86.19": {
                individualResponseTimes: [
                    "00:00:00.005988",
                    "00:00:00.006948",
                    "00:00:00.005981",
                    "00:00:00.005964",
                    "00:00:00.005942"
                 ],
                individualStatus: [
                           "true",
                           "true",
                     true,
                     true,
                     true
                 ],
                 responseTime: "00:00:00.006165",
                 successful: true,
           },
                     "10.26.86.20": {
                individualResponseTimes: [
                     "00:00:00.005926",
                     "00:00:00.006072",
                     "00:00:00.005675",
                     "00:00:00.009904",
                     "00:00:00.006225"
                 ],
                         "individualStatus": [
                     true,
                     true,
                     true,
                     true,
                     true
```

```
                    ],
                     responseTime: "00:00:00.006760",
                     successful: true
                   }
              },
      "duration": "00:00:00.595982",
      "result": "Passed"
   }
 }
```

New since version

9.6

TestNetworkConfig

You can use the `TestNetworkConfig` method to test that the configured network settings match the network settings being used on the system.

Parameters

When you configure a node with the SetNetworkConfig method, in the UI or TUI, the configuration is validated and stored. The TestNetworkConfig API test uses the stored configuration for post-validation logic. For example, in the event of a power outage or network failure, you can use this API method to ensure a node is running with the most currently stored network configuration. This validates that there are no errors in the configuration and that the current configuration is in use.

This test is designed to only show failures in the response output. If there are no errors, this test does not return any output. See the following response examples.

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
details	Contains any errors found when validating the currently stored network settings with the running network configuration.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "TestNetworkConfig",
   "params": {},
   "id" : 1
}
```

Response example 1

If no errors are detected, then no responses are returned.

```
{
   "id" : 1,
   "result": {
   "details": {
   "network": {...}
  },
   "duration": "00:00:00.144514",
   "result": "Passed"
  }
}
```

Response example 2

Example of an MTU Mismatch.

```
{
  "id" : 1,
  "result":
  {
     "details" :
     {
        "error":
        {
           "message" : "Network configuration mismatch on Bond10G:
Incorrect MTU expectedMTU=[1500]  actualMTU=[9600]", name:
"xAssertionFailure"
        }
     },
     "duration": "0.125213sec",
     "result": "Failed"
  }
}
```

Response example 3

Example of a missing static route.

```
{
  "id": 1,
  "result":
  {
     "details" :
     {
     "error":
        {
        "message" : "Network configuration mismatch on Bond1G: Routing
table missing route=[192.168.137.2 via 192.168.159.254 dev Bond1G]", name:
"xAssertionFailure"
        }
     },
     "duration" : "0.128547sec",
     "result" : "Failed"
  }
}
```

New since version

9.6

Find more information

SetNetworkConfig

TestPing

You can use the `TestPing` method to test network connectivity to all nodes in the cluster on both 1G and 10G interfaces using ICMP packets. The test uses the appropriate MTU sizes for each packet based on the MTU settings in the network configuration. `TestPing` does not create a temporary VLAN interface.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
attempts	Specifies the number of times the system should repeat the test ping.	integer	5	No

Name	Description	Type	Default value	Required
hosts	Specifies a comma-separated list of addresses or hostnames of devices to ping. If no hosts are specified, the method pings the hosts in the storage cluster.	string	None	No
interface	The existing (base) interface from which the pings should be sent. Possible values: • `Bond10G`: Send pings from the Bond10G interface. • `Bond1G`: Send pings from the Bond1G interface.	string	None	No
packetSize	Specifies the number of bytes to send in the ICMP packet that is sent to each IP. The number of bytes must be less than the maximum MTU specified in the network configuration.	integer	None	No
pingTimeoutMsec	Specifies the number of milliseconds to wait for each individual ping response.	integer	500 milliseconds	No
prohibitFragmentation	Enables the DF (Do not Fragment) flag for the ICMP packets.	boolean	false	No

Name	Description	Type	Default value	Required
sourceAddressV4	The source IPv4 address to use in the ICMP ping packets.	string	None	No
sourceAddressV6	The source IPv6 address to use in the ICMP ping packets.	string	None	No
totalTimeoutSec	Specifies the time in seconds the ping should wait for a system response before issuing the next ping attempt or ending the process.	integer	5	No
virtualNetworkTag	The VLAN ID to use when sending the ping packets.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
details	List of each IP the node was able to communicate with and ping response statistics.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
    "method": "TestPing",
    "params": {
        "interface": "Bond1G",
        "hosts": "192.168.0.1"
    },
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
    "details": {
      "192.168.0.1": {
        "individualResponseCodes": [
          "Success",
          "Success",
          "Success",
          "Success",
          "Success"
        ],
        "individualResponseTimes": [
          "00:00:00.000304",
          "00:00:00.000123",
          "00:00:00.000116",
          "00:00:00.000113",
          "00:00:00.000111"
        ],
        "individualStatus": [
          true,
          true,
          true,
          true,
          true
        ],
        "interface": "Bond1G",
        "responseTime": "00:00:00.000154",
        "sourceAddressV4": "192.168.0.5",
        "successful": true
      }
    },
    "duration": "00:00:00.001747",
    "result": "Passed"
  }
}
```


New since version

5.0

TestRemoteConnectivity

You can use `TestRemoteConnectivity` method to ping each node of the remote cluster and check the remote ensemble database connection. Clusters must be paired in order to return useful results with this method. If the remote database connection fails, the response from the system lists the exceptions.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
details	Individual ping response times for each node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "TestRemoteConnectivity",
    "params": {
      "force": "true"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": null,
  "result": {
    "details": {
      "1": {
        "details": {
          "10.26.86.17": {
            "individualResponseTimes": [
              "00:00:00.006868",
              "00:00:00.005933",
              "00:00:00.006655",
```

```
            "00:00:00.006584",
            "00:00:00.006334"
          ],
          "individualStatus": [
            "true",
            "true",
            "true",
            "true",
            "true"
          ],
          "responseTime": "00:00:00.006475",
          "successful": true
        },
        "10.26.86.18": {
          "individualResponseTimes": [
            "00:00:00.006201",
            "00:00:00.006187",
            "00:00:00.005990",
            "00:00:00.006029",
            "00:00:00.005917"
          ],
          "individualStatus": [
            "true",
            "true",
            "true",
            "true",
            "true"
          ],
          "responseTime": "00:00:00.006065",
          "successful": true
        },
        "10.26.86.19": {
          "individualResponseTimes": [
            "00:00:00.005988",
            "00:00:00.006948",
            "00:00:00.005981",
            "00:00:00.005964",
            "00:00:00.005942"
          ],
          "individualStatus": [
            "true",
            "true",
            "true",
            "true",
            "true"
          ],
```

```
            "responseTime": "00:00:00.006165",
            "successful": true,
          },
          "10.26.86.20": {
            "individualResponseTimes": [
              "00:00:00.005926",
              "00:00:00.006072",
              "00:00:00.005675",
              "00:00:00.009904",
              "00:00:00.006225"
            ],
            "individualStatus": [
              "true",
              "true",
              "true",
              "true",
              "true"
            ],
            "responseTime": "00:00:00.006760",
            "successful": true
          }
        },
          "successful": true
      }
    },
  "duration": "00:00:00.595982",
  "result": "Passed"
  }
}
```


New since version

9.6

Replication API methods

Replication API methods enable you to connect two clusters for continuous data protection (CDP). When you connect two clusters, active volumes within a cluster can be continuously replicated to a second cluster to provide data recovery. By pairing volumes for replication, you can protect your data from events that might render it inaccessible.

- Cluster pairing order of operations
- Volume pairing order of operations
- Supported modes of replication for paired clusters
- CompleteClusterPairing

- CompleteVolumePairing
- ListClusterPairs
- ListActivePairedVolumes
- ModifyVolumePair
- RemoveClusterPair
- RemoveVolumePair
- StartClusterPairing
- StartVolumePairing

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Cluster pairing order of operations

You must establish a connection between a pair of storage clusters running Element software before remote replication can be used.

Use the following set of API methods to establish a cluster connection:

- StartClusterPairing:

  This API method creates and returns a pairing key that is used to establish a cluster pair. The key is encoded and contains information that is used to establish communications between clusters. A single cluster can be paired with up to four other clusters. However, a new key must be generated for each cluster pairing. The StartClusterPairing method generates a new key each time the method is called. Use each unique key with the CompleteClusterPairing method to pair each additional cluster.

  > For security reasons, the pairing key should not be sent to other users via email. The key contains a user name and password.

- CompleteClusterPairing:

  This method uses the pairing key created with the StartClusterPairing API method to create a cluster pair. Issue the CompleteClusterPairing API method with the clusterPairingKey parameter to the destination. The origination cluster is the cluster that created the key.

Find more information

- StartClusterPairing
- CompleteClusterPairing

Volume pairing order of operations

You must create a cluster pair between two corresponding clusters before volumes can be paired.

Use the following set of API methods to establish a cluster connection:

- StartVolumePairing:

  This API method creates and returns a volume pairing key that is used to create a volume pair. The key contains information that is used to establish communications between volumes.

- CompleteVolumePairing:

  This method uses the pairing key created with the StartVolumePairing API method to create a volume pair. Issue the CompleteVolumePairing API method with the volumeID and volumePairingKey parameters to the destination volume.

Only one of the paired volumes can be identified as a replication target volume. Use the ModifyVolumePair API method to establish the direction of the volume's data replication by identifying which volume is the target. Data is replicated from the source volume to the target volume.

Find more information

- StartVolumePairing
- CompleteVolumePairing
- ModifyVolumePair

Supported modes of replication for paired clusters

The following modes of replication are supported on the paired clusters:

- Asynchronous replication of data: The data sent to the replication target volume is sent asynchronously. The system does not wait for an acknowledgment to be sent before writing data.
- Synchronous replication of data: The data sent to the replication target volume is sent synchronously. When the I/O operations sent from the host are acknowledged by the system, the system acknowledgment is sent back to the host and the data is sent to the replication target volume.
- Snapshots-only replication of data: Only volume snapshots are replicated to the target cluster.

CompleteClusterPairing

The `CompleteClusterPairing` method is the second step in the cluster pairing process. Use this method with the encoded key received from the `StartClusterPairing` method to complete the cluster pairing process.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
clusterPairingKey	A string of characters that is returned from the StartClusterPairing API method.	string	None	Yes

Return value

This method has the following return value:

Name	Description	Type
clusterPairID	Unique identifier for the cluster pair.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CompleteClusterPairing",
   "params": {
        "clusterPairingKey" :
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223
a2231636561313336322d346338662d343631612d626537322d373435363661393533643263
6222c22636c7573746572556e697175654944223a2278736d36222c226d766970223a22313
9322e3136382e3133392e313232222c226e616d65223a224175746f5465737432 2d6330755
2222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222c227270634
36f6e6e656374696f6e4944223a393133313432363439 2c22757365726e616d65223a225f5
f53465f706169725f50597a796647704c7246564432444a42227d"
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
       "clusterPairID" : 1
   }
}
```

New since version

9.6

Find more information

StartClusterPairing

CompleteVolumePairing

You can use `CompleteVolumePairing` to complete the pairing of two volumes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeID	The ID of volume that will complete the volume pair.	integer	None	Yes
volumePairingKey	The key returned from the StartVolumePairing API method.	string	None	Yes

Return value

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CompleteVolumePairing",
   "params": {
        "volumeID" : 12,
        "volumePairingKey" :
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223
a2231636561313336322d346338662d343631612d626537322d373435363661393536432
6222c22636c7573746572556e697175654944223a2278736d36222c226d766970223a22313
9322e3136382e3133392e313232222c226e616d65223a224175746f54657374322d6330755
2222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222c227270634
36f6e6e656374696f6e4944223a393133313432363439 2c22757365726e616d65223a225f5
f53465f706169725f50597a796647704c7246564432444a42227d"
        },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
"result": {}
}
```

New since version

9.6

Find more information

StartVolumePairing

ListClusterPairs

You can use the `ListClusterPairs` method to list all clusters that are paired with the current cluster. This method returns information about active and pending cluster pairings, such as statistics about the current pairing as well as the connectivity and latency (in milliseconds) of the cluster pairing.

Parameter

This method has no input parameter:

Return value

This method has the following return value:

Name	Description	Type
clusterPairs	Information about each paired cluster.	clusterPair array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListClusterPairs",
   "params": {
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
 "id": 1,
 "result": {
   "clusterPairs": [
     {
      "clusterName": "cluster2",
      "clusterPairID": 3,
      "clusterPairUUID": "9866fbeb-c2f8-4df3-beb9-58a5c4e49c9b",
      "clusterUUID": 5487,
      "latency": 1,
      "mvip": "172.1.1.5",
      "status": "Connected"
      "version": "8.0.0.1361"
     },
     {
      "clusterName": "cluster3",
      "clusterPairID": 2,
      "clusterPairUUID": "8132a699-ce82-41e0-b406-fb914f976042",
      "clusterUUID": 1383,
      "latency": 1,
      "mvip": "172.1.1.6",
      "status": "Connected"
      "version": "8.0.0.1361"
     }
   ]
 }
```


New since version

9.6

ListActivePairedVolumes

You can use the `ListActivePairedVolumes` method to list all of the active volumes paired with a volume. This method returns information about volumes with active and pending pairings.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
volumes	Volume information for the paired volumes.	volumePair array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListActivePairedVolumes",
   "params": {
      },
   "id" : 1
}
```

Response example

Responses for this method are similar to the following example:

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 1,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2016-06-24T15:21:59Z",
                "deleteTime": "",
                "enable512e": true,
                "iqn": "iqn.2010-01.com.solidfire:0oto.bk.24",
                "name": "BK",
                "purgeTime": "",
                "qos": {
                    "burstIOPS": 15000,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
```

```
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 15000,
                    "minIOPS": 50
                },
                "scsiEUIDeviceID": "306f746f00000018f47acc0100000000",
                "scsiNAADeviceID": "6f47acc100000000306f746f00000018",
                "sliceCount": 1,
                "status": "active",
                "totalSize": 10737418240,
                "virtualVolumeID": null,
                "volumeAccessGroups": [],
                "volumeID": 24,
                "volumePairs": [
                    {
                        "clusterPairID": 2,
                        "remoteReplication": {
                            "mode": "Async",
                            "pauseLimit": 3145728000,
                            "remoteServiceID": 14,
                            "resumeDetails": "",
                            "snapshotReplication": {
                                "state": "Idle",
                                "stateDetails": ""
                            },
                            "state": "Active",
                            "stateDetails": ""
                        },
                        "remoteSliceID": 8,
                        "remoteVolumeID": 8,
                        "remoteVolumeName": "PairingDoc",
                        "volumePairUUID": "229fcbf3-2d35-4625-865a-
d04bb9455cef"
                    }
                ]
            }
        ]
    }
}
```


New since version

9.6

ModifyVolumePair

You can use the `ModifyVolumePair` method to pause or restart replication between a pair of volumes. This method is set on the source volume (the volume with read/write access).

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeID	Identification number of the volume to be modified.	integer	None	Yes
pausedManual	Remote replication can be paused or restarted on the source (read/write) volume. Possible values: • true: Pause volume replication. • false: Restart volume replication. If no value is specified, no change in replication is performed.	boolean	None	No

mode	Volume replication mode. Possible values: • Async: Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. • Sync: The source acknowledges the write when the data is stored locally and on the remote cluster. • SnapshotsOnly: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.	string	None	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyVolumePair",
  "params": {
    "pausedManual": false,
    "volumeID": 5,
    "mode": "sync"
   },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {}
}
```

New since version

9.6

RemoveClusterPair

You can use the `RemoveClusterPair` method to close the open connections between two paired clusters.

Parameter

Before you remove a cluster pair, you must first remove all volume pairing to the clusters with the RemoveVolumePair API method.

This method has the following input parameter:

Name	Description	Type	Default value	Required
clusterPairID	Unique identifier used to pair two clusters.	integer	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveClusterPair",
   "params": {
               "clusterPairID": 1
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {}
}
```

New since version

9.6

RemoveVolumePair

You can use the `RemoveVolumePair` method to remove the remote pairing between two volumes. Use this method on both the source and target volumes that are paired together. When you remove the volume pairing information, data is no longer replicated to or from the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	ID of the volume on which to stop the replication process.	integer	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveVolumePair",
   "params": {
      "volumeID": 5
   "id" : 1
   }
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
   }
}
```

New since version

9.6

StartClusterPairing

You can use the `StartClusterPairing` method to create an encoded key from a cluster that is used to pair with another cluster. The key created from this API method is used in the `CompleteClusterPairing` method to establish a cluster pairing. You can pair a cluster with a maximum of four other clusters.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
clusterPairingKey	A string of characters that is used by the CompleteClusterPairing API method.	string
clusterPairID	Unique identifier for the cluster pair.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "StartClusterPairing",
   "params": {
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:


```
{
   "id": 1,
   "result": {
     "clusterPairID": 1,
     "clusterPairingKey":
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223
a2231636561313336322d346338662d343631612d626537322d373435363661393533643263
6222c22636c7573746572556e697175654944223a2278736d36222c226d766970223a22313
9322e3136382e3133392e313232222c226e616d65223a224175746f54657374322d6330755
2222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222c227270634
36f6e6e656374696f6e4944223a393133313432363439 2c22757365726e616d65223a225f5
f53465f706169725f50597a796647704c7246564432444a42227d"
   }
}
```


New since version

9.6

Find more information

CompleteClusterPairing

StartVolumePairing

You can use the `StartVolumePairing` method to create an encoded key from a volume that is used to pair with another volume. The key that this method creates is used in the `CompleteVolumePairing` method to establish a volume pairing.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
mode	The mode of the volume on which to start the pairing process. The mode can only be set if the volume is the source volume. Possible values: • `Async`: Writes are acknowledged when they complete locally. The cluster does not wait for writes to be replicated to the target cluster. (Default if no mode parameter specified.) • `Sync`: Source acknowledges write when the data is stored locally and on the remote cluster. • `SnapshotsOnly`: Only snapshots created on the source cluster are replicated. Active writes from the source volume are not replicated.	string	None	No
volumeID	The ID of the volume on which to start the pairing process.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumePairingKey	A string of characters that is used by the CompleteVolumePairing API method.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "StartVolumePairing",
   "params": {
         "mode": "Async",
     "volumeID" : 14
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "volumePairingKey" :
"7b226d766970223a223139322e3136382e3133392e313232222c22766f6c756d654944223
a312c22766f6c756d654e616d65223a2254657374222c22766f6c756d65506169725555494
4223a22363936323466636 22d323032652d343332352d613536392d6563396336353563376
23561227d"
      }
}
```

New since version

9.6

Find more information

CompleteVolumePairing

Security API methods

You can integrate Element software with external security-related services, such as an external key management server. These security-related methods enable you to

configure Element security features such as external key management for Encryption at Rest.

- AddKeyServerToProviderKmip
- CreateKeyProviderKmip
- CreateKeyServerKmip
- CreatePublicPrivateKeyPair
- DeleteKeyProviderKmip
- DeleteKeyServerKmip
- DisableEncryptionAtRest
- EnableEncryptionAtRest
- GetClientCertificateSignRequest
- GetKeyProviderKmip
- GetKeyServerKmip
- ListKeyProvidersKmip
- ListKeyServersKmip
- ModifyKeyServerKmip
- RemoveKeyServerFromProviderKmip
- SignSshKeys
- TestKeyProviderKmip
- TestKeyServerKmip

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddKeyServerToProviderKmip

You can use the `AddKeyServerToProviderKmip` method to assign a Key Management Interoperability Protocol (KMIP) key server to the specified key provider. During assignment, the server is contacted to verify functionality. If the specified key server is already assigned to the specified key provider, no action is taken and no error is returned. You can remove the assignment using the `RemoveKeyServerFromProviderKmip` method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderID	The ID of the key provider to assign the key server to.	integer	None	Yes
keyServerID	The ID of the key server to assign.	integer	None	Yes

Return values

This method has no return value. The assignment is considered successful as long as there is no error returned.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "AddKeyServerToProviderKmip",
  "params": {
    "keyProviderID": 1,
    "keyServerID": 15
    },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {}
    }
}
```

New since version

11.7

CreateKeyProviderKmip

You can use the `CreateKeyProviderKmip` method to create a Key Management Interoperability Protocol (KMIP) key provider with the specified name. A key provider

defines a mechanism and location to retrieve authentication keys. When you create a new KMIP key provider, it does not have any KMIP key servers assigned to it. To create a KMIP key server, use the `CreateKeyServerKmip` method. To assign it to a provider, see `AddKeyServerToProviderKmip`.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderName	The name to associate with the created KMIP key provider. This name is only used for display purposes and does not need to be unique.	string	None	Yes

Return values

This method has the following return values:

Name	Description	Type
kmipKeyProvider	An object containing details about the newly created key provider.	KeyProviderKmip

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CreateKeyProviderKmip",
  "params": {
    "keyProviderName": "ProviderName",
    },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
    "kmipKeyProvider": {
        "keyProviderName": "ProviderName",
        "keyProviderIsActive": true,
        "kmipCapabilities": "SSL",
        "keyServerIDs": [
            15
        ],
        "keyProviderID": 1
    }
}
}
```

New since version

11.7

CreateKeyServerKmip

You can use the `CreateKeyServerKmip` method to create a Key Management Interoperability Protocol (KMIP) key server with the specified attributes. During creation, the server is not contacted; it does not need to exist before you use this method. For clustered key server configurations, you must provide the hostnames or IP addresses of all server nodes in the kmipKeyServerHostnames parameter. You can use the `TestKeyServerKmip` method to test a key server.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
kmipCaCertificate	The public key certificate of the external key server's root CA. This will be used to verify the certificate presented by external key server in the TLS communication. For key server clusters where individual servers use different CAs, provide a concatenated string containing the root certificates of all the CAs.	string	None	Yes
kmipClientCertificate	A PEM format Base64 encoded PKCS#10 X.509 certificate used by the Solidfire KMIP client.	string	None	Yes
kmipKeyServerHostnames	Array of the hostnames or IP addresses associated with this KMIP key server. Multiple hostnames or IP addresses must only be provided if the key servers are in a clustered configuration.	string array	None	Yes
kmipKeyServerName	The name of the KMIP key server. This name is only used for display purposes and does not need to be unique.	string	None	Yes
kmipKeyServerPort	The port number associated with this KMIP key server (typically 5696).	integer	None	No

Return values

This method has the following return values:

Name	Description	Type
kmipKeyServer	An object containing details about the newly created key server.	KeyServerKmip

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CreateKeyServerKmip",
  "params": {
    "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
    "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
    "kmipKeyServerHostnames" : ["server1.hostname.com",
"server2.hostname.com"],
    "kmipKeyServerName" : "keyserverName",
    "kmipKeyServerPort" : 5696
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
        "kmipKeyServer": {
            "kmipCaCertificate":"MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
            "kmipKeyServerHostnames":[
                "server1.hostname.com", "server2.hostname.com"
            ],
            "keyProviderID":1,
            "kmipKeyServerName":"keyserverName",
            "keyServerID":1
            "kmipKeyServerPort":1,
            "kmipClientCertificate":"dKkkirWmnWXbj9T/UWZYB2oK0z5...",
            "kmipAssignedProviderIsActive":true
        }
     }
}
```


New since version

11.7

CreatePublicPrivateKeyPair

You can use the `CreatePublicPrivateKeyPair` method to create public and private SSL keys. You can use these keys to generate certificate signing requests. There can only be one key pair in use for each storage cluster. Before using this method to replacing existing keys, ensure the keys are no longer in use by any providers.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
commonName	The X.509 distinguished name **Common Name** field (CN).	string	None	No
country	The X.509 distinguished name **Country** field ©.	string	None	No

Name	Description	Type	Default value	Required
emailAddress	The X.509 distinguished name **Email Address** field (MAIL).	string	None	No
locality	The X.509 distinguished name **Locality Name** field (L).	string	None	No
organization	The X.509 distinguished name **Organization Name** field (O).	string	None	No
organizationalUnit	The X.509 distinguished name **Organizational Unit Name** field (OU).	string	None	No
state	The X.509 distinguished name **State** or **Province Name** field (ST or SP or S).	string	None	No

Return values

This method has no return values. If there is no error, key creation is considered successful.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CreatePublicPrivateKeyPair",
  "params": {
    "commonName": "Name",
    "country": "US",
    "emailAddress" : "email@domain.com"
   },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {}
}
```

New since version

11.7

DeleteKeyProviderKmip

You can use the `DeleteKeyProviderKmip` method to delete the specified inactive Key Management Interoperability Protocol (KMIP) key provider.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderID	The ID of the key provider to delete.	integer	None	Yes

Return values

This method has no return values. The delete operation is considered successful as long is there is no error.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DeleteKeyProviderKmip",
  "params": {
    "keyProviderID": "1"
    },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {}
}
```

New since version

11.7

DeleteKeyServerKmip

You can use the `DeleteKeyServerKmip` method to delete an existing Key Management Interoperability Protocol (KMIP) key server. You can delete a key server unless it is the last one assigned to its provider, and that provider is providing keys which are currently in use.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyServerID	The ID of the KMIP key server to delete.	integer	None	Yes

Return values

This method has the no return values. The delete operation is considered successful if there are no errors.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DeleteKeyServerKmip",
  "params": {
     "keyServerID": 15
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {}
}
```

New since version

11.7

DisableEncryptionAtRest

You can use the `DisableEncryptionAtRest` method to remove the encryption that was previously applied to the cluster using the `EnableEncryptionAtRest` method. This disable method is asynchronous and returns a response before encryption is disabled. You can use the `GetClusterInfo` method to poll the system to see when the process has completed.

To see the current status of encryption at rest and/or software encryption at rest on the cluster, use the get cluster info method. You can use the `GetSoftwareEncryptionAtRestInfo` method to get information the cluster uses to encrypt data at rest.

You cannot use this method to disable software encryption at rest. To disable software encryption at rest, you need to create a new cluster with software encryption at rest disabled.

Parameters

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DisableEncryptionAtRest",
   "params": {},
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
  "result" : {}
}
```

New since version

9.6

Find more information

- GetClusterInfo
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

EnableEncryptionAtRest

You can use the `EnableEncryptionAtRest` method to enable the Advanced Encryption Standard (AES) 256-bit encryption at rest on the cluster so that the cluster can manage the encryption key used for the drives on each node. This feature is not enabled by default.

To see the current status of encryption at rest and/or software encryption at rest on the cluster, use the get cluster info method. You can use the `GetSoftwareEncryptionAtRestInfo` method to get information the cluster uses to encrypt data at rest.

This method does not enable software encryption at rest. This can only be done using the create cluster method with `enableSoftwareEncryptionAtRest` set to `true`.

When you enable encryption at rest, the cluster automatically manages encryption keys internally for the drives on each node in the cluster.

If a keyProviderID is specified, the password is generated and retrieved according to the type of key provider. This is usually done using a Key Management Interoperability Protocol (KMIP) key server in the case of a KMIP key provider. After this operation, the specified provider is considered active and cannot be deleted until Encryption at Rest is disabled using the `DisableEncryptionAtRest` method.

If you have a node type with a model number ending in "-NE", the `EnableEncryptionAtRest` method call will fail with a response of "Encryption not allowed. Cluster detected non-encryptable node".

You should only enable or disable encryption when the cluster is running and in a healthy state. You can enable or disable encryption at your discretion and as often as you need.

This process is asynchronous and returns a response before encryption is enabled. You can use the `GetClusterInfo` method to poll the system to see when the process has completed.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderID	The ID of a KMIP key provider to use.	integer	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "EnableEncryptionAtRest",
   "params": {},
   "id": 1
}
```

Response examples

This method returns a response similar to the following example from the EnableEncryptionAtRest method. There is no result to report.

```
{
   "id": 1,
   "result": {}
}
```

While Encryption At Rest is being enabled on a cluster, GetClusterInfo returns a result describing the state of Encryption at Rest ("encryptionAtRestState") as "enabling". After Encryption at Rest is fully enabled, the returned state changes to "enabled".

```
{
   "id": 1,
      "result": {
         "clusterInfo": {
            "attributes": { },
               "encryptionAtRestState": "enabling",
            "ensemble": [
               "10.10.5.94",
               "10.10.5.107",
               "10.10.5.108"
            ],
            "mvip": "192.168.138.209",
            "mvipNodeID": 1,
            "name": "Marshall",
            "repCount": 2,
            "svip": "10.10.7.209",
            "svipNodeID": 1,
            "uniqueID": "91dt"
      }
   }
}
```


New since version

9.6

Find more information

- SecureEraseDrives
- GetClusterInfo
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

GetClientCertificateSignRequest

You can use the `GetClientCertificateSignRequest` method to generate a certificate signing request that can be signed by a certificate authority to generate a client certificate for the cluster. Signed certificates are needed to establish a trust relationship for interacting with external services.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
clientCertificateSignRequest	A PEM format Base64 encoded PKCS#10 X.509 client certificate sign request.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetClientCertificateSignRequest",
  "params": {
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
        "clientCertificateSignRequest":
"MIIByjCCATMCAQAwgYkxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybm..."
   }
}
```

New since version

11.7

GetKeyProviderKmip

You can use the `GetKeyProviderKmip` method to retrieve information about the specified Key Management Interoperability Protocol (KMIP) key provider.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderID	The ID of the KMIP key provider object to return.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
kmipKeyProvider	An object containing details about the requested key provider.	KeyProviderKmip

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetKeyProviderKmip",
  "params": {
    "keyProviderID": 15
    },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
    {
    "kmipKeyProvider": {
        "keyProviderID": 15,
        "kmipCapabilities": "SSL",
        "keyProviderIsActive": true,
        "keyServerIDs": [
            1
        ],
        "keyProviderName": "ProviderName"
    }
   }
}
```

New since version

11.7

GetKeyServerKmip

You can use the `GetKeyServerKmip` method to return information about the specified Key Management Interoperability Protocol (KMIP) key server.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyServerID	The ID of the KMIP key server to return information about.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
kmipKeyServer	An object containing details about the requested key server.	KeyServerKmip

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetKeyServerKmip",
  "params": {
     "keyServerID": 15
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
        "kmipKeyServer": {
            "kmipCaCertificate":"MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
            "kmipKeyServerHostnames":[
                "server1.hostname.com", "server2.hostname.com"
            ],
            "keyProviderID":1,
            "kmipKeyServerName":"keyserverName",
            "keyServerID":15
            "kmipKeyServerPort":1,
            "kmipClientCertificate":"dKkkirWmnWXbj9T/UWZYB2oK0z5...",
            "kmipAssignedProviderIsActive":true
        }
    }
}
```

New since version

11.7

GetSoftwareEncryptionAtRestInfo

You can use the `GetSoftwareEncryptionAtRestInfo` method to get software encryption-at-rest information the cluster uses to encrypt data at rest.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Parameter	Description	Type	Optional
masterKeyInfo	Information about the current software encryption-at-rest master key.	EncryptionKeyInfo	True
rekeyMasterKeyAsyncResultID	The async result ID of the current or most recent rekey operation (if any), if it has not been deleted yet. `GetAsyncResult` output will include a `newKey` field that contains information about the new master key and a `keyToDecommission` field that contains information about the old key.	integer	True
state	The current software encryption-at-rest state. Possible values are `disabled` or `enabled`.	string	False
version	A version number that is incremented each time software encryption at rest is enabled.	integer	False

Request example

Requests for this method are similar to the following example:

```
{
 "method": "getsoftwareencryptionatrestinfo"
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "masterKeyInfo": {
            "keyCreatedTime": "2021-09-20T23:15:56Z",
            "keyID": "4d80a629-a11b-40ab-8b30-d66dd5647cfd",
            "keyManagementType": "internal"
        },
        "state": "enabled",
        "version": 1
    }
}
```


New since version

12.3

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

ListKeyProvidersKmip

You can use the `ListKeyProvidersKmip` method to retrieve a list of all existing Key Management Interoperability Protocol (KMIP) key providers. You can filter the list by specifying additional parameters.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderIsActive	Filters returned KMIP key server objects based on whether they are active. Possible values: • true: Returns only KMIP key providers which are active (providing keys which are currently in use). • false: Returns only KMIP key providers which are inactive (not providing any keys and able to be deleted). If omitted, returned KMIP key providers are not filtered based on whether they are active.	boolean	None	No

Name	Description	Type	Default value	Required
kmipKeyProviderHasServerAssigned	Filters returned KMIP key providers based on whether they have a KMIP key server assigned. Possible values: • true: Returns only KMIP key providers which have a KMIP key server assigned. • false: Returns only KMIP key providers which do not have a KMIP key server assigned. If omitted, returned KMIP key providers are not filtered based on whether they have a KMIP key server assigned.	boolean	None	No

Return values

This method has the following return values:

Name	Description	Type
kmipKeyProviders	A list of KMIP key providers that have been created.	KeyProviderKmip array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListKeyProvidersKmip",
  "params": {},
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
    {
    "kmipKeyProviders": [
        {
            "keyProviderID": 15,
            "kmipCapabilities": "SSL",
            "keyProviderIsActive": true,
            "keyServerIDs": [
                1
            ],
            "keyProviderName": "KeyProvider1"
        }
    ]
}
}
```

New since version

11.7

ListKeyServersKmip

You can use the `ListKeyServersKmip` method to list all Key Management Interoperability Protocol (KMIP) key servers that have been created. You can filter the results by specifying additional parameters.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderID	When specified, the method only returns KMIP key servers that are assigned to the specified KMIP key provider. If omitted, returned KMIP key servers will not be filtered based on whether they are assigned to the specified KMIP Key Provider.	integer	None	No
kmipAssignedProviderIsActive	Filters returned KMIP key server objects based on whether they are active. Possible values: • true: Returns only KMIP key servers which are active (providing keys which are currently in use). • false: Returns only KMIP key servers which are inactive (not providing any keys and able to be deleted). If omitted, returned KMIP key servers are not filtered based on whether they are active.	boolean	None	No

Name	Description	Type	Default value	Required
kmipHasProviderAssigned	Filters returned KMIP key servers based on whether they have a KMIP key provider assigned. Possible values: • true: Returns only KMIP key servers which have a KMIP key provider assigned. • false: Returns only KMIP key servers which do not have a KMIP key provider assigned. If omitted, returned KMIP key servers are not filtered based on whether they have a KMIP key provider assigned.	boolean	None	No

Return values

This method has the following return values:

Name	Description	Type
kmipKeyServers	The complete list of KMIP key servers which have been created.	KeyServerKmip array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListKeyServersKmip",
  "params": {},
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "kmipKeyServers": [
        {
            "kmipKeyServerName": "keyserverName",
            "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
            "keyServerID": 15,
            "kmipAssignedProviderIsActive": true,
            "kmipKeyServerPort": 5696,
            "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
            "kmipKeyServerHostnames": [
                "server1.hostname.com", "server2.hostname.com"
            ],
            "keyProviderID": 1
        }
    ]
}
```

New since version

11.7

ModifyKeyServerKmip

You can use the `ModifyKeyServerKmip` method to modify an existing Key Management Interoperability Protocol (KMIP) key server to the specified attributes. Although the only required parameter is the keyServerID, a request containing only the keyServerID will take no action and return no error. Any other parameters you specify will replace the existing values for the key server with the specified keyServerID. The key server is contacted during the operation to ensure that it is functional. You can provide multiple hostnames or IP addresses with the kmipKeyServerHostnames parameter, but only if the key servers are in a clustered configuration.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyServerID	The ID of the KMIP Key Server to modify.	integer	None	Yes

kmipCaCertificate	The public key certificate of the external key server's root CA. This will be used to verify the certificate presented by external key server in the TLS communication. For key server clusters where individual servers use different CAs, provide a concatenated string containing the root certificates of all the CAs.	string	None	No
kmipClientCertificate	A PEM format Base64 encoded PKCS#10 X.509 certificate used by the Solidfire KMIP client.	string	None	No
kmipKeyServerHostnames	Array of the hostnames or IP addresses associated with this KMIP key server. Multiple hostnames or IP addresses must only be provided if the key servers are in a clustered configuration.	string array	None	No
kmipKeyServerName	The name of the KMIP key server. This name is only used for display purposes and does not need to be unique.	string	None	No
kmipKeyServerPort	The port number associated with this KMIP key server (typically 5696).	integer	None	No

Return values

This method has the following return values:

Name	Description	Type
kmipKeyServer	An object containing details about the newly modified key server.	KeyServerKmip

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyKeyServerKmip",
  "params": {
    "keyServerID": 15
    "kmipCaCertificate": "CPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
    "kmipClientCertificate": "kirWmnWXbj9T/UWZYB2oK0z5...",
    "kmipKeyServerHostnames" : ["server1.hostname.com",
"server2.hostname.com"],
    "kmipKeyServerName" : "keyserverName",
    "kmipKeyServerPort" : 5696
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
        "kmipKeyServer": {
            "kmipCaCertificate":"CPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
            "kmipKeyServerHostnames":[
                "server1.hostname.com", "server2.hostname.com"
            ],
            "keyProviderID":1,
            "kmipKeyServerName":"keyserverName",
            "keyServerID":1
            "kmipKeyServerPort":1,
            "kmipClientCertificate":"kirWmnWXbj9T/UWZYB2oK0z5...",
            "kmipAssignedProviderIsActive":true
        }
     }
}
```


New since version

11.7

RekeySoftwareEncryptionAtRestMasterKey

You can use the `RekeySoftwareEncryptionAtRestMasterKey` method to rekey the software encryption-at-rest master key used to encrypt DEKs (Data Encryption Keys). During cluster creation, software encryption at rest is configured to use Internal Key Management (IKM). This rekey method can be used after cluster creation to use either IKM or External Key Management (EKM).

Parameters

This method has the following input parameters. If the `keyManagementType` parameter is not specified, the rekey operation is performed using the existing key management configuration. If the `keyManagementType` is specified and the key provider is external, the `keyProviderID` parameter must also be used.

Parameter	Description	Type	Optional
keyManagementType	The type of key management used to manage the master key. Possible values are: `Internal`: Rekey using internal key management. `External`: Rekey using external key management. If this parameter is not specified, the rekey operation is performed using the existing key management configuration.	string	True
keyProviderID	The ID of the key provider to use. This is a unique value returned as part of one of the `CreateKeyProvider` methods. The ID is only required when `keyManagementType` is `External` and is otherwise invalid.	integer	True

Return values

This method has the following return values:

Parameter	Description	Type	Optional
asyncHandle	Determine the status of the rekey operation using this `asyncHandle` value with `GetAsyncResult`. `GetAsyncResult` output will include a `newKey` field that contains information about the new master key and a `keyToDecommission` field that contains information about the old key.	integer	False

Request example

Requests for this method are similar to the following example:

```
{
 "method": "rekeysoftwareencryptionatrestmasterkey",
 "params": {
   "keyManagementType": "external",
   "keyProviderID": "<ID number>"
 }
}
```

Response example

This method returns a response similar to the following example:

```
{
    "asyncHandle": 1
}
```

New since version

12.3

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

RemoveKeyServerFromProviderKmip

You can use the `RemoveKeyServerFromProviderKmip` method to unassign the specified Key Management Interoperability Protocol (KMIP) key server from the provider it was assigned to. You can unassign a key server from its provider unless it is the last one and its provider is active (providing keys which are currently in use). If the specified key server is not assigned to a provider, no action is taken and no error is returned.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyServerID	The ID of the KMIP key server to unassign.	integer	None	Yes

Return values

This method has no return values. The removal is considered successful as long as no error is returned.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "RemoveKeyServerFromProviderKmip",
  "params": {
    "keyServerID": 1
   },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {}
    }
}
```

New since version

11.7

SignSshKeys

After SSH is enabled on the cluster using the EnableSSH method, you can use the `SignSshKeys` method to gain access to a shell on a node.

Beginning with Element 12.5, `sfreadonly` is a new system account allows for basic troubleshooting on a node. This API enables SSH access using the `sfreadonly` system account across all nodes in the cluster.

Unless advised by NetApp Support, any alterations to the system are unsupported, voiding your support contract, and may result in instability or inaccessibility of data.

After you use the method, you must copy the keychain from the response, save it to the system that will be initiating the SSH connection, then run the following command:

```
ssh -i <identity_file> sfreadonly@<node_ip>
```

`identity_file` is a file from which the identity (private key) for public key authentication is read and `node_ip` is the IP address of the node. For more information on `identity_file`, see the SSH man page.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
duration	Integer from 1 to 24 reflecting number of hours for signed key to be valid. If duration is not specified, the default is used.	integer	1	No
publicKey	If provided, this parameter will only return the signed_public_key instead of creating a full keychain to the user.  Public keys submitted using the URL bar in a browser with + are interpreted as spaced and break signing.	string	Null	No
sfadmin	Allows access to the sfadmin shell account when you make the API call with supportAdmin cluster access, or when the node is not in a cluster.	boolean	False	No

Return values

This method has the following return values:

Name	Description	Type
keygen_status	Contains the identity in the signed key, the principals allowed, and the valid start and end dates for the key.	string
private_key	A private SSH key value is only returned if the API is generating a complete keychain for the end user. The value is Base64 encoded; you must decode the value when it is written to a file to ensure that it is read as a valid private key.	string
public_key	A public SSH key value is only returned if the API is generating a complete keychain for the end user. When you pass a public_key parameter to the API method, only the `signed_public_key` value is returned in the response.	string
signed_public_key	The SSH public key that results from signing the public key, whether this was user provided or generated by API.	string

Request example

Requests for this method are similar to the following example:

```
{
    "method": "SignSshKeys",
    "params": {
        "duration": 2,
        "publicKey":<string>
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": null,
  "result": {
    "signedKeys": {
      "keygen_status": <keygen_status>,
      "signed_public_key": <signed_public_key>
    }
  }
}
```

In this example, a public key is signed and returned that is valid for the duration (1-24 hours).

New since version

12.5

TestKeyProviderKmip

You can use the `TestKeyProviderKmip` method to test whether the specified Key Management Interoperability Protocol (KMIP) key provider is reachable and functioning normally.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyProviderID	The ID of the key provider to test.	integer	None	Yes

Return values

This method has no return values. The test is considered successful as long as no error is returned.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "TestKeyProviderKmip",
  "params": {
    "keyProviderID": 15
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {}
    }
}
```

New since version

11.7

TestKeyServerKmip

You can use the `TestKeyServerKmip` method to test whether the specified Key Management Interoperability Protocol (KMIP) key server is reachable and functioning normally.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
keyServerID	The ID of the KMIP key server to test.	integer	None	Yes

Return values

This method has no return values. The test is considered successful if there are no errors returned.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "TestKeyServerKmip",
  "params": {
    "keyServerID": 15
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {}
    }
```

New since version

11.7

SnapMirror API methods

SnapMirror API methods are used by the Element web UI for managing snapshots mirrored with remote ONTAP systems. These methods are meant for use by the Element web UI only. If you need API access to SnapMirror functionality, use the ONTAP APIs. Request and return examples are not provided for SnapMirror API methods.

- AbortSnapMirrorRelationship
- BreakSnapMirrorRelationship
- BreakSnapMirrorVolume
- CreateSnapMirrorEndpoint
- CreateSnapMirrorEndpointUnmanaged
- CreateSnapMirrorRelationship
- CreateSnapMirrorVolume

- DeleteSnapMirrorEndpoints
- DeleteSnapMirrorRelationships
- GetOntapVersionInfo
- GetSnapMirrorClusterIdentity
- InitializeSnapMirrorRelationship
- ListSnapMirrorAggregates
- ListSnapMirrorEndpoints
- ListSnapMirrorLuns
- ListSnapMirrorNetworkInterfaces
- ListSnapMirrorNodes
- ListSnapMirrorPolicies
- ListSnapMirrorSchedules
- ListSnapMirrorRelationships
- ListSnapMirrorVolumes
- ListSnapMirrorVservers
- ModifySnapMirrorEndpoint
- ModifySnapMirrorEndpoint (unmanaged)
- ModifySnapMirrorRelationship
- UpdateSnapMirrorRelationship
- QuiesceSnapMirrorRelationship
- ResumeSnapMirrorRelationship
- ResyncSnapMirrorRelationship

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AbortSnapMirrorRelationship

The Element software web UI uses the `AbortSnapMirrorRelationship` method to stop SnapMirror transfers that have started but are not yet complete.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeInfo	None	Yes
clearCheckpoint	Determines whether or not to clear the restart checkpoint. Possible values: • true • false	boolean	false	No

Return values

This method has the following return values:

Name	Description	Type
snapMirrorRelationship	An object containing information about the aborted SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

BreakSnapMirrorRelationship

The Element web UI uses the `BreakSnapMirrorRelationship` method to break a SnapMirror relationship. When a SnapMirror relationship is broken, the destination volume is made read-write and independent, and can then diverge from the source. You can reestablish the relationship with the `ResyncSnapMirrorRelationship` API method. This method requires the ONTAP cluster to be available.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeInfo	None	Yes

Return values

This method has the following return values:

Name	Description	Type
snapMirrorRelationship	An object containing information about the broken SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

Find more information

BreakSnapMirrorVolume

BreakSnapMirrorVolume

The Element web UI uses the `BreakSnapMirrorVolume` method to break the SnapMirror relationship between an ONTAP source container and Element target volume. Breaking an Element SnapMirror volume is useful if an ONTAP system becomes unavailable while replicating data to an Element volume. This feature enables a storage administrator to take control of an Element SnapMirror volume, break its relationship with the remote ONTAP system, and revert the volume to a previous snapshot.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	The volume on which to perform the break operation. The volume access mode must be snapMirrorTarget.	integer	None	Yes
snapshotID	Roll back the volume to the snapshot identified by this ID. The default behavior is to roll back to the most recent snapshot.	integer	None	No
preserve	Preserve any snapshots newer than the snapshot identified by snapshotID. Possible values: • true: Preserve snapshots newer than snapshotID. • false: Do not preserve snapshots newer than snapshotID. If false, any snapshots newer than snapshotID are deleted.	boolean	false	No
access	Resulting volume access mode. Possible values: • readWrite • readOnly • locked	string	readWrite	No

Return values

This method has no return values.

New since version

10.0

Find more information

BreakSnapMirrorRelationship

CreateSnapMirrorEndpoint

The Element web UI uses the `CreateSnapMirrorEndpoint` method to create a relationship with a remote SnapMirror endpoint.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
managementIP	The management IP address of the remote SnapMirror endpoint.	string	None	Yes
username	The management user name for the ONTAP system.	string	None	Yes
password	The management password for the ONTAP system.	string	None	Yes

Return values

This method has the following return values:

Name	Description	Type
snapMirrorEndpoint	The newly created SnapMirror endpoint.	snapMirrorEndpoint

New since version

10.0

CreateSnapMirrorEndpointUnmanaged

The Element software storage system uses the `CreateSnapMirrorEndpointUnmanaged` method to enable remote, unmanaged SnapMirror endpoints to communicate with a Element storage cluster. Unmanaged

endpoints cannot be administered using the Element SnapMirror APIs. They must be managed with ONTAP management software or APIs.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
clusterName	The name of the endpoint.	string	None	Yes
ipAddresses	The list of IP addresses for a cluster of ONTAP storage systems that should communicate with this Element storage cluster.	string array	None	Yes

Return values

This method has the following return values:

Name	Description	Type
snapMirrorEndpoint	The newly created SnapMirror endpoint.	snapMirrorEndpoint

New since version

10.3

CreateSnapMirrorRelationship

The Element web UI uses the `CreateSnapMirrorRelationship` method to create a SnapMirror extended data protection relationship between a source and destination endpoint.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
sourceVolume	The source volume in the relationship.	snapMirrorVolumeInfo	None	Yes
destinationVolume	The destination volume in the relationship.	snapMirrorVolumeInfo	None	Yes
relationshipType	The type of relationship. On storage systems running Element software, this value is always “extended_data_protection”.	string	None	No
policyName	Specifies the name of the ONTAP SnapMirror policy for the relationship. If not specified, the default policy name is MirrorLatest.	string	None	No
scheduleName	The name of the pre-existing cron schedule on the ONTAP system that is used to update the SnapMirror relationship. If no schedule is designated, snapMirror updates are not scheduled and must be updated manually.	string	None	No

Name	Description	Type	Default value	Required
maxTransferRate	Specifies the maximum data transfer rate between the volumes in kilobytes per second. The default value, 0, is unlimited and permits the SnapMirror relationship to fully utilize the available network bandwidth.	integer	None	No

Return values

This method has the following return values:

Name	Description	Type
snapMirrorRelationship	Information about the newly created SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

CreateSnapMirrorVolume

The Element web UI uses the `CreateSnapMirrorVolume` method to create a volume on the remote ONTAP system.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
vserver	The name of the Vserver.	string	None	Yes

Name	Description	Type	Default value	Required
name	The destination ONTAP volume name.	string	None	Yes
type	The volume type. Possible values: • rw: Read-write volume • ls: Load-sharing volume • dp: Data protection volume If no type is provided, the default type is dp.	string	None	No
aggregate	The containing ONTAP aggregate in which to create the volume. You can use ListSnapMirrorAggregates to get information about available ONTAP aggregates.	string	None	Yes
size	The size of the volume in bytes.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
snapMirrorVolume	Information about a SnapMirror volume.	snapMirrorVolume

New since version

10.1

DeleteSnapMirrorEndpoints

The Element web UI uses `DeleteSnapMirrorEndpoints` to delete one or more

SnapMirror endpoints from the system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointIDs	An array of IDs of SnapMirror endpoints to delete.	integer array	None	Yes

Return values

This method has no return values.

New since version

10.0

DeleteSnapMirrorRelationships

The Element web UI uses the `DeleteSnapMirrorRelationships` method to remove one or more SnapMirror relationships between a source and destination endpoint.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
destinationVolumes	The destination volume or volumes in the SnapMirror relationship.	snapMirrorVolumeInfo array	None	Yes

Return values

This method has the following return values:

Name	Description	Type

result	If the delete action succeeded, this object contains a success message. If the action failed, it contains an error message.	JSON object

New since version

10.1

GetOntapVersionInfo

The Element web UI uses `GetOntapVersionInfo` to get information about API version support from the ONTAP cluster in a SnapMirror relationship.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	If provided, the system lists the version information from the endpoint with the specified snapMirrorEndpointID. If not provided, the system lists the version information of all known SnapMirror endpoints.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
ontapVersionInfo	The software version information of the ONTAP endpoint.	ontapVersionInfo array

New since version

10.1

GetSnapMirrorClusterIdentity

The Element software web UI uses `GetSnapMirrorClusterIdentity` to get identity

information about the ONTAP cluster.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	If provided, the system lists the cluster identity of the endpoint with the specified snapMirrorEndpointID. If not provided, the system lists the cluster identity of all known SnapMirror endpoints.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorClusterIdentity	A list of cluster identities of SnapMirror endpoints.	snapMirrorClusterIdentity array

New since version

10.1

InitializeSnapMirrorRelationship

The Element software web UI uses the `InitializeSnapMirrorRelationship` method to initialize the destination volume in a SnapMirror relationship by performing an initial baseline transfer between clusters.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The ID of the remote ONTAP system.	integer	None	Yes

Name	Description	Type	Default value	Required
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeInfo	None	Yes
maxTransferRate	Specifies the maximum data transfer rate between the volumes in kilobytes per second. The default value, 0, is unlimited and permits the SnapMirror relationship to fully utilize the available network bandwidth.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorRelationship	Information about the initialized SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

ListSnapMirrorAggregates

The Element software web UI uses the `ListSnapMirrorAggregates` method to list all SnapMirror aggregates that are available on the remote ONTAP system. An aggregate describes a set of physical storage resources.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	Return only the aggregates associated with the specified endpoint ID. If no endpoint ID is provided, the system lists aggregates from all known SnapMirror endpoints.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorAggregates	A list of the aggregates available on the ONTAP storage system.	snapMirrorAggregate array

New since version

10.1

ListSnapMirrorEndpoints

The Element software web UI uses the `ListSnapMirrorEndpoints` method to list all SnapMirror endpoints that the Element storage cluster is communicating with.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointIDs	Return only the objects associated with these IDs. If no IDs are provided or the array is empty, the method returns all SnapMirror endpoint IDs.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorEndpoints	A list of existing SnapMirror endpoints.	snapMirrorEndpoint array

New since version

10.0

ListSnapMirrorLuns

The Element software web UI uses the `ListSnapMirrorLuns` method to list the LUN information for the SnapMirror relationship from the remote ONTAP cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	List only the LUN information associated with the specified endpoint ID.	integer	None	Yes
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeInfo	None	Yes

Return values

This method has the following return values:

Name	Description	Type
snapMirrorLunInfos	A list of objects containing information about SnapMirror LUNs.	snapMirrorLunInfo array

New since version

10.1

ListSnapMirrorNetworkInterfaces

The Element software web UI uses the `ListSnapMirrorNetworkInterfaces` method to list all available SnapMirror interfaces on a remote ONTAP system.

Parameter

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	Return only the network interfaces associated with the specified endpoint ID. If no endpoint ID is provided, the system lists interfaces from all known SnapMirror endpoints.	integer	None	No
interfaceRole	List only the network interface serving the specified role.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorNetworkInterfaces	A list of the SnapMirror network interfaces available on the remote ONTAP storage system.	snapMirrornetworkInterface array

New since version

10.1

ListSnapMirrorNodes

The Element software web UI uses the `ListSnapMirrorNodes` method to get a list of nodes in a remote ONTAP cluster.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	If provided, the system lists the nodes of the endpoint with the specified snapMirrorEndpointID. If not provided, the system lists the nodes of all known SnapMirror endpoints.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorNodes	A list of the nodes on the ONTAP cluster.	snapMirrorNode array

New since version

10.1

ListSnapMirrorPolicies

The Element software web UI uses the `ListSnapMirrorPolicies` method to list all SnapMirror policies on a remote ONTAP system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	List only the policies associated with the specified endpoint ID. If no endpoint ID is provided, the system lists policies from all known SnapMirror endpoints.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorPolicies	A list of the SnapMirror policies on the ONTAP storage system.	snapMirrorPolicy array

New since version

10.1

ListSnapMirrorSchedules

The Element software web UI uses the `ListSnapMirrorSchedules` method to get a list of schedules that are available on a remote ONTAP cluster.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	If provided, the system lists the schedules of the endpoint with the specified SnapMirror endpoint ID. If not provided, the system lists the schedules of all known SnapMirror endpoints.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorSchedules	A list of the SnapMirror schedules on the remote ONTAP cluster.	snapMirrorJobScheduleCronInfo array

New since version

10.1

ListSnapMirrorRelationships

The Element software web UI uses the `ListSnapMirrorRelationships` method to list one or all SnapMirror relationships on an Element storage cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	List only the relationships associated with the specified endpoint ID. If no endpoint ID is provided, the system lists relationships from all known SnapMirror endpoints.	integer	None	No
destinationVolume	List relationships associated with the specified destination volume.	snapMirrorVolumeInfo	None	No
sourceVolume	List relationships associated with the specified source volume.	snapMirrorVolumeInfo	None	No
vserver	List relationships on the specified Vserver.	string	None	No
relationshipID	List relationships associated with the specified relationship ID.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorRelationships	A list of objects containing information about SnapMirror relationships.	snapMirrorRelationship array

New since version

10.1

ListSnapMirrorVolumes

The Element software web UI uses the `ListSnapMirrorVolumes` method to list all SnapMirror volumes available on a remote ONTAP system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	List only the volumes associated with the specified endpoint ID. If no endpoint ID is provided, the system lists volumes from all known SnapMirror endpoints.	integer	None	No
vserver	List volumes hosted on the specified Vserver. The Vserver must be of type "data".	string	None	No
name	List only ONTAP volumes with the specified name.	string	None	No
type	List only ONTAP volumes of the specified type. Possible values: • rw: Read-write volumes • ls: Load-sharing volumes • dp: Data protection volumes	string	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorVolumes	A list of the SnapMirror volumes available on the ONTAP storage system.	snapMirrorVolume array

New since version

10.1

ListSnapMirrorVservers

The Element software web UI uses the `ListSnapMirrorVservers` method to list all SnapMirror Vservers available on a remote ONTAP system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	List only the Vservers associated with the specified endpoint ID. If no endpoint ID is provided, the system lists Vservers from all known SnapMirror endpoints.	integer	None	No
vserverType	List only Vservers of the specified type. Possible values: • admin • data • node • system	string	None	No
vserverName	List only Vservers with the specified name.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorVservers	A list of the SnapMirror Vservers available on the ONTAP storage system.	snapMirrorVserver array

New since version

10.1

ModifySnapMirrorEndpoint

The Element software web UI uses the `ModifySnapMirrorEndpoint` method to change the name and management attributes for a SnapMirror endpoint.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
SnapMirrorEndpointID	The SnapMirror endpoint to modify.	integer	None	Yes
managementIP	The new management IP Address for the ONTAP system.	string	None	No
username	The new management user name for the ONTAP system.	string	None	No
password	The new management password for the ONTAP system.	string	None	No

Return value

This method has the following return value:

Name	Description	Type

snapMirrorEndpoint	Information about the modified SnapMirror endpoint.	snapMirrorEndpoint

New since version

10.0

ModifySnapMirrorEndpoint (unmanaged)

Element software uses this version of the `ModifySnapMirrorEndpoint` method to modify the storage cluster name or IP address attributes for an unmanaged SnapMirror endpoint. Unmanaged endpoints cannot be administered using the Element SnapMirror APIs. They must be managed with ONTAP management software or APIs.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
SnapMirrorEndpointID	The SnapMirror endpoint to modify.	integer	None	Yes
clusterName	The new name of the endpoint.	string	None	No
ipAddresses	The new list of IP addresses for a cluster of ONTAP storage systems that should communicate with this Element storage cluster.	string array	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorEndpoint	Information about the modified SnapMirror endpoint.	snapMirrorEndpoint

New since version

10.3

ModifySnapMirrorRelationship

You can use `ModifySnapMirrorRelationship` to change the intervals at which a scheduled snapshot occurs. You can also delete or pause a schedule by using this method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeinfo	None	Yes
maxTransferRate	Specifies the maximum data transfer rate between the volumes in kilobytes per second. The default value, 0, is unlimited and permits the SnapMirror relationship to fully utilize the available network bandwidth.	integer	None	No
policyName	Specifies the name of the ONTAP SnapMirror policy for the relationship.	string	None	No
scheduleName	The name of the pre-existing cron schedule on the ONTAP system that is used to update the SnapMirror relationship.	string	None	No
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
snapMirrorRelationship	An object containing the modified SnapMirror relationship attributes.	snapMirrorRelationship

New since version

10.1

UpdateSnapMirrorRelationship

The Element software web UI uses the `UpdateSnapMirrorRelationship` method to make the destination volume in a SnapMirror relationship an up-to-date mirror of the source volume.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeinfo	None	Yes
maxTransferRate	Specifies the maximum data transfer rate between the volumes in kilobytes per second. The default value, 0, is unlimited and permits the SnapMirror relationship to fully utilize the available network bandwidth.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorRelationship	An object containing information about the updated SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

QuiesceSnapMirrorRelationship

The Element software web UI uses the `QuiesceSnapMirrorRelationship` method to disable future data transfers for a SnapMirror relationship. If a transfer is in progress, the relationship status becomes "quiescing" until the transfer is complete. If the current transfer is aborted, it will not restart. You can reenable data transfers for the relationship using the `ResumeSnapMirrorRelationship` API method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeinfo	None	Yes

Return value

This method has the following return value:

Name	Description	Type
snapMirrorRelationship	An object containing information about the quiesced SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

ResumeSnapMirrorRelationship

The Element software web UI uses the `ResumeSnapMirrorRelationship` method to enable future transfers for a quiesced SnapMirror relationship.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeinfo	None	Yes

Return value

This method has the following return value:

Name	Description	Type
snapMirrorRelationship	An object containing information about the resumed SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

ResyncSnapMirrorRelationship

The Element software web UI uses the `ResyncSnapMirrorRelationship` method to establish or reestablish a mirror relationship between a source and destination endpoint. When you resync a relationship, the system removes snapshots on the destination volume that are newer than the common snapshot copy, and then mounts the destination volume as a data protection volume with the common snapshot copy as the exported snapshot copy.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapMirrorEndpointID	The endpoint ID of the remote ONTAP storage system communicating with the Element storage cluster.	integer	None	Yes
destinationVolume	The destination volume in the SnapMirror relationship.	snapMirrorVolumeinfo	None	Yes
maxTransferRate	Specifies the maximum data transfer rate between the volumes in kilobytes per second. The default value, 0, is unlimited and permits the SnapMirror relationship to fully utilize the available network bandwidth.	integer	None	No
sourceVolume	The source volume in the SnapMirror relationship.	snapMirrorVolumeinfo	None	No

Return value

This method has the following return value:

Name	Description	Type
snapMirrorRelationship	An object containing information about the resynced SnapMirror relationship.	snapMirrorRelationship

New since version

10.1

System configuration API methods

System configuration API methods enable you to obtain and set configuration values that apply to all nodes in the cluster.

- DisableBmcColdReset
- DisableClusterSsh
- DisableSnmp
- EnableBmcColdReset
- EnableClusterSsh
- EnableSnmp
- GetBinAssignmentProperties
- GetClusterSshInfo
- GetClusterStructure
- GetFipsReport
- GetLldpConfig
- GetLldpInfo
- GetNodeFipsDrivesReport
- GetNtpInfo
- GetNvramInfo
- GetProtectionDomainLayout
- GetRemoteLoggingHosts
- GetSnmpACL
- GetSnmpInfo
- GetSnmpState
- GetSnmpTrapInfo
- GetSSLCertificate
- ListProtectionDomainLevels
- RemoveSSLCertificate
- ResetNetworkConfig
- ResetSupplementalTlsCiphers
- SetClusterStructure
- SetLldpConfig
- SetNtpInfo
- SetProtectionDomainLayout
- SetRemoteLoggingHosts
- SetSnmpACL
- SetSnmpInfo

- SetSnmpTrapInfo
- SetSSLCertificate
- SnmpSendTestTraps
- TestAddressAvailability

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

DisableBmcColdReset

You can use the `DisableBmcColdReset` method to disable the background task that periodically resets the Baseboard Management Controller (BMC) for all nodes in the cluster.

Parameter

This method has no input parameter.

Return values

This method has the following return value:

Name	Description	Type
cBmcResetDurationMinutes	Returns the time between reset intervals. The interval should always be 0 after the command completes.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableBmcColdReset",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "cBmcResetDurationMinutes": 0
    }
}
```

New since version

12.0

DisableClusterSsh

You can use the `DisableClusterSsh` method to disable the SSH service for the entire storage cluster. When you add nodes to the storage cluster, the new nodes will inherit this cluster-wide setting.

Parameter

This method has no input parameter.

Return value

This method has the following return value:

Name	Description	Type
result	A JSON object containing the status of the SSH service for the storage cluster, the time remaining until SSH is disabled, and the SSH service status for each node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableClusterSsh",
  "params": {
      },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result" : {
    "enabled": true,
    "timeRemaining": "00:43:21",
    "nodes": [
    {
        "nodeID": 1,
        "enabled": true
    },
    {
        "nodeID": 2,
        "enabled": true
    },
    {
        "nodeID": 3,
        "enabled": false
    },
    {
        "nodeID": 4,
        "enabled": false
    } ]
            }
    }
```

New since version

10.3

DisableSnmp

You can use the `DisableSnmp` method to disable SNMP on the cluster nodes.

Parameter

This method has no input parameter.

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableSnmp",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "result" : {},
  "id" : 1
}
```

New since version

9.6

EnableBmcColdReset

You can use the `EnableBmcColdReset` method to enable a background task that periodically resets the Baseboard Management Controller (BMC) for all nodes in the cluster.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
timeout	The time between BMC reset operations, in minutes.	integer	20160 minutes	No

Return values

This method has the following return value:

Name	Description	Type
cBmcResetDurationMinutes	Returns the time between reset intervals. The interval should always be 0 after the command completes.	integer

Request example

Requests for this method are similar to the following example:

```
{
    "method": "EnableBmcColdReset",
    "params": {
                "timeout": 36000
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "cBmcResetDurationMinutes": 36000
    }
}
```

New since version

12.0

EnableClusterSsh

You can use the `EnableClusterSsh` method to enable the SSH service on all nodes in the storage cluster.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
duration	The amount of time that the SSH service will remain enabled.	string	None	Yes

Return values

This method has the following return values:

Name	Description	Type
result	A JSON object containing the status of the SSH service for the storage cluster, the time remaining until SSH is disabled, and the SSH service status for each node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableClusterSsh",
  "params": {
      "duration"  : "02:00:00.00"
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result" : {
    "enabled": true,
    "timeRemaining": "00:43:21",
    "nodes": [
    {
        "nodeID": 1,
        "enabled": true
    },
    {
        "nodeID": 2,
        "enabled": true
    },
    {
        "nodeID": 3,
        "enabled": false
    },
    {
        "nodeID": 4,
        "enabled": false
    } ]
            }
    }
```

New since version

10.3

EnableSnmp

You can use the `EnableSnmp` method to enable SNMP on cluster nodes. When you enable SNMP, the action applies to all nodes in the cluster, and the values that are passed replace all values set in any previous call to `EnableSnmp`.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
snmpV3Enabled	If set to true, then SNMP v3 is enabled on each node in the cluster. If set to false, then SNMP v2 is enabled.	boolean	false	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableSnmp",
  "params": {
      "snmpV3Enabled"  : "true"
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

9.6

GetBinAssignmentProperties

You can use the `GetBinAssignmentProperties` method to retrieve the bin assignment properties in the database.

Parameter

This method has the no input parameters.

Return value

This method has the following return value:

Name	Description	Type
properties	Details the properties for all current bin assignments in the database.	binAssignmentProperties array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "GetBinAssignmentProperties",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "properties": {
            "algorithmRuntimeMS": 1105,
            "areReplicasValid": true,
            "binCount": 65536,
            "isBalanced": true,
            "isStable": true,
            "isWellCoupled": false,
            "layout": [
                {
                    "protectionDomainName": "1",
                    "services": [
                        {
                            "budget": 7281,
                            "serviceID": 16
                        },
                        {
                            "budget": 7281,
                            "serviceID": 19
                        },
```

```
                    {
                        "budget": 7281,
                        "serviceID": 24
                    }
                ]
            },
            {
                "protectionDomainName": "2",
                "services": [
                    {
                        "budget": 7281,
                        "serviceID": 17
                    },
                    {
                        "budget": 7281,
                        "serviceID": 20
                    },
                    {
                        "budget": 7281,
                        "serviceID": 22
                    }
                ]
            },
            {
                "protectionDomainName": "3",
                "services": [
                    {
                        "budget": 7281,
                        "serviceID": 18
                    },
                    {
                        "budget": 7281,
                        "serviceID": 21
                    },
                    {
                        "budget": 7281,
                        "serviceID": 23
                    }
                ]
            }
        ],
        "numSwaps": 0,
        "numUpdatingBins": 0,
        "protectionDomainType": "node",
        "reason": "Final",
        "replicationCount": 2,
```

```
            "requestRebalance": false,
            "serviceStrandedCapacities": [],
            "timePublished": "2020-04-02T18:34:07.807681Z",
            "validSchemes": []
        }
    }
}
```

New since version

12.0

GetClusterSshInfo

You can use the `GetClusterSshInfo` method to query the status of the SSH service for the entire storage cluster.

Parameter

This method has no input parameter.

Return value

This method has the following return value:

Name	Description	Type
result	A JSON object containing the status of the SSH service for the storage cluster, the time remaining until SSH is disabled, and the SSH service status for each node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetClusterSshInfo",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result" : {
    “enabled”: “true”,
    “timeRemaining”: "00:43:21",
    "nodes": [
    {
        "nodeID": 1,
        "enabled": true
    },
    {
        "nodeID": 2,
        "enabled": true
    },
    {
        "nodeID": 3,
        "enabled": false
    },
    {
        "nodeID": 4,
        "enabled": false
    } ]
            }
    }
```

New since version

10.3

GetClusterStructure

You can use the `GetClusterStructure` method to back up the current storage cluster configuration information. If the storage cluster configuration is changed while this method is running, the contents of the configuration backup will be unpredictable. You can save this data to a text file and restore it on other clusters, or the same cluster in the case of a disaster.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
result	A JSON object containing the current storage cluster configuration information.	clusterStructure

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetClusterStructure",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result" : <clusterStructure object containing configuration
information>
    }
```

New since version

10.3

GetFipsReport

You can use the `GetFipsReport` method to check the FIPS 140-2 encryption feature support status of all nodes in the storage cluster.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
result	A JSON object containing the status of FIPS 140-2 feature support for every node, and error information for each node that did not respond to the query.	fipsReport

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetFipsReport",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
    "nodes": [
        {
           "nodeID": 1,
           "fipsDrives": "None",
           "httpsEnabled": true
        },
        {
           "nodeID": 3,
           "fipsDrives": "None",
           "httpsEnabled": true
        }
    ],
    "errorNodes": [
        {
           "nodeID": 2,
           "error": {
               "message": "The RPC timed out.",
               "name": "xRpcTimeout"
               }
        }
    ]
    }
}
```


New since version

10.3

GetLldpConfig

You can use the `GetLldpConfig` method to get the Link Layer Discovery Protocol (LLDP) configuration for each node of a storage cluster.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
lldpConfig	Information about the storage cluster LLDP configuration.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetLldpConfig",
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": null,
        "result": {
            "lldpConfig": {
                "enableLldp": true,
                "enableMed": false,
                "enableOtherProtocols": true
        }
    }
}
```

GetLldpInfo

You can use the `GetLldpInfo` method to get the Link Layer Discovery Protocol (LLDP) configuration for each node of a storage cluster, or an individual storage node.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
lldpInfo	Information about the chassis, interface, and neighbor LLDP settings for each node of a storage cluster.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetLldpInfo",
   "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

11.0

Find more information

GetLldpInfo

GetNodeFipsDrivesReport

You can use the `GetNodeFipsDrivesReport` method to check the FIPS 140-2 drive encryption capability status of a single node in the storage cluster. You must run this method against an individual storage node.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
fipsDrives	A JSON object containing the status of FIPS 140-2 feature support for this node. Possible values: • None: Node is not FIPS capable. • Partial: Node is FIPS capable but not all drives in the node are FIPS drives. • Ready: Node is FIPS capable and all drives in the node are FIPS drives (or no drives are present).	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetNodeFipsDrivesReport",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "fipsDrives": "None"
    }
}
```

New since version

11.5

GetNtpInfo

You can use the `GetNtpInfo` method to get the current network time protocol (NTP) configuration information.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
servers	List of NTP servers.	string array
broadcastclient	Indicates whether or not the nodes in the cluster are listening for broadcast NTP messages. Possible values: • true • false	boolean

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetNtpInfo",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
    "broadcastclient" : false,
    "servers" : [ "us.pool.ntp.org" ]
   }
}
```

New since version

9.6

GetNvramInfo

You can use the `GetNvramInfo` method to get information from each node about the NVRAM card.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
force	The force parameter must be included on this method to successfully run on all nodes in the cluster.	boolean	None	Yes

Return value

This method has the following return value:

Name	Description	Type
nvramInfo	Arrays of events and errors detected on the NVRAM card.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetNvramInfo",
  "params": {
      "force": true
      },
  "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

GetNvramInfo

GetProtectionDomainLayout

You can use the `GetProtectionDomainLayout` method to return all protection domain information for a cluster, including which chassis and which custom protection domain each node is in.

Parameter

This method has the no input parameters.

Return value

This method has the following return value:

Name	Description	Type
protectionDomainLayout	List of nodes, each with its associated protection domains.	JSON list of nodeProtectionDomains objects.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetProtectionDomainLayout",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "protectionDomainLayout": [
      {
        "nodeID": 1,
        "protectionDomains": [
          {
            "protectionDomainName": "QTFCR2914008D",
            "protectionDomainType": "chassis"
```

```
        },
        {
          "protectionDomainName": "Rack-1",
          "protectionDomainType": "custom"
        }
      ]
    },
    {
      "nodeID": 2,
      "protectionDomains": [
        {
          "protectionDomainName": "QTFCR291500EA",
          "protectionDomainType": "chassis"
        },
        {
          "protectionDomainName": "Rack-1",
          "protectionDomainType": "custom"
        }
      ]
    },
    {
      "nodeID": 3,
      "protectionDomains": [
        {
          "protectionDomainName": "QTFCR291500C3",
          "protectionDomainType": "chassis"
        },
        {
          "protectionDomainName": "Rack-2",
          "protectionDomainType": "custom"
        }
      ]
    },
    {
      "nodeID": 4,
      "protectionDomains": [
        {
          "protectionDomainName": "QTFCR291400E6",
          "protectionDomainType": "chassis"
        },
        {
          "protectionDomainName": "Rack-2",
          "protectionDomainType": "custom"
        }
      ]
    }
```

```
      ]
    }
}
```

New since version

12.0

GetRemoteLoggingHosts

You can use the `GetRemoteLoggingHosts` method to get the current list of log servers.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
remoteHosts	List of IP address and port information about hosts configured to receive forwarded logging information.	loggingServer array

Request example

Requests for this method are similar to the following example:

```
{
   "id": 3386609,
   "method": "GetRemoteLoggingHosts",
   "params": {}
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 3386609,
   "result": {
      "remoteHosts": [
         {
           "host": "172.16.1.20",
           "port": 10514
         },
         {
           "host": "172.16.1.25"
         }
      ]
   }
}
```


New since version

9.6

Find more information

SetRemoteLoggingHosts

GetSnmpACL

You can use the `GetSnmpACL` method to get the current SNMP access permissions on the cluster nodes.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
networks	List of networks and what type of access they have to the SNMP servers running on the cluster nodes. This value is present if SNMP v3 is disabled.	network array

Name	Description	Type
usmUsers	List of users and the type of access they have to the SNMP servers running on the cluster nodes. This value is present if SNMP v3 is enabled.	usmUser array

Request example

Requests for this method are similar to the following example:

```
{
"method": "GetSnmpACL",
"params": {},
"id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
     "usmUsers"  : [
       {
          "name": "jdoe",
          "access": "rouser",
          "secLevel": "priv",
          "password": "mypassword",
          "passphrase": "mypassphrase",
       }
     ]
   }
}
```

New since version

9.6

GetSnmpInfo

You can use the `GetSnmpInfo` method to get the current simple network management protocol (SNMP) configuration information.

Parameters

GetSnmpInfo is deprecated for versions later than Element version 8.0. The GetSnmpState and SetSnmpACL methods replace the GetSnmpInfo method.

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
networks	List of networks and access types enabled for SNMP. **Note:** networks is only displayed if SNMP v3 is disabled.	network
enabled	Indicates if the nodes in the cluster are configured for SNMP. Possible values: • true • false	boolean
snmpV3Enabled	If the node in the cluster is configured for SNMP v3. Possible values: • true • false	boolean
usmUsers	If SNMP v3 is enabled, a list of user access parameters for SNMP is returned from the cluster. This is returned instead of the networks parameter.	usmUser

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetSnmpInfo",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "enabled" : true,
      "networks" : [
         {
            "access" : "rosys",
            "cidr" : 0,
            "community" : "public",
            "network" : "localhost"
         }
      ]
   }
}
```

New since version

9.6

Find more information

- GetSnmpState
- SetSnmpACL

GetSnmpState

You can use the `GetSnmpState` method to get the current state of the SNMP feature.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
enabled	Possible values: • true • false Default value is false. Returns true if the nodes in the cluster are configured for SNMP.	boolean
snmpV3Enabled	Possible values: • true • false Default value is false. Returns true if the nodes in the cluster are configured for SNMP v3.	boolean

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetSnmpState",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
   "enabled": true,
   "snmpV3Enabled": false
   }
}
```

New since version

9.6

Find more information

SetSnmpACL

GetSnmpTrapInfo

You can use the `GetSnmpTrapInfo` method to get current SNMP trap configuration information.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
trapRecipients	List of hosts that are to receive the traps generated by the cluster.	snmpTrapRecipient array
clusterFaultTrapsEnabled	The value true indicates that a solidFireClusterFaultNotification is configured to be sent to the list of trap recipients when a cluster fault is logged.	boolean
clusterFaultResolvedTrapsEnabled	The value true indicates that a solidFireClusterFaultResolvedNotification is configured to be sent to the list of trap recipients when a cluster fault is resolved.	boolean
clusterEventTrapsEnabled	The value true indicates that a solidFireClusterEventNotification is configured to be sent to the list of trap recipients when a cluster event is logged.	boolean

Request example

Requests for this method are similar to the following example:

```
{
   "method":"GetSnmpTrapInfo"
   "params":{},
   "id":1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "clusterEventTrapsEnabled": true,
    "clusterFaultResolvedTrapsEnabled": true,
    "clusterFaultTrapsEnabled": true,
    "trapRecipients": [
     {
      "community": "public",
      "host": "192.168.151.60",
      "port": 162
     },
     {
      "community": "solidfireAlerts",
      "host": "NetworkMonitor",
      "port": 162
     },
     {
      "community": "wakeup",
      "host": "PhoneHomeAlerter",
      "port": 1008
     }
   ]
 }
}
```

New since version

9.6

GetSSLCertificate

You can use the `GetSSLCertificate` method to retrieve the SSL certificate that is currently active on the storage nodes of the cluster.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
certificate	The full PEM-encoded text of the certificate.	string
details	The decoded information of the certificate.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "GetSSLCertificate",
    "params" : {},
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:


```
{
    "id": 1,
    "result": {
        "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD\nVQQGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G\nA1UEChMYV2hhdC
BIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1
UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGF
doYXQgSGFwcGVucyBpbiBWZWdh\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdm
VnYXNzdGF5c2luLnZl\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28f
nLKQNWEWMR\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FH
P4\nt6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nmrgp
GZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\n7dg9cyXu4afXcgKy14Ji
NBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrn
DJyTsdnEbaGBiaSBhjCBgzELMAkG\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxW
ZWdhcywgQmFieSExITAf\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqG
SIb3DQEJARYe\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYD
VR0T\nBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe\nMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N\niebEgAaE5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl+DwkxnS1cx1v
```

```
ERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDg
l0QoQDWNDoTeRBsz80nwiouA==\n-----END CERTIFICATE-----\n",
        "details": {
            "issuer":
"/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org",
            "modulus":
"F14FB6F1F9CB290356116311E9A91E0CAB9E852A52EFDA1D2C68A0235F2A94257F0146396
4B8EAB138C1BD325546FE38CA809380DAF1DFA53B1473F8B7A3FF4A2D1A62BE28BF1979C03
A44337432CB924F07B25E94E07A003EDF9A24F078FDB41D162966F63E533ECB6041429AB82
9199405DE239221C047B4B284E75F3A2554FA8F9760EB28D41903B7E76CA573D1D71DC9FA9
5BFE3CA5D0399535467471A430026212DC99A8CB1FB38FF61AE162AAFB64AA4C05FB6D7D05
DF01C77D79D99479CCF1F113E4DFFD03E2BA952EDD83D7325EEE1A7D77202B2D78262341BE
A6C18E1809B44EFAC80CBAAD31EED313378E376471BF58F2688DCF117E002ABE8AD6B",
            "notAfter": "2027-03-06T22:50:26Z",
            "notBefore": "2017-03-08T22:50:26Z",
            "serial": "CC1B221598E37FF3",
            "sha1Fingerprint":
"1D:70:7A:6F:18:8A:CD:29:50:C7:95:B1:DD:5E:63:21:F4:FA:6E:21",
            "subject":
"/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org"
        }
    }
}
```


New since version

10.0

ListProtectionDomainLevels

You can use the `ListProtectionDomainLevels` method to list the tolerance and resiliency levels of the storage cluster. Tolerance levels indicate the cluster's ability to continue reading and writing data in the event of a failure, and resiliency levels indicate the storage cluster's ability to automatically heal itself from one or more failures.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
protectionDomainLevels	A list of the different protection domain levels, where each supplies the storage cluster's tolerance and resiliency information.	protectionDomainLevel

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListProtectionDomainLevels",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "protectionDomainLevels": [
            {
                "protectionDomainType": "node",
                "resiliency": {
                    "protectionSchemeResiliencies": [
                            {
                                "protectionScheme": "doubleHelix",
                                "sustainableFailuresForBlockData": 0,
                                "sustainableFailuresForMetadata": 1
                            }
                    ],
                    "singleFailureThresholdBytesForBlockData": 0,
                    "sustainableFailuresForEnsemble": 1
                },
                "tolerance": {
                    "protectionSchemeTolerances": [
                            {
                                "protectionScheme": "doubleHelix",
                                "sustainableFailuresForBlockData": 0,
                                "sustainableFailuresForMetadata": 1
                            }
                    ],
```

```
                    "sustainableFailuresForEnsemble": 1
                }
            },
            {
                "protectionDomainType": "chassis",
                "resiliency": {
                    "protectionSchemeResiliencies": [
                            {
                                "protectionScheme": "doubleHelix",
                                "sustainableFailuresForBlockData": 0,
                                "sustainableFailuresForMetadata": 1
                            }
                    ],
                    "singleFailureThresholdBytesForBlockData": 0,
                    "sustainableFailuresForEnsemble": 1
                },
                "tolerance": {
                    "protectionSchemeTolerances": [
                            {
                                "protectionScheme": "doubleHelix",
                                "sustainableFailuresForBlockData": 0,
                                "sustainableFailuresForMetadata": 1
                            }
                    ],
                    "sustainableFailuresForEnsemble": 1
                }
            }
        ]
    }
}
```

New since version

11.0

RemoveSSLCertificate

You can use the `RemoveSSLCertificate` method to remove the user SSL certificate and private key for the storage nodes in the cluster. After the certificate and private key are removed, the storage nodes are configured to use the default certificate and private key.

Parameters

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "RemoveSSLCertificate",
    "params" : {},
    "id" : 3
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id" : 3,
    "result" : {}
}
```

New since version

10.0

ResetNetworkConfig

You can use the `ResetNetworkConfig` method to help resolve network configuration issues for an individual node. This method resets an individual node's network configuration to the factory default settings.

Parameters

This method has no input parameters.

Return value

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ResetNetworkConfig",
   "params": {},
   "id" : 1
}
```

Response example

This method does not return a response.

New since version

11.0

ResetSupplementalTlsCiphers

You can use the `ResetSupplementalTlsCiphers` method to restore the list of supplemental TLS ciphers to the default. You can use this method on the entire cluster.

Parameter

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ResetSupplementalTlsCiphers",
  "params": {},
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

11.3

SetClusterStructure

You can use the `SetClusterStructure` method to restore the storage cluster configuration information from a backup. When you call the method, you pass the clusterStructure object containing the configuration information you want to restore as the params parameter.

Parameter

This method has the following input parameter:

Name	Description	Type
params	A JSON object containing the current storage cluster configuration information.	clusterStructure

Return values

This method has the following return values:

Name	Description	Type
result	Asynchronous result handle.	asyncHandle

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetClusterStructure",
  "params": <insert clusterStructure object here>,
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result" : {
    "asyncHandle": 1
    }
}
```

New since version

10.3

SetLldpConfig

You can use the `SetLldpConfig` method to configure the Link Layer Discovery Protocol (LLDP) settings for a storage cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
enableOtherProtocols	Enable automatic use of other discovery protocols - CDP, FDP, EDP, and SONMP.	boolean	true	No
enableMed	Enable Media Endpoint Discovery (LLDP-MED).	boolean	false	No
enableLldp	Enable or disable LLDP.	boolean	true	No

Return values

This method has the following return value:

Name	Description	Type
lldpConfig	Information about the current storage cluster LLDP configuration, including newly changed settings.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "id": 3920,
  "method":"SetLldpConfig",
    "params": {
        "lldpConfig": {
            "enableMed": true
        }
    }
  }
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 3920,
  "result": {
    "lldpConfig": {
        "enableLldp": true,
        "enableMed": true,
        "enableOtherProtocols": true
     }
  }
}
```

SetNtpInfo

You can use the `SetNtpInfo` method to configure NTP on cluster nodes. The values you set with this interface apply to all nodes in the cluster. If an NTP broadcast server periodically broadcasts time information on your network, you can optionally configure nodes as broadcast clients.

Parameters

Ensure that you use NTP servers that are internal to your network, rather than the installation defaults.

This method has the following input parameters:

Name	Description	Type	Default value	Required
servers	List of NTP servers to add to each node NTP configuration.	string array	None	Yes

Name	Description	Type	Default value	Required
broadcastclient	Enables every node in the cluster as a broadcast client.	boolean	false	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "SetNtpInfo",
   "params": {
     "servers"  : [
       "ntpserver1.example.org",
       "ntpserver2.example.org",
       "ntpserver3.example.org"
     ],
     "broadcastclient" : false
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id" : 1,
    "result" : {}
}
```

New since version

9.6

SetProtectionDomainLayout

You can use the `SetProtectionDomainLayout` method to assign nodes to custom protection domains.

Information must be provided for all active nodes in the cluster and no information can be provided for inactive

nodes. All nodes in a given chassis must be assigned to the same custom protection domain. The same protectionDomainType must be supplied for all nodes. protectionDomainTypes that are not custom, such as node and chassis, should not be included. If either of these are provided, then the custom protection domains are ignored and an appropriate error is returned.

Custom protection domains are not supported with the following configurations:

- Storage clusters containing shared chassis
- Two-node storage clusters

The method returns an error when used on storage clusters with these configurations.

Parameter

This method has the following input parameters:

Name	Description	Type	Default value	Required
protectionDomainLayout	Protection domain information for each node.	JSON list of nodeProtectionDomains objects.	None	Yes

Return value

This method has the following return value:

Name	Description	Type
protectionDomainLayout	List of nodes, each with its associated protection domains.	JSON list of nodeProtectionDomains objects.

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "SetProtectionDomainLayout",
  "params": {
    "protectionDomainLayout": [
      {
        "nodeID": 1,
        "protectionDomains": [
          {
            "protectionDomainName": "Rack-1",
            "protectionDomainType": "custom"
          }
        ]
      },
      {
        "nodeID": 2,
        "protectionDomains": [
          {
            "protectionDomainName": "Rack-1",
            "protectionDomainType": "custom"
          }
        ]
      },
      {
        "nodeID": 3,
        "protectionDomains": [
          {
            "protectionDomainName": "Rack-2",
            "protectionDomainType": "custom"
          }
        ]
      },
      {
        "nodeID": 4,
        "protectionDomains": [
          {
            "protectionDomainName": "Rack-2",
            "protectionDomainType": "custom"
          }
        ]
      }
    ]
  }
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
    "protectionDomainLayout": [
      {
        "nodeID": 1,
        "protectionDomains": [
          {
            "protectionDomainName": "QTFCR2914008D",
            "protectionDomainType": "chassis"
          },
          {
            "protectionDomainName": "Rack-1",
            "protectionDomainType": "custom"
          }
        ]
      },
      {
        "nodeID": 2,
        "protectionDomains": [
          {
            "protectionDomainName": "QTFCR291500EA",
            "protectionDomainType": "chassis"
          },
          {
            "protectionDomainName": "Rack-1",
            "protectionDomainType": "custom"
          }
        ]
      },
      {
        "nodeID": 3,
        "protectionDomains": [
          {
            "protectionDomainName": "QTFCR291500C3",
            "protectionDomainType": "chassis"
          },
          {
            "protectionDomainName": "Rack-2",
            "protectionDomainType": "custom"
          }
        ]
```

```
        },
        {
          "nodeID": 4,
          "protectionDomains": [
            {
              "protectionDomainName": "QTFCR291400E6",
              "protectionDomainType": "chassis"
            },
            {
              "protectionDomainName": "Rack-2",
              "protectionDomainType": "custom"
            }
          ]
        }
      ]
    }
}
```

New since version

12.0

SetRemoteLoggingHosts

You can use the `SetRemoteLoggingHosts` method to configure remote logging from the nodes in the storage cluster to a centralized log server or servers. Remote logging is performed over TCP using the default port 514. This API does not add to the existing logging hosts. Rather, it replaces what currently exists with new values specified by this API method. You can use `GetRemoteLoggingHosts` to determine what the current logging hosts are and then use `SetRemoteLoggingHosts` to set the desired list of current and new logging hosts.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
remoteHosts	List of hosts that are log message recipients.	loggingServer array	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "id": 1,
   "method": "SetRemoteLoggingHosts",
   "params": {
     "remoteHosts": [
        {
          "host": "172.16.1.20",
          "port": 10514
        },
        {
          "host": "172.16.1.25"
        }
      ]
   }
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id" : 1,
    "result" : {}
}
```

New since version

9.6

Find more information

GetRemoteLoggingHosts

SetSnmpACL

You can use the `SetSnmpACL` method to configure SNMP access permissions on the cluster nodes. The values you set with this interface apply to all nodes in the cluster, and the values that are passed replace all values set in any previous call to `SetSnmpACL`. Also note that the values set with this interface replace all network or usmUsers values set with the `SetSnmpInfo` method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
networks	List of networks and what type of access they have to the SNMP servers running on the cluster nodes. See SNMP network object for possible networks values. This parameter is required if SNMP v3 is disabled.	network	None	No
usmUsers	List of users and the type of access they have to the SNMP servers running on the cluster nodes. This parameter is required if SNMP v3 is enabled.	usmUser	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetSnmpACL",
  "params": {
    "usmUsers"  : [
      {
         "name": "jdoe",
         "access": "rouser",
         "secLevel": "priv",
         "password": "mypassword",
         "passphrase": "mypassphrase",
      }
    ]
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

9.6

Find more information

SetSnmpInfo

SetSnmpInfo

You can use the `SetSnmpInfo` method to configure SNMP version 2 and version 3 on cluster nodes. The values you set with this interface apply to all nodes in the cluster, and the values that are passed replace all values set in any previous call to `SetSnmpInfo`.

Parameters

SetSnmpInfo is deprecated for Element versions 6.0 and later. Use the EnableSnmp and SetSnmpACL methods instead.

This method has the following input parameters:

Name	Description	Type	Default value	Required
networks	List of networks and what type of access they have to the SNMP servers running on the cluster nodes. See the SNMP network object for possible values. This parameter is required for SNMP v2 only.	network array	None	No
enabled	If set to true, SNMP is enabled on each node in the cluster.	boolean	false	No
snmpV3Enabled	If set to true, SNMP v3 is enabled on each node in the cluster.	boolean	false	No
usmUsers	If SNMP v3 is enabled, this value must be passed in place of the networks parameter. This parameter is required for SNMP v3 only.	usmUser	None	No

Return values

This method has no return values.

Request example with SNMP v3 enabled

Requests for this method are similar to the following example:

```
{
"method":"SetSnmpInfo",
"params":{
    "enabled":true,
    "snmpV3Enabled":true,
    "usmUsers":[
      {
        "name":"user1",
        "access":"rouser",
        "secLevel":"auth",
        "password":"namex1",
        "passphrase":"yourpassphrase"
      }
    ]
  },
 "id":1
}
```

Request example with SNMP v2 enabled

Requests for this method are similar to the following example:

```
{
"method":"SetSnmpInfo",
"params":{
    "enabled":true,
    "snmpV3Enabled":false,
    "networks":[
      {
        "community":"public",
        "access":"ro",
        "network":"localhost",
      }
    ]
  },
  "id":1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1
  "result" :{
  }
}
```

New since version

9.6

SetSnmpTrapInfo

You can use the `SetSnmpTrapInfo` method to enable and disable the generation of cluster SNMP notifications (traps) and to specify the set of hosts that receive the notifications. The values you pass with each `SetSnmpTrapInfo` method call replace all values set in any previous call.

Parameters

This method has the following input parameters:

Name	Description	Type
trapRecipients	List of hosts that are to receive the traps generated by the storage cluster. At least one object is required if any one of the trap types is enabled. This parameter is required only if any boolean parameters are set to true. (No default value. Not required.)	snmpTrapRecipient array
clusterFaultTrapsEnabled	If set to true, a corresponding cluster fault notification is sent to the configured list of trap recipients when a cluster fault is logged. (Default value: false. Not required.)	boolean
clusterFaultResolvedTrapsEnabled	If set to true, a corresponding cluster fault resolved notification is sent to the configured list of trap recipients when a cluster fault is resolved. (Default value: false. Not required.)	boolean

Name	Description	Type
clusterEventTrapsEnabled	If set to true, a corresponding cluster event notification is sent to the configured list of trap recipients when a cluster event is logged. (Default value: false. Not required.)	boolean

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
"method":"SetSnmpTrapInfo",
"params":{
    "clusterFaultTrapsEnabled":true,
    "clusterFaultResolvedTrapsEnabled":true,
    "clusterEventTrapsEnabled":true,
    "trapRecipients":[
    {
      "host":"192.30.0.10",
      "port":162,
      "community":"public"
    }
   ]
  },
  "id":1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id" : 1,
    "result" : {}
}
```

New since version

9.6

SetSSLCertificate

You can use the `SetSSLCertificate` method to set a user SSL certificate and private key for the storage nodes in the cluster.

After using the API, you must reboot the management node.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
certificate	The PEM-encoded text version of the certificate. **Note:** When setting a node or cluster certificate, the certificate must include the extendedKeyUsage extension for serverAuth. This extension allows the certificate to be used without error on common operating systems and browsers. If the extension is not present, the API will reject the certificate as invalid.	string	None	Yes
privateKey	The PEM-encoded text version of the private key.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "SetSSLCertificate",
    "params" : {
        "privateKey": "-----BEGIN RSA PRIVATE KEY-----
```

```
\nMIIEowIBAAKCAQEA8U+28fnLKQNWEWMR6akeDKuehSpS79odLGigI18qlCV/AUY5\nZLjqsT
jBvTJVRv44yoCTgNrx36U7FHP4t6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP
20HRYpZvY+Uz7LYEFCmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg\n6yjUGQO352ylc9HXHcn6lb
/jyl0DmVNUZ0caQwAmIS3Jmoyx+zj/Ya4WKq+2SqTA\nX7bX0F3wHHfXnZlHnM8fET5N/9A+K6
lS7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE\n76yAy6rTHu0xM3jjdkcb9Y8miNzxF+ACq+itaw
IDAQABAoIBAH1jlIZr6/sltqVW\nO0qVC/49dyNu+KWVSq92ti9rFe7hBPueh9gklh78hP9Qli
tLkir3YK4GFsTFUMux\n7z1NRCxA/4LrmLSkAjW2kRXDfVl2bwZq0ua9NefGw92O8D2OZvbuOx
k7Put2p6se\nfgNzSjf2SI5DIX3UMe5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7w
nq\nt+Am5/Vu1LI6rGiG6oHEW0oGSuHl1esIyXXa2hqkU+1+iF2iGRMTiXac4C8d11NU\nWGIR
CXFJAmsAQ+hQm7pmtsKdEqumj/PIoGXf0BoFVEWaIJIMEgnfuLZp8IelJQXn\nSFJbk2ECgYEA
+d5ooU4thZXylWHUZqomaxyzOruA1T53UeH69HiFTrLjvfwuaiqj\nlHzPlhms6hxexwz1dzAp
gog/NOM+2bAc0rn0dqvtV4doejtlDZKRqrNCf/cuN2QX\njaCJClCWau3sEHCckLOhWeY4HaPS
oWq0GKLmKkKDChB4nWUYg3gSWQkCgYEA9zuN\nHW8GPS+yjixeKXmkKO0x/vvxzR+J5HH5znaI
Hss48THyhzXpLr+v30Hy2h0yAlBS\nny5Ja6wsomb0mVe4NxVtVawg2E9vVvTa1UC+TNmFBBuL
RPfjcnjDerrSuQ5lYY+M\nC9MJtXGfhp//G0bzwsRzZxOBsUJb15tppaZIs9MCgYAJricpkKjM
0xlZ1jdvXsos\nPilnbho4qLngrzuUuxKXEPEnzBxUOqCpwQgdzZLYYw788TCVVIVXLEYem2s0
7dDA\nDTo+WrzQNkvC6IgqtXH1RgqegIoG1VbgQsbsYmDhdaQ+os4+AOeQXw3vgAhJ/qNJ\njQ
4Ttw3ylt7FYkRH26ACWQKBgQC74Zmf4JuRLAo5WSZFxpcmMvtnlvdutqUH4kXA\nzPssy6t+QE
La1fFbAXkZ5Pg1ITK752aiaX6KQNG6qRsA3VS1J6drD9/2AofOQU17\n+jOkGzmmoXf49Zj3iS
akwg0ZbQNGXNxEsCAUr0BYAobPp9/fB4PbtUs99fvtocFr\njS562QKBgCb+JMDP5q7jpUuspj
0obd/ZS+MsomE+gFAMBJ71KFQ7KuoNezNFO+ZE\n3rnR8AqAm4VMzqRahs2PWNe2Hl4J4hKu96
qNpNHbsW1NjXdAL9P7oqQIrhGLVdhX\nInDXvTgXMdMoet4BKnftelrXFKHgGqXJoczq4JWzGS
IHNgvkrH60\n-----END RSA PRIVATE KEY-----\n",
        "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD\nVQQGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G\nA1UEChMYV2hhdC
BIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1
UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGF
doYXQgSGFwcGVucyBpbiBWZWdh\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdm
VnYXNzdGF5c2luLnZl\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28f
nLKQNWEWMR\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FH
P4\nt6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nmrgp
GZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\n7dg9cyXu4afXcgKy14Ji
NBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrn
DJyTsdnEbaGBiaSBhjCBgzELMAkG\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxW
ZWdhcywgQmFieSExITAf\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqG
SIb3DQEJARYe\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYD
VR0T\nBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe\nMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N\niebEgAaE5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl+DwkxnS1cx1v
ERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDg
l0QoQDWNDoTeRBsz80nwiouA==\n-----END CERTIFICATE-----\n"
```

```
    },
    "id" : 2
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id" : 2,
    "result" : {}
}
```

New since version

10.0

SnmpSendTestTraps

`SnmpSendTestTraps` enables you to test SNMP functionality for a cluster. This method instructs the cluster to send test SNMP traps to the currently configured SNMP manager.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
status	Status of the test.	string

Request example

Requests for this method are similar to the following example:

```
{
    "method": "SnmpSendTestTraps",
    "params": {},
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "status": "complete"
    }
}
```

New since version

9.6

TestAddressAvailability

You can use the `TestAddressAvailability` method to check to see if a certain IP address is in use on an interface within the storage cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
interface	The target network interface (such as eth0, Bond10G, etc).	string	None	Yes
address	The IP address to scan for on the target interface.	string	None	Yes
virtualNetworkTag	The target VLAN ID.	integer	None	No
timeout	The timeout in seconds for testing the target address.	integer	5	No

Return values

This method has the following return values:

Name	Description	Type
address	The IP address tested.	string

available	True if the requested IP address is in use, and false if it is not.	boolean

Request example

Requests for this method are similar to the following example:

```
{
    "method": "TestAddressAvailability",
    "params": {
        "interface": "Bond10G",
        "address": "10.0.0.1",
        "virtualNetworkTag": 1234
    }
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "address":"10.0.0.1",
        "available": true
    }
}
```

New since version

11.0

Multitenant networking API methods

Multitenant networking in Element storage clusters allows traffic between multiple clients that are on separate logical networks to be connected to one Element storage cluster without layer 3 routing.

Connections to the storage cluster are segregated in the networking stack through the use of VLAN tagging.

Prerequisites for setting up a multitenant virtual network

- You must have identified the block of client network IP addresses to be assigned to the virtual networks on the storage nodes.
- You must have identified a client storage network IP (SVIP) address to be used as an endpoint for all

storage traffic.

Virtual networking order of operations

1. Use the AddVirtualNetwork method to bulk provision the IP addresses you enter.

   After you add a virtual network, the cluster automatically performs the following steps:

   - Each storage node creates a virtual network interface.
   - Each storage node is assigned a VLAN address that can be routed to using the virtual SVIP.
   - VLAN IP addresses persist on each node in the event of a node reboot.
2. When the virtual network interface and VLAN addresses have been assigned, you can assign client network traffic to the virtual SVIP.

Find more information

- Virtual network naming conventions
- AddVirtualNetwork
- ModifyVirtualNetwork
- ListVirtualNetworks
- RemoveVirtualNetwork
- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Virtual network naming conventions

NetApp Element storage systems use monotonically increasing numbers as unique identifiers for all objects in the system.

When you create a new volume, the new volumeID is an increment of exactly 1. This convention holds true with virtual networks in storage clusters running Element software. The first virtual network you create in an Element cluster has a VirtualNetworkID of 1. This ID is not the same thing as a VLAN tag number.

You can use VirtualNetworkID and the VirtualNetworkTag (VLAN tag) interchangeably where noted in the API methods.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddVirtualNetwork

You can use the `AddVirtualNetwork` method to add a new virtual network to a cluster configuration.

When you add a virtual network, an interface for each node is created and each interface requires a virtual network IP address. The number of IP addresses you specify as a parameter for this API method must be

equal to or greater than the number of nodes in the cluster. The system bulk provisions virtual network addresses and assigns them to individual nodes automatically. You do not need to assign virtual network addresses to nodes manually.

The AddVirtualNetwork method is used only to create a new virtual network. If you want to make changes to an existing virtual network, use the ModifyVirtualNetwork method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
addressBlocks	Unique range of IP addresses to include in the virtual network. Required members for the object: • start: The start of the IP address range. (string) • size: The number of IP addresses to include in the block. (integer)	JSON object array	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
gateway	The IP address of a gateway of the virtual network. This parameter is valid only if the namespace parameter is set to true.	string	None	No
name	A user-defined name for the new virtual network.	string	None	Yes

Name	Description	Type	Default value	Required
namespace	When set to true, enables the Routable Storage VLANs functionality by creating and configuring a namespace and the virtual network contained by it.	boolean	None	No
netmask	Unique network mask for the virtual network being created.	string	None	Yes
svip	Unique storage IP address for the virtual network being created.	string	None	Yes
virtualNetworkTag	A unique virtual network (VLAN) tag. Supported values are 1 through 4094.	integer	None	Yes

Note: Virtual network parameters must be unique to each virtual network when you set namespace to false.

Return value

This method has the following return value:

Name	Description	Type
virtualNetworkID	The virtual network ID of the new virtual network.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "AddVirtualNetwork",
  "params": {
    "virtualNetworkTag": 2010,
    "name": "network1",
    "addressBlocks" : [
        { "start": "192.86.5.1", "size": 10 },
        { "start": "192.86.5.50", "size": 20 }
    ],
    "netmask" : "255.255.192.0",
    "gateway" : "10.0.1.254",
    "svip" : "192.86.5.200",
    "attributes" : {}
    "namespace" : true
  },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result":
      {
        "virtualNetworkID": 5
   }
}
```

New since version

9.6

ModifyVirtualNetwork

You can use the `ModifyVirtualNetwork` method to change the attributes of an existing virtual network.

This method enables you to add or remove address blocks, change the netmask, or modify the name or description of the virtual network. You can also use it to enable or disable namespaces, as well as add or remove a gateway if namespaces are enabled on the virtual network.

> This method requires either the virtualNetworkID or the virtualNetworkTag as a parameter, but not both.

CAUTION:

Enabling or disabling the Routable Storage VLANs functionality for an existing virtual network by changing the namespace parameter disrupts any traffic handled by the virtual network. It is best if you change the namespace parameter during a scheduled maintenance window.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
virtualNetworkID	Unique identifier of the virtual network to modify. This is the virtual network ID assigned by the cluster.	integer	None	No
virtualNetworkTag	The network tag that identifies the virtual network to modify.	integer	None	No

addressBlocks	The new address block to set for this virtual network. This might include new address blocks to add to the existing object or omit unused address blocks that need to be removed. Alternatively, you can extend or reduce the size of existing address blocks. You can only increase the size of the starting addressBlocks for a Virtual Network object; you can never decrease it. Required members for this object: • start: The start of the IP address range. (string) • size: The number of IP addresses to include in the block. (integer)	JSON object	None	No
gateway	The IP address of a gateway of the virtual network. This parameter is valid only if the namespace parameter is set to true.	string	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
name	The new name for the virtual network.	string	None	No

namespace	When set to true, enables the Routable Storage VLANs functionality by recreating the virtual network and configuring a namespace to contain it. When set to false, disables the VRF functionality for the virtual network. Changing this value disrupts traffic running through this virtual network.	boolean	None	No
netmask	New network mask for this virtual network.	string	None	No
svip	The storage virtual IP address for this virtual network. The SVIP for a virtual network cannot be changed. You must create a new virtual network to use a different SVIP address.	string	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyVirtualNetwork",
  "params": {
    "virtualNetworkID": 2,
    "name": "ESX-VLAN-3112",
    "addressBlocks": [
     {
      "start": "10.1.112.1",
      "size": 20
     },
     {
      "start": "10.1.112.100",
      "size": 20
     }
   ],
    "netmask": "255.255.255.0",
    "gateway": "10.0.1.254",
    "svip": "10.1.112.200",
    "attributes": {}
  },
  "id":1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
  }
}
```

New since version

9.6

ListVirtualNetworks

You can use the `ListVirtualNetworks` method to list all configured virtual networks for the cluster.

You can use this method to verify the virtual network settings in the cluster. There are no required parameters for this method. However, to filter the results, you can pass one or more virtualNetworkID or virtualNetworkTag values.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
virtualNetworkID	Network ID to filter the list for a single virtual network.	integer	None	No
virtualNetworkTag	Network tag to filter the list for a single virtual network.	integer	None	No
virtualNetworkIDs	Network IDs to include in the list.	integer array	None	No
virtualNetworkTags	Network tag to include in the list.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
virtualNetworks	Object containing virtual network IP addresses.	virtualNetwork

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVirtualNetworks",
  "params": {
     "virtualNetworkIDs": [5,6]
   },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
```

```
  "result": {
    "virtualNetworks": [
   {
    "addressBlocks": [
   {
    "available": "11000000",
    "size": 8,
    "start": "10.26.250.207"
   }
  ],
    "attributes": null,
    "gateway": "10.26.250.254",
    "name": "2250",
    "namespace": false,
    "netmask": "255.255.255.0",
    "svip": "10.26.250.200",
    "virtualNetworkID": 2250
   },
   {
     "addressBlocks": [
   {
     "available": "11000000",
     "size": 8,
     "start": "10.26.241.207"
   }
  ],
     "attributes": null,
     "gateway": "10.26.241.254",
     "name": "2241",
     "namespace": false,
     "netmask": "255.255.255.0",
     "svip": "10.26.241.200",
     "virtualNetworkID": 2241
   },
  {
     "addressBlocks": [
  {
     "available": "11000000",
     "size": 8,
     "start": "10.26.240.207"
   }
  ],
     "attributes": null,
     "gateway": "10.26.240.254",
     "name": "2240",
     "namespace": false,
```

```
      "netmask": "255.255.255.0",
      "svip": "10.26.240.200",
      "virtualNetworkID": 2240
     },
    {
   }
  ]
}
```

New since version

9.6

RemoveVirtualNetwork

You can use the `RemoveVirtualNetwork` method to remove a previously added virtual network.

> This method requires either the virtualNetworkID or the virtualNetworkTag as a parameter, but not both.

> You cannot remove a virtual network if there are initiators associated with it. Disassociate the initiators first, and then remove the virtual network.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
virtualNetworkID	Network ID that identifies the virtual network to remove.	integer	None	Yes
virtualNetworkTag	Network tag that identifies the virtual network to remove.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "RemoveVirtualNetwork",
  "params": {
     "virtualNetworkID": 5
    }
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {}
}
```

New since version

9.6

Volume API methods

Element software volume API methods enable you to manage volumes that reside on a storage node. You can create, modify, clone, and delete volumes with these methods. You can also use volume API methods to gather and display data measurements for a volume.

- CancelClone
- CancelGroupClone
- CloneMultipleVolumes
- CloneVolume
- CopyVolume
- CreateQoSPolicy
- CreateVolume
- CreateBackupTarget
- DeleteQoSPolicy
- DeleteVolume
- DeleteVolumes
- GetBackupTarget
- GetVolumeStats
- GetDefaultQoS

- GetQoSPolicy
- GetVolumeCount
- GetVolumeEfficiency
- ListActiveVolumes
- ListBackupTargets
- ListBulkVolumeJobs
- ListDeletedVolumes
- ListQoSPolicies
- ListSyncJobs
- ListVolumeQoSHistograms
- ListVolumes
- ListVolumeStats
- ListVolumesForAccount
- ListVolumeStatsByAccount
- ListVolumeStatsByVirtualVolume
- ListVolumeStatsByVolume
- ListVolumeStatsByVolumeAccessGroup
- ModifyBackupTarget
- ModifyQoSPolicy
- ModifyVolume
- ModifyVolumes
- PurgeDeletedVolume
- PurgeDeletedVolumes
- RemoveBackupTarget
- RestoreDeletedVolume
- SetDefaultQoS
- StartBulkVolumeRead
- StartBulkVolumeWrite
- UpdateBulkVolumeStatus

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

CancelClone

You can use the `CancelClone` method to stop an ongoing volume clone or volume copy process. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
cloneID	The cloneID for the ongoing clone process.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CancelClone",
   "params": {
      "cloneID" : 5,
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

9.6

CancelGroupClone

You can use the `CancelGroupClone` method to stop an ongoing clone process occurring on a group of volumes. When you cancel a group clone operation, the system completes and removes the operation's associated asyncHandle.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
groupCloneID	The cloneID for the ongoing clone process.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CancelGroupClone",
   "params": {
      "cloneID" : 5,
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {}
}
```

New since version

9.6

CloneMultipleVolumes

You can use the `CloneMultipleVolumes` method to create a clone of a group of specified volumes. You can assign a consistent set of characteristics to a group of multiple volumes when they are cloned together.

Before using the groupSnapshotID parameter to clone the volumes in a group snapshot, you must first create the group snapshot using the CreateGroupSnapshot API method or the web UI. Using groupSnapshotID is

optional when cloning multiple volumes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	New default access method for the new volumes if not overridden by information passed in the volume's array.	string	None	No
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • true • false	boolean	false	No
groupSnapshotID	ID of the group snapshot to use as a basis for the clone.	integer	None	No
newAccountID	New account ID for the volumes if not overridden by information passed in the volumes array.	integer	None	No

Name	Description	Type	Default value	Required
volumes	Collection of members that you specify for the new volumes. Members: • volumeID: (Required) • access: (Optional) Can be one of readOnly, readWrite, locked, or replicationTarget. • attributes: (Optional) List of name-value pairs in JSON object format. • name: (Optional) New name for the clone. • newAccountID: (Optional) Account ID for the new volumes. • newSize: (Optional) Total size of the volume, in bytes. Size is rounded up to the nearest megabyte. If optional members are not specified, the values are inherited from the source volumes.	JSON object array	None	Yes (volumeID)

Return values

This method has the following return values:

Name	Description	Type

asyncHandle	A value returned from an asynchronous method call.	integer
groupCloneID	Unique ID of the new group clone.	integer
members	List of volumeIDs for the source and destination volume pairs.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CloneMultipleVolumes",
   "params": {
       "volumes": [
           {
              "volumeID": 5
              "name":"foxhill",
              "access":"readOnly"
              },
           {
              "volumeID": 18
              },
           {
             "volumeID": 20
              }
     ]
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "asyncHandle": 12,
    "groupCloneID": 4,
    "members": [
     {
      "srcVolumeID": 5,
      "volumeID": 29
     },
     {
      "srcVolumeID": 18,
      "volumeID": 30
     },
     {
      "srcVolumeID": 20,
      "volumeID": 31
      }
    ]
  }
}
```


New since version

9.6

CloneVolume

You can use the `CloneVolume` method to create a copy of a volume. This method is asynchronous and might take a variable amount of time to complete.

The cloning process begins immediately when you make the `CloneVolume` request and is representative of the state of the volume when the API method is issued. You can use the GetAsyncResult method to determine when the cloning process is complete and the new volume is available for connections. You can use ListSyncJobs to see the progress of creating the clone. The initial attributes and quality of service settings for the volume are inherited from the volume being cloned. You can change these settings with ModifyVolume.

Cloned volumes do not inherit volume access group membership from the source volume.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
access	Access allowed for the new volume. If a value is not specified, the access value does not change. Possible values: • `readOnly`: (Optional) Only read operations are allowed. • `readWrite`: (Optional) Reads and writes are allowed. • `locked`: (Optional) No reads or writes are allowed. If not specified, the access value of the volume being cloned is used. • `replicationTarget`: (Optional) Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked.	string	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No

Name	Description	Type	Default value	Required
enable512e	Specifies whether the new volume should use 512-byte sector emulation. If unspecified, the setting of the volume being cloned is used.	boolean	Setting of original volume	No
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • true • false	boolean	false	No
name	Name of the new cloned volume; must be 1 to 64 characters in length.	string	None	Yes
newAccountID	AccountID for the owner of the new volume. If unspecified, the accountID of the owner of the volume being cloned is used.	integer	AccountID of the owner of original volume	No
newSize	New size of the volume, in bytes. Might be greater or less than the size of the volume being cloned. If not specified, the volume size is not changed. Size is rounded up to the nearest 1MB in size.	integer	None	No

Name	Description	Type	Default value	Required
snapshotID	ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.	integer	None	No
volumeID	VolumeID for the volume to be cloned.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	The handle value used to obtain the operation result.	integer
cloneID	The cloneID for the newly cloned volume.	integer
curve	The QoS curve values applied to the clone.	JSON object
volume	An object containing information about the newly cloned volume.	volume
volumeID	VolumeID for the newly cloned volume.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CloneVolume",
   "params": {
      "volumeID" : 5,
      "name"  : "mysqldata-snapshot1",
      "access" : "readOnly"
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
      "asyncHandle": 42,
      "cloneID": 37,
      "volume": {
          "access": "readOnly",
          "accountID": 1,
          "attributes": {},
          "blockSize": 4096,
          "createTime": "2016-03-31T22:26:03Z",
          "deleteTime": "",
          "enable512e": true,
          "iqn": "iqn.2010-01.com.solidfire:jyay.mysqldata-snapshot1.680",
          "name": "mysqldata-snapshot1",
          "purgeTime": "",
          "qos": {
              "burstIOPS": 100,
              "burstTime": 60,
              "curve": {
                  "4096": 100,
                  "8192": 160,
                  "16384": 270,
                  "32768": 500,
                  "65536": 1000,
                  "131072": 1950,
                  "262144": 3900,
                  "524288": 7600,
                  "1048576": 15000
              },
              "maxIOPS": 100,
              "minIOPS": 50
          },
          "scsiEUIDeviceID": "6a796179000002a8f47acc0100000000",
          "scsiNAADeviceID": "6f47acc1000000006a796179000002a8",
          "sliceCount": 0,
          "status": "init",
          "totalSize": 1000341504,
          "virtualVolumeID": null,
          "volumeAccessGroups": [],
          "volumeID": 680,
          "volumePairs": []
```

```
        },
        "volumeID": 680
    }
}
```

New since version

9.6

Find more information

- GetAsyncResult
- ListSyncJobs
- ModifyVolume

CopyVolume

You can use the `CopyVolume` method to overwrite the data contents of an existing volume with the data contents of another volume (or snapshot). Attributes of the destination volume such as IQN, QoS settings, size, account, and volume access group membership are not changed. The destination volume must already exist and must be the same size as the source volume.

It is best if clients unmount the destination volume before the operation begins. If the destination volume is modified during the operation, the changes are lost. This operation can take a variable amount of time to complete. You can use the GetAsyncResult method to determine when the process has finished, and ListSyncJobs to see the progress of the copy.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
dstVolumeID	VolumeID of the volume to overwrite.	integer	None	Yes
volumeID	VolumeID of the volume to be read from.	integer	None	Yes
snapshotID	ID of the snapshot that is used as the source of the clone. If no ID is provided, the current active volume is used.	integer	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	Handle value used to obtain the operation result.	integer
cloneID	CloneID for the newly cloned volume.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CopyVolume",
   "params": {
      "volumeID" : 3,
      "dstVolumeID" : 2
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "asyncHandle": 9,
    "cloneID": 5
  }
}
```

New since version

9.6

Find more information

- GetAsyncResult
- ListSyncJobs

CreateQoSPolicy

You can use the `CreateQoSPolicy` method to create a QoSPolicy object that you can later apply to a volume upon creation or modification. A QoS policy has a unique ID, a name, and QoS settings.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	The name of the QoS policy; for example, gold, platinum, or silver.	string	None	Yes
qos	The QoS settings that this policy represents.	QoS	None	Yes

Return value

This method has the following return value:

Name	Description	Type
qosPolicy	The newly created QoSPolicy object.	QoSPolicy

Request example

Requests for this method are similar to the following example:

```
{
  "id": 68,
  "method": "CreateQoSPolicy",
  "params": {
    "name": "bronze",
    "qos": {
      "minIOPS": 50,
      "maxIOPS": 15000,
      "burstIOPS": 15000
    }
  }
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 68,
  "result": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15000,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 15000,
        "minIOPS": 50
      },
      "qosPolicyID": 2,
      "volumeIDs": []
    }
  }
}
```


New since version

10.0

CreateVolume

You can use the `CreateVolume` method to create a new, empty volume on the cluster. As soon as the volume is created, the volume is available for connection via iSCSI.

Volumes created without specified QoS values use the default values. You can view default values for a volume by using the `GetDefaultQoS` method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`access`	The access mode for the volume. If this parameter is included, the only supported value is `snapMirrorTarget`.	string	None	No
`accountID`	The ID of the account that owns this volume.	integer	None	Yes
`associateWithQoSPolicy`	Associate the volume with the specified QoS policy. Possible values: • `true`: Associate the volume with the QoS policy specified in the QoSPolicyID parameter. • `false`: Do not associate the volume with the QoS policy specified in the QoSPolicyID parameter. When false, any existing policy association is removed, regardless of whether you specify a QoS policy in the QoSPolicy parameter.	boolean	true	No
`attributes`	List of name-value pairs in JSON object format. The total attribute size must be less than 1000B, or 1KB, including JSON formatting characters.	JSON object	None	No

Name	Description	Type	Default value	Required
`enable512e`	Enable 512-byte sector emulation. Possible values: • `true`: The volume provides 512-byte sector emulation. • `false`: 512e emulation is not enabled.	boolean	None	Yes
`enableSnapMirrorReplication`	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	false	No
`fifoSize`	Specifies the maximum number of First-In-First-Out (FIFO) snapshots supported by the volume. Note that FIFO and non-FIFO snapshots both use the same pool of available snapshot slots on a volume. Use this option to limit FIFO snapshot consumption of the available snapshot slots. If omitted, the value defaults to 24.	integer	24	No

Name	Description	Type	Default value	Required
`minFifoSize`	Specifies the minimum number of First-In-First-Out (FIFO) snapshot slots reserved by the volume. This guarantees that if you are using both FIFO snapshots and non-FIFO snapshots on a volume that the non-FIFO snapshots do not unintentionally consume too many FIFO slots. It also ensures that at least this many FIFO snapshots are always available. Since FIFO and non-FIFO snapshots share the same pool, the `minFifoSize` reduces the total number of possible non-FIFO snapshots by the same amount. If omitted, the value defaults to 0.	integer	0	No
`name`	Name of the volume access group (may be user-specified). Not required to be unique, but recommended. Must be 1 to 64 characters in length.	string	None	Yes
`qos`	The initial quality of service settings for this volume. Default values are used if none are specified. Possible values: • `minIOPS` • `maxIOPS` • `burstIOPS`	QoS object	None	No

Name	Description	Type	Default value	Required
`qosPolicyID`	The ID for the policy whose QoS settings should be applied to the specified volumes. This parameter is mutually exclusive with the `qos` parameter.	integer	None	No
`totalSize`	Total size of the volume, in bytes. Size is rounded up to the nearest megabyte.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
volume	Object containing information about the newly created volume.	volume
volumeID	The volumeID for the newly created volume.	integer
curve	The curve is a set of key-value pairs. The keys are the I/O sizes in bytes. The values represent the cost of performing an IOP at a specific I/O size. The curve is calculated relative to a 4096 byte operation set at 100 IOPS.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CreateVolume",
   "params": {
      "name": "mysqldata",
      "accountID": 1,
      "totalSize": 107374182400,
      "enable512e": false,
      "attributes": {
         "name1": "value1",
         "name2": "value2",
         "name3": "value3"
      },
      "qos": {
         "minIOPS": 50,
         "maxIOPS": 500,
         "burstIOPS": 1500,
         "burstTime": 60
      }
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "volume": {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {
                "name1": "value1",
```

```
                "name2": "value2",
                "name3": "value3"
            },
            "blockSize": 4096,
            "createTime": "2016-03-31T22:20:22Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:mysqldata.677",
            "name": "mysqldata",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 500,
                "minIOPS": 50
            },
            "scsiEUIDeviceID": "6a796179000002a5f47acc0100000000",
            "scsiNAADeviceID": "6f47acc1000000006a796179000002a5",
            "sliceCount": 0,
            "status": "active",
            "totalSize": 107374182400,
            "virtualVolumeID": null,
            "volumeAccessGroups": [],
            "volumeID": 677,
            "volumePairs": []
        },
        "volumeID": 677
    }
}
```


New since version

9.6

Find more information

GetDefaultQoS

CreateBackupTarget

You can use `CreateBackupTarget` to create and store backup target information so that you do not need to re-enter it each time a backup is created.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	Name for the backup target.	string	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	Yes (but can be empty)

Return value

This method has the following return value:

Name	Description	Type
backupTargetID	Unique identifier assigned to the new backup target.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CreateBackupTarget",
   "params": {
      "name": "mytargetbackup"
   },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
      "backupTargetID": 1
   }
}
```

New since version

9.6

DeleteQoSPolicy

You can use the `DeleteQoSPolicy` method to delete a QoS policy from the system. The QoS settings for all volumes created or modified with this policy are unaffected.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
qosPolicyID	The ID of the QoS policy to be deleted.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "id": 663,
  "method": "DeleteQoSPolicy",
  "params": {
    "qosPolicyID": 4
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 663,
  "result": {}
}
```

New since version

9.6

DeleteVolume

You can use the `DeleteVolume` method to mark an active volume for deletion. When marked, the volume is purged (permanently deleted) after the cleanup interval elapses.

After making a request to delete a volume, any active iSCSI connections to the volume are immediately terminated and no further connections are allowed while the volume is in this state. A marked volume is not returned in target discovery requests.

Any snapshots of a volume that has been marked for deletion are not affected. Snapshots are kept until the volume is purged from the system. If a volume is marked for deletion and has a bulk volume read or bulk volume write operation in progress, the bulk volume read or write operation is stopped.

If the volume you delete is paired with a volume, replication between the paired volumes is suspended and no data is transferred to it or from it while in a deleted state. The remote volume the deleted volume was paired with enters into a PausedMisconfigured state and data is no longer sent to it or from the deleted volume. Until the deleted volume is purged, it can be restored and data transfers resume. If the deleted volume gets purged from the system, the volume it was paired with enters into a StoppedMisconfigured state and the volume pairing status is removed. The purged volume becomes permanently unavailable.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	The ID of the volume to delete.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
volume	Object containing information about the deleted volume.	volume
volumeID	The volumeID of the deleted volume.	integer

curve	The curve is a set of key-value pairs. The keys are the I/O sizes in bytes. The values represent the cost of performing an IOP at a specific I/O size. The curve is calculated relative to a 4096 byte operation set at 100 IOPS.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteVolume",
   "params": {
      "volumeID"  : 5
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volume": {
      "access": "readWrite",
      "accountID": 1,
      "attributes": {
        "name1": "value1",
        "name2": "value2",
        "name3": "value3"
      },
      "blockSize": 4096,
      "createTime": "2016-03-28T16:16:13Z",
      "deleteTime": "2016-03-31T22:59:42Z",
      "enable512e": true,
      "iqn": "iqn.2010-01.com.solidfire:jyay.1459181777648.5",
      "name": "1459181777648",
      "purgeTime": "2016-04-01T06:59:42Z",
      "qos": {
        "burstIOPS": 150,
        "burstTime": 60,
```

```
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 100,
        "minIOPS": 60
      },
      "scsiEUIDeviceID": "6a79617900000005f47acc0100000000",
      "scsiNAADeviceID": "6f47acc1000000006a79617900000005",
      "sliceCount": 1,
      "status": "deleted",
      "totalSize": 1000341504,
      "virtualVolumeID": null,
      "volumeAccessGroups": [
        1
      ],
      "volumeID": 5,
      "volumePairs": []
    }
  }
}
```


New since version

9.6

DeleteVolumes

You can use the `DeleteVolumes` method to mark multiple (up to 500) active volumes for deletion. When marked, the volume is purged (permanently deleted) after the cleanup interval elapses.

After making a request to delete volumes, any active iSCSI connections to the volumes are immediately terminated and no further connections are allowed while the volumes are in this state. A marked volume is not returned in target discovery requests.

Any snapshots of a volume that has been marked for deletion are not affected. Snapshots are kept until the volume is purged from the system. If a volume is marked for deletion and has a bulk volume read or bulk volume write operation in progress, the bulk volume read or write operation is stopped.

If the volumes you delete are paired with a volume, replication between the paired volumes is suspended and

no data is transferred to them or from them while in a deleted state. The remote volumes the deleted volumes were paired with enter into a PausedMisconfigured state and data is no longer sent to them or from the deleted volumes. Until the deleted volumes are purged, they can be restored and data transfers resume. If the deleted volumes are purged from the system, the volumes they were paired with enter into a StoppedMisconfigured state and the volume pairing status is removed. The purged volumes become permanently unavailable.

Parameters

This method has the following input parameters.

> At least one of the following parameters are required, and you must use only one of the parameters (they are all mutually exclusive with one another).

Name	Description	Type	Default value	Required
volumeIDs	The list of IDs of the volumes to delete from the system.	integer array	None	See Note.
volumeAccessGroupIDs	A list of volume access group IDs. All of the volumes from all of the volume access groups you specify in this list are deleted from the system.	integer array	None	See Note.
accountIDs	A list of account IDs. All volumes from these accounts are deleted from the system.	integer array	None	See Note.

Return values

This method has the following return values:

Name	Description	Type
volumes	Information about the newly deleted volume.	volume
curve	The curve is a set of key-value pairs. The keys are the I/O sizes in bytes. The values represent the cost of performing an IOP at a specific I/O size. The curve is calculated relative to a 4096 byte operation set at 100 IOPS.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteVolumes",
   "params": {
      "accountIDs" : [1, 2, 3]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {
    "volumes" : [ {
      "access": "readWrite",
      "accountID": 1,
      "attributes": {},
      "blockSize": 4096,
      "createTime": "2015-03-06T18:50:56Z",
      "deleteTime": "",
      "enable512e": False,
      "iqn": "iqn.2010-01.com.solidfire:pzsr.vclient-030-v00001.1",
      "name": "vclient-030-v00001",
      "qos": {
        "burstIOPS": 15000,
        "burstTime": 60,
        "curve": {},
        "maxIOPS": 15000,
        "minIOPS": 100
      },
      "purgeTime": "",
      "sliceCount": 1,
      "scsiEUIDeviceID": "707a737200000001f47acc0100000000",
      "scsiNAADeviceID": "6f47acc100000000707a737200000001",
      "status": "active",
      "totalSize": 10000003072,
      "virtualVolumeID": 5,
      "volumeAccessGroups": [],
      "volumePairs": [],
      "volumeID": 1
    } ]
  }
}
```


New since version

9.6

GetBackupTarget

You can use the `GetBackupTarget` method to return information about a specific backup target that you have created.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
backupTargetID	Unique identifier assigned to the backup target.	integer	None	Yes
name	Name of the backup target.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
backupTarget	List of name-value pairs in JSON object format.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "GetBackupTarget",
  "params": {
      "backupTargetID": 1
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "backupTarget": {
     "attributes" : {
        "size" : 100
     },
     "backupTargetID" : 1,
     "name" : "mytargetbackup"
    }
  }
}
```

New since version

9.6

GetVolumeStats

You can use the `GetVolumeStats` method to get high-level activity measurements for a single volume. Values are cumulative from the creation of the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	Specifies the volume for which statistics are gathered.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumeStats	Volume activity information.	volumeStats

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetVolumeStats",
   "params": {
      "volumeID": 32
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": {
      "accountID": 1,
      "actualIOPS": 0,
      "asyncDelay": null,
      "averageIOPSize": 0,
      "burstIOPSCredit": 0,
      "clientQueueDepth": 0,
      "desiredMetadataHosts": null,
      "latencyUSec": 0,
      "metadataHosts": {
        "deadSecondaries": [],
        "liveSecondaries": [
          32
        ],
        "primary": 60
      },
      "nonZeroBlocks": 0,
      "readBytes": 0,
      "readBytesLastSample": 0,
      "readLatencyUSec": 0,
      "readOps": 0,
      "readOpsLastSample": 0,
      "samplePeriodMSec": 0,
      "throttle": 0,
      "timestamp": "2016-04-01T21:01:39.130840Z",
      "unalignedReads": 0,
      "unalignedWrites": 0,
      "volumeAccessGroups": [],
      "volumeID": 1,
      "volumeSize": 5000658944,
      "volumeUtilization": 0,
      "writeBytes": 0,
      "writeBytesLastSample": 0,
      "writeLatencyUSec": 0,
      "writeOps": 0,
      "writeOpsLastSample": 0,
      "zeroBlocks": 1220864
    }
  }
}
```

New since version

9.6

GetDefaultQoS

You can use the `GetDefaultQoS` method to get the default quality of service (QoS) values for a newly created volume.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
QoS	The default QoS values.	QoS

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetDefaultQoS",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "burstIOPS" : 15000,
      "burstTime" : 60,
      "curve" : {
         "1048576" : 15000,
         "131072" : 1900,
         "16384" : 270,
         "262144" : 3000,
         "32768" : 500,
         "4096" : 100,
         "524288" : 7500,
         "65536" : 1000,
         "8192" : 160
      },
      "maxIOPS" : 15000,
      "minIOPS" : 100
   }
}
```


New since version

9.6

GetQoSPolicy

You can use the `GetQoSPolicy` method to get details about a specific QoS policy from the system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
qosPolicyID	The ID of the policy to be retrieved.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
qosPolicy	Details of the requested QoS policy.	QoSPolicy

Request example

Requests for this method are similar to the following example:

```
{
    "method": "GetQoSPolicy",
    "params": {
        "qosPolicyID": 2
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15002,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 15002,
        "minIOPS": 51
      },
      "qosPolicyID": 2,
      "volumeIDs": [
          2
      ]
    }
  }
}
```


New since version

10.0

GetVolumeCount

You can use the `GetVolumeCount` method to get the number of volumes currently in the system.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
count	The number of volumes currently in the system.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetVolumeCount",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "count": 7
  }
}
```

New since version

9.6

GetVolumeEfficiency

You can use the `GetVolumeEfficiency` method to get information about a volume. Only the volume you give as a parameter in this API method is used to compute the capacity.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	Specifies the volume for which capacity is computed.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
compression	The amount of space being saved by compressing data on a single volume. Stated as a ratio, where 1 means data has been stored without being compressed.	float
deduplication	The amount of space being saved on a single volume by not duplicating data. Stated as a ratio.	float
missingVolumes	The volumes that could not be queried for efficiency data. Missing volumes can be caused by Garbage Collection (GC) being less than an hour old, temporary network loss or restarted services since the GC cycle.	integer array
thinProvisioning	The ratio of space used to the amount of space allocated for storing data. Stated as a ratio.	float
timestamp	The last time efficiency data was collected after GC.	ISO 8601 data string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetVolumeEfficiency",
   "params": {
      "volumeID": 606
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
       "compression": 2.001591240821456,
       "deduplication": 1,
       "missingVolumes": [],
       "thinProvisioning": 1.009861932938856,
       "timestamp": "2014-03-10T16:06:33Z"
   }
}
```

New since version

9.6

ListActiveVolumes

You can use the `ListActiveVolumes` method to get the list of active volumes currently in the system. The list of volumes is sorted in VolumeID order and can be returned in multiple parts (pages).

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response, by default. To exclude virtual volumes, set to false.	boolean	true	No
startVolumeID	Starting VolumeID to return. If no volume exists with this VolumeID, the next volume by VolumeID order is used as the start of the list. To page through the list, pass the VolumeID of the last volume in the previous response + 1.	integer	0	No

Name	Description	Type	Default value	Required
limit	Maximum number of volume info objects to return. 0 (zero) returns all volumes (unlimited).	integer	(unlimited)	No

Return value

This method has the following return value:

Name	Description	Type
volumes	List of active volumes.	volume array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListActiveVolumes",
   "params": {
      "startVolumeID"  : 0,
      "limit" : 1000
   },
   "id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

ListBackupTargets

You can use the `ListBackupTargets` method to get information about all backup targets that have been created.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
backupTargets	Objects returned for each backup target. Included objects: • attributes: List of name-value pairs in JSON object format. (JSON object) • backupTargetID: Unique identifier assigned to the backup target. (integer) • name: Name of the backup target. (string)	JSON object

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListBackupTargets",
   "params": {},
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
     "backupTargets": [
       {
        "attributes" : {},
        "backupTargetID" : 1,
        "name" : "mytargetbackup"
       }
    ]
  }
}
```

New since version

9.6

ListBulkVolumeJobs

You can use the `ListBulkVolumeJobs` method to get information about each bulk volume read or write operation that is occurring in the system.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
bulkVolumeJobs	An array of information for each bulk volume job.	bulkVolumeJob array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListBulkVolumeJobs",
   "params": {
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "bulkVolumeJobs": [
        {
          "attributes": {
            "blocksPerTransfer": 1024,
            "firstPendingLba": 216064,
            "nLbas": 2441472,
            "nextLba": 226304,
            "pendingLbas": "[220160, 223232, 221184, 224256, 217088,
225280, 222208, 218112, 219136, 216064]",
            "percentComplete": 8,
            "startLba": 0
          },
          "bulkVolumeID": 2,
          "createTime": "2015-05-07T14:52:17Z",
          "elapsedTime": 44,
          "format": "native",
          "key": "eaffb0526d4fb47107061f09bfc9a806",
          "percentComplete": 8,
          "remainingTime": 506,
          "script": "bv_internal.py",
          "snapshotID": 509,
          "srcVolumeID": 3,
          "status": "running",
          "type": "read"
        }
      ]
   }
}
```


New since version

9.6

ListDeletedVolumes

You can use the `ListDeletedVolumes` method to retrieve the list of volumes that have been marked for deletion and purged from the system.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response, by default. To exclude virtual volumes, set to false.	boolean	true	No

Return value

This method has the following return value:

Name	Description	Type
volumes	List of deleted volumes.	volume array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListDeletedVolumes",
   "params": {},
   "id" : 1
}
```

Response example

Responses for this method are similar to the following example:

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 2,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2018-06-24T03:13:13Z",
                "deleteTime": "2018-07-22T16:12:39Z",
                "enable512e": true,
                "iqn": "iqn.2010-01.com.solidfire:0oto.deletethis.23",
                "name": "deleteThis",
                "purgeTime": "2016-07-23T00:12:39Z",
                "qos": {
                    "burstIOPS": 15000,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 15000,
                    "minIOPS": 50
                },
                "scsiEUIDeviceID": "306f746f00000017f47acc0100000000",
                "scsiNAADeviceID": "6f47acc100000000306f746f00000017",
                "sliceCount": 1,
                "status": "deleted",
                "totalSize": 1396703232,
                "virtualVolumeID": null,
                "volumeAccessGroups": [],
                "volumeID": 23,
                "volumePairs": []
            }
        ]
    }
}
```

New since version

9.6

ListQoSPolicies

You can use the `ListQoSPolicies` method to list the settings of all QoS policies on the system.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
qosPolicies	A list of details about each QoS policy.	QoSPolicy array

Request example

Requests for this method are similar to the following example:

```
{
  "id": 231,
  "method": "ListQoSPolicies",
  "params": {}
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 231,
  "result": {
    "qosPolicies": [
      {
        "name": "silver",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
            "4096": 100,
```

```
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 14000,
          "minIOPS": 50
        },
        "qosPolicyID": 1,
        "volumeIDs": [
          1
        ]
      },
      {
        "name": "bronze",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 15000,
          "minIOPS": 50
        },
        "qosPolicyID": 2,
        "volumeIDs": [
          2
        ]
      }
    ]
  }
}
```

New since version

10.0

ListSyncJobs

You can use the `ListSyncJobs` method to get information about synchronization jobs that are running on an Element storage cluster. This method returns information about slice, clone, block, and remote synchronization jobs.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
syncJobs	List of objects describing synchronization processes that are currently running in the system.	syncJob array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListSyncJobs",
   "params": { },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id":1,
    "result":{
      "syncJobs":[
        {
           "bytesPerSecond":275314.8834458956,
           "currentBytes":178257920,
           "dstServiceID":36,
           "elapsedTime":289.4568382049871,
           "percentComplete":8.900523560209423,
```

```
            "remainingTime":2962.675921065957,
            "sliceID":5,
            "srcServiceID":16,
            "stage":"whole",
            "totalBytes":2002780160,
            "type":"slice"
        },
        {
            "bytesPerSecond":305461.3198607744,
            "cloneID":1,
            "currentBytes":81788928,
            "dstServiceID":16,
            "dstVolumeID":6,
            "elapsedTime":291.7847648200743,
            "nodeID":1,
            "percentComplete":8.167539267015707,
            "remainingTime":3280.708270981153,
            "sliceID":6,
            "srcServiceID":16,
            "srcVolumeID":5,
            "stage":"whole",
            "totalBytes":1001390080,
            "type":"clone"
         },
         {
            "blocksPerSecond":0,
            "branchType": "snapshot",
            "dstServiceID":8,
            "dstVolumeID":2,
            "elapsedTime":0,
            "percentComplete":0,
            "remainingTime":0,
            "sliceID":2,
            "stage":"metadata",
            "type":"remote"
        }
      ]
    }
}
```


New since version

9.6

ListVolumeQoSHistograms

You can use the `ListVolumeQoSHistograms` method to generate a histogram of volume QoS usage for one volume or multiple volumes. This enables you to better understand how volumes are using QoS.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeIDs	An optional list of volume IDs specifying which volumes should have QoS histograms generated.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
qosHistograms	A list of objects describing volume usage for one or more volumes.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "ListVolumeQoSHistograms",
        "params": {
            "volumeIDs": [1]
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
```

```
        "qosHistograms": [
        {
            "histograms": {
            "belowMinIopsPercentages": {
                "Bucket1To19": 2406,
                "Bucket20To39": 3,
                "Bucket40To59": 0,
                "Bucket60To79": 4,
                "Bucket80To100": 0
            },
                "minToMaxIopsPercentages": {
                "Bucket101Plus": 0,
                "Bucket1To19": 0,
                "Bucket20To39": 0,
                "Bucket40To59": 2,
                "Bucket60To79": 0,
                "Bucket80To100": 0
            },
                "readBlockSizes": {
                "Bucket131072Plus": 0,
                "Bucket16384To32767": 0,
                "Bucket32768To65535": 0,
                "Bucket4096To8191": 0,
                "Bucket65536To131071": 0,
                "Bucket8192To16383": 0
            },
                "targetUtilizationPercentages": {
                "Bucket0": 134943,
                "Bucket101Plus": 0,
                "Bucket1To19": 2409,
                "Bucket20To39": 4,
                "Bucket40To59": 0,
                "Bucket60To79": 2,
                "Bucket80To100": 0
            },
                "throttlePercentages": {
                "Bucket0": 137358,
                "Bucket1To19": 0,
                "Bucket20To39": 0,
                "Bucket40To59": 0,
                "Bucket60To79": 0,
                "Bucket80To100": 0
            },
                "writeBlockSizes": {
                "Bucket131072Plus": 0,
                "Bucket16384To32767": 0,
```

```
                "Bucket32768To65535": 0,
                "Bucket4096To8191": 0,
                "Bucket65536To131071": 0,
                "Bucket8192To16383": 0
            }
        },
            "timestamp": "2018-06-21T18:45:52.010844Z",
            "volumeID": 1
        }
    ]
}
}
```


ListVolumes

You can use the `ListVolumes` method to get a list of volumes that are in a cluster. You can specify the volumes you want to return in the list by using the available parameters.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
accounts	Only volumes owned by the accounts you specify here are returned. Mutually exclusive with the volumeIDs parameter.	integer array	None	No
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No

Name	Description	Type	Default value	Required
isPaired	Returns volumes that are paired or not paired. Possible values: • true: Returns all paired volumes. • false: Returns all volumes not paired.	boolean	None	No
limit	Enables you to set the maximum number of volume results that are returned. Mutually exclusive with the volumeIDs parameter.	integer	10000	No
startVolumeID	Only volumes with an ID greater than or equal to this value are returned. Mutually exclusive with the volumeIDs parameter.	integer	None	No
volumeIDs	A list of volume IDs. If you specify this parameter, other parameters operate only on this set of volumes. Mutually exclusive with the accounts, startVolumeID, and limit parameters.	integer array	No	No
volumeName	Only volume object information matching the volume name is returned.	string	No	No

Name	Description	Type	Default value	Required
volumeStatus	Only volumes with a status equal to the status value are returned. Possible values: • creating • snapshotting • active • deleted	string	No	No

Return value

This method has the following return value:

Name	Description	Type
volumes	List of volumes.	volume array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumes",
    "params": {
        "volumeIDs": [1],
        "volumeStatus": "active",
        "isPaired": "false"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 1,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2016-03-28T14:39:05Z",
                "deleteTime": "",
                "enable512e": true,
                "iqn": "iqn.2010-01.com.solidfire:testvolume1.1",
                "name": "testVolume1",
                "purgeTime": "",
                "qos": {
                    "burstIOPS": 15000,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 15000,
                    "minIOPS": 50
                },
                "scsiEUIDeviceID": "6a79617900000001f47acc0100000000",
                "scsiNAADeviceID": "6f47acc1000000006a79617900000001",
                "sliceCount": 1,
                "status": "active",
                "totalSize": 5000658944,
                "virtualVolumeID": null,
                "volumeAccessGroups": [],
                "volumeID": 1,
                "volumePairs": []
            }
        ]
    }
}
```

New since version

9.6

ListVolumeStats

You can use the `ListVolumeStats` method to get high-level activity measurements for a single volume, list of volumes, or all volumes (if you omit the volumeIDs parameter). Measurement values are cumulative from the creation of the volume.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
volumeIDs	A list of volumes from which to retrieve activity information.	integer array	No	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumeStats",
    "params": {
        "volumeIDs": [1]
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 1,
        "actualIOPS": 0,
        "asyncDelay": null,
        "averageIOPSize": 0,
        "burstIOPSCredit": 30000,
        "clientQueueDepth": 0,
        "desiredMetadataHosts": null,
        "latencyUSec": 0,
        "metadataHosts": {
          "deadSecondaries": [],
          "liveSecondaries": [
            47
          ],
          "primary": 33
        },
        "nonZeroBlocks": 22080699,
        "readBytes": 657262370816,
        "readBytesLastSample": 0,
        "readLatencyUSec": 0,
        "readOps": 160464446,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "throttle": 0,
        "timestamp": "2016-03-09T19:39:15.771697Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [
          1
        ],
        "volumeID": 1,
        "volumeSize": 107374182400,
        "volumeUtilization": 0,
        "writeBytes": 219117547520,
        "writeBytesLastSample": 0,
        "writeLatencyUSec": 0,
        "writeOps": 53495495,
        "writeOpsLastSample": 0,
```

```
          "zeroBlocks": 4133701
        }
      ]
    }
}
```

New since version

9.6

ListVolumesForAccount

You can use the `ListVolumesForAccount` method to list active and (pending) deleted volumes for an account.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
accountID	All volumes owned by this accountID are returned.	integer	No	Yes

Return value

This method has the following return value:

Name	Description	Type
volumes	List of volume information.	volume array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListVolumesForAccount",
   "params": {
      "accountID"  : 1
   },
   "id" : 1
}
```

Response example

Responses for this method are similar to the following example:

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 1,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2018-07-22T16:15:25Z",
                "deleteTime": "",
                "enable512e": false,
                "iqn": "iqn.2010-01.com.solidfire:0oto.test1.25",
                "name": "test1",
                "purgeTime": "",
                "qos": {
                    "burstIOPS": 15000,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 15000,
                    "minIOPS": 50
                },
                "scsiEUIDeviceID": "306f746f00000019f47acc0100000000",
                "scsiNAADeviceID": "6f47acc100000000306f746f00000019",
                "sliceCount": 1,
                "status": "active",
                "totalSize": 1000341504,
                "virtualVolumeID": null,
                "volumeAccessGroups": [],
                "volumeID": 25,
                "volumePairs": []
            }
        ]
    }
}
```

New since version

9.6

ListVolumeStatsByAccount

You can use the `ListVolumeStatsByAccount` method to list high-level volume activity measurements for every account. Values are summed from all volumes owned by the account.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
accounts	A list of account IDs for which to return volume statistics. If omitted, statistics for all accounts are returned.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information for each account.**Note:** The volumeID member is 0 for each entry, as the values represent the summation of all volumes owned by the account.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
"method": "ListVolumeStatsByAccount",
  "params": {"accounts": [3]},
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 3,
        "nonZeroBlocks": 155040175,
        "readBytes": 3156273328128,
        "readBytesLastSample": 0,
        "readOps": 770574543,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "timestamp": "2016-10-17T20:42:26.231661Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [],
        "volumeID": 0,
        "volumeSize": 1127428915200,
        "writeBytes": 1051988406272,
        "writeBytesLastSample": 0,
        "writeOps": 256833107,
        "writeOpsLastSample": 0,
        "zeroBlocks": 120211025
      }
    ]
  }
}
```

New since version

9.6

ListVolumeStatsByVirtualVolume

You can use the `ListVolumeStatsByVirtualVolume` method to list volume statistics

for any volumes in the system that are associated with virtual volume. Statistics are cumulative from the creation of the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
virtualVolumeIDs	A list of one or more virtual volume IDs for which to retrieve information. If you specify this parameter, the method returns information about only these virtual volumes.	UUID string array	No	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	A list of objects containing activity information for each virtual volume in the system.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumeStatsByVirtualVolume",
  "params": {},
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
```

```
      {
        "accountID": 17,
        "actualIOPS": 0,
        "asyncDelay": null,
        "averageIOPSize": 1074265444,
        "burstIOPSCredit": 0,
        "clientQueueDepth": 0,
        "desiredMetadataHosts": null,
        "latencyUSec": 0,
        "metadataHosts": {
          "deadSecondaries": [],
          "liveSecondaries": [
            26
          ],
          "primary": 56
        },
        "nonZeroBlocks": 36,
        "readBytes": 18366464,
        "readBytesLastSample": 0,
        "readLatencyUSec": 0,
        "readOps": 156,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "throttle": 0,
        "timestamp": "2016-10-10T17:46:35.914642Z",
        "unalignedReads": 156,
        "unalignedWrites": 185,
        "virtualVolumeID": "070ac0ba-f344-4f4c-b79c-142efa3642e8",
        "volumeAccessGroups": [],
        "volumeID": 12518,
        "volumeSize": 91271200768,
        "volumeUtilization": 0,
        "writeBytes": 23652213248,
        "writeBytesLastSample": 0,
        "writeLatencyUSec": 0,
        "writeOps": 185,
        "writeOpsLastSample": 0,
        "zeroBlocks": 22282972
      }
    ]
  }
}
```


New since version

9.6

ListVolumeStatsByVolume

You can use the `ListVolumeStatsByVolume` method to list high-level activity measurements for every volume, by volume. Values are cumulative from the creation of the volume.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information.	volumeStats array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListVolumeStatsByVolume",
   "params": {},
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 3,
```

```
        "actualIOPS": 0,
        "asyncDelay": null,
        "averageIOPSize": 4096,
        "burstIOPSCredit": 30000,
        "clientQueueDepth": 0,
        "desiredMetadataHosts": null,
        "latencyUSec": 0,
        "metadataHosts": {
          "deadSecondaries": [],
          "liveSecondaries": [
              16
          ],
          "primary": 12
        },
        "nonZeroBlocks": 7499205,
        "readBytes": 159012818944,
        "readBytesLastSample": 0,
        "readLatencyUSec": 0,
        "readOps": 38821489,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "throttle": 0,
        "timestamp": "2016-10-17T20:55:31.087537Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [
          1
        ],
        "volumeID": 1,
        "volumeSize": 53687091200,
        "volumeUtilization": 0,
        "writeBytes": 52992585728,
        "writeBytesLastSample": 0,
        "writeLatencyUSec": 0,
        "writeOps": 12937643,
        "writeOpsLastSample": 0,
        "zeroBlocks": 5607995
      }
    ]
  }
}
```


New since version

9.6

ListVolumeStatsByVolumeAccessGroup

You can use the `ListVolumeStatsByVolumeAccessGroup` method to list total activity measurements for all of the volumes that are members of the specified volume access groups.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
includeVirtualVolumes	Virtual volumes are included in the response by default. To exclude virtual volumes, set to false.	boolean	true	No
volumeAccessGroups	An array of VolumeAccessGroupIDs for which volume activity is returned. If omitted, statistics for all volume access groups are returned.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
volumeStats	List of volume activity information for all volumes in the specified volume access group.**Note:** The volumeID member is 0 for each entry, because the values represent the summation of all volumes owned by the account.	volumeStats

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVolumeStatsByVolumeAccessGroup",
    "params": {"volumeAccessGroups": [1]},
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeStats": [
      {
        "accountID": 0,
        "nonZeroBlocks": 149366393,
        "readBytes": 3156273328128,
        "readBytesLastSample": 0,
        "readOps": 770574543,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "timestamp": "2016-10-17T21:04:10.712370Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [
            1
        ],
        "volumeID": 0,
        "volumeSize": 1073741824000,
        "writeBytes": 1051988406272,
        "writeBytesLastSample": 0,
        "writeOps": 256833107,
        "writeOpsLastSample": 0,
        "zeroBlocks": 112777607
      }
    ]
  }
}
```

New since version

9.6

ModifyBackupTarget

You can use the `ModifyBackupTarget` method to change attributes of a backup target.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
backupTargetID	Unique target ID for the target to modify.	integer	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
name	New name for the backup target.	string	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ModifyBackupTarget",
   "params": {
      "backupTargetID" : 1,
      "name": "yourtargetS3"
      "attributes" : {
          "size" : 500,
      }
   },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {}
}
```

New since version

9.6

ModifyQoSPolicy

You can use the `ModifyQoSPolicy` method to modify an existing QoS policy on the system.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
qosPolicyID	The ID of the policy to be modified.	integer	None	Yes
name	If supplied, the name of the QoS policy (e.g. gold, platinum, silver) is changed to this value.	string	None	No
qos	If supplied, the QoS settings for this policy are changed to these settings. You can supply partial QoS values and only change some of the QoS settings.	QoS object	None	No

Return values

This method has the following return values:

Name	Description	Type
qosPolicy	Details of the newly modified QoS policy.	QoSPolicy

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1950,
  "method": "ModifyQoSPolicy",
  "params": {
    "qosPolicyID": 2,
    "qos": {
      "minIOPS": 51,
      "maxIOPS": 15002,
      "burstIOPS": 15002
    }
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1950,
  "result": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15002,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 15002,
        "minIOPS": 51
      },
      "qosPolicyID": 2,
      "volumeIDs": [
        2
      ]
    }
  }
}
```


New since version

10.0

ModifyVolume

You can use the `ModifyVolume` method to modify settings on an existing volume. You can make modifications to one volume at a time and changes take place immediately.

If you do not specify QoS values when you modify a volume, they remain the same as before the modification. You can retrieve default QoS values for a newly created volume by running the `GetDefaultQoS` method.

When you need to increase the size of a volume that is being replicated, do so in the following order to prevent replication errors:

1. Increase the size of the volume with replicationTarget access.

2. Increase the size of the source or the volume with readWrite access.

Ensure that both the target and source volumes are the same size.

If you change the access status to locked or replicationTarget, all existing iSCSI connections are terminated.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeID	The volumeID for the volume to be modified.	integer	None	Yes

Name	Description	Type	Default value	Required
access	Access allowed for the volume. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. If not specified, the access value does not change. • `replicationTarget`: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked. If a value is not specified, the access value does not change. • `snapMirrorTarget`: Identify a volume as the target volume for SnapMirror replication.	string	None	No
accountID	The accountID to which the volume is reassigned. If none is specified, the previous account name is used.	integer	None	No

Name	Description	Type	Default value	Required
associateWithQoSPolicy	Associate the volume with the specified QoS policy. Possible values: • `true`: Associate the volume with the QoS policy specified in the QoSPolicyID parameter. • `false`: Do not associate the volume with the QoS policy specified in the QoSPolicyID parameter. When false, any existing policy association is removed, regardless of whether you specify a QoS policy in the QoSPolicy parameter.	boolean	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
createTime	An ISO 8601 date string to set as the new volume creation date. Required if setCreateTime is set to true.	ISO 8601 string	None	No

Name	Description	Type	Default value	Required
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	false	No
fifoSize	Specifies the maximum number of First-In-First-Out (FIFO) snapshots supported by the volume. Note that FIFO and non-FIFO snapshots both use the same pool of available snapshot slots on a volume. Use this option to limit FIFO snapshot consumption of the available snapshot slots. Note that you cannot modify this value to be less than the current FIFO snapshot count.	integer	None	No
minFifoSize	Specifies the number of snapshot slots that are reserved for only First-In-First-Out (FIFO) snapshots. Since FIFO and non-FIFO snapshots share the same pool, the minFifoSize parameter reduces the total number of possible non-FIFO snapshots by the same amount. Note that you cannot modify this value so that it conflicts with the current non-FIFO snapshot count.	integer	None	No

Name	Description	Type	Default value	Required
mode	Volume replication mode. Possible values: • `asynch`: Waits for system to acknowledge that data is stored on source before writing to the target. • `sync`: Does not wait for data transmission acknowledgment from source to begin writing data to the target.	string	None	No
qos	The new quality of service settings for this volume. If not specified, the QoS settings are not changed. Possible values: • `minIOPS` • `maxIOPS` • `burstIOPS`	QoS	None	No
qosPolicyID	The ID for the policy whose QoS settings should be applied to the specified volumes. This parameter is mutually exclusive with the qos parameter.	integer	None	No
setCreateTime	Set to true to change the recorded date of volume creation.	boolean	None	No

Name	Description	Type	Default value	Required
totalSize	The new size of the volume in bytes. 1000000000 is equal to 1GB. Size is rounded up to the nearest megabyte in size. This parameter can only be used to increase the size of a volume.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
volume	Object containing information about the newly modified volume.	volume

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyVolume",
  "params": {
     "volumeID": 5,
     "attributes": {
        "name1": "value1",
        "name2": "value2",
        "name3": "value3"
     },
     "qos": {
        "minIOPS": 60,
        "maxIOPS": 100,
        "burstIOPS": 150,
        "burstTime": 60
     },
      "access" :"readWrite"
     },
      "totalSize": 20000000000,
     "id": 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
      "volume": {
          "access": "readWrite",
          "accountID": 1,
          "attributes": {
              "name1": "value1",
              "name2": "value2",
              "name3": "value3"
          },
          "blockSize": 4096,
          "createTime": "2016-03-28T16:16:13Z",
          "deleteTime": "",
          "enable512e": true,
          "iqn": "iqn.2010-01.com.solidfire:jyay.1459181777648.5",
          "name": "1459181777648",
          "purgeTime": "",
          "qos": {
              "burstIOPS": 150,
              "burstTime": 60,
              "curve": {
                  "4096": 100,
                  "8192": 160,
                  "16384": 270,
                  "32768": 500,
                  "65536": 1000,
                  "131072": 1950,
                  "262144": 3900,
                  "524288": 7600,
                  "1048576": 15000
              },
              "maxIOPS": 100,
              "minIOPS": 60
          },
          "scsiEUIDeviceID": "6a79617900000005f47acc0100000000",
          "scsiNAADeviceID": "6f47acc1000000006a79617900000005",
          "sliceCount": 1,
          "status": "active",
          "totalSize": 1000341504,
          "virtualVolumeID": null,
          "volumeAccessGroups": [
```

```
                1
            ],
            "volumeID": 5,
            "volumePairs": []
        }
    }
}
```

New since version

9.6

Find more information

GetDefaultQoS

ModifyVolumes

You can use the `ModifyVolumes` method to configure up to 500 existing volumes at one time. Changes take place immediately. If `ModifyVolumes` fails to modify any of the specified volumes, none of the specified volumes are changed.

If you do not specify QoS values when you modify volumes, the QoS values for each volume remain unchanged. You can retrieve default QoS values for a newly created volume by running the `GetDefaultQoS` method.

When you need to increase the size volumes that are being replicated, do so in the following order to prevent replication errors:

1. Increase the size of the volume with replicationTarget access.
2. Increase the size of the source or the volume with readWrite access.

Ensure that both the target and source volumes are the same size.

If you change the access status to locked or replicationTarget, all existing iSCSI connections are terminated.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required

access	Access allowed for the volumes. Possible values: • `readOnly`: Only read operations are allowed. • `readWrite`: Reads and writes are allowed. • `locked`: No reads or writes are allowed. If not specified, the access value does not change. • `replicationTarget`: Identify a volume as the target volume for a paired set of volumes. If the volume is not paired, the access status is locked. If a value is not specified, the access value does not change.	string	None	No
accountID	The accountID to which the volumes are reassigned. If none is specified, the previous account name is used.	integer	None	No

associateWithQoSPolicy	Associate the volume with the specified QoS policy. Possible values: • true: Associate the volume with the QoS policy specified in the QoSPolicyID parameter. • false: Do not associate the volume with the QoS policy specified in the QoSPolicyID parameter. When false, any existing policy association is removed, regardless of whether you specify a QoS policy in the QoSPolicy parameter.	boolean	None	No
attributes	List of name-value pairs in JSON object format.	JSON object	None	No
createTime	An ISO 8601 date string to set as the new volume creation date. Required if setCreateTime is set to true.	ISO 8601 string	None	No
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	false	No

fifoSize	Specifies the maximum number of First-In-First-Out (FIFO) snapshots supported by the volume. Note that FIFO and non-FIFO snapshots both use the same pool of available snapshot slots on a volume. Use this option to limit FIFO snapshot consumption of the available snapshot slots. Note that you cannot modify this value to be less than the current FIFO snapshot count.	integer	None	No
minFifoSize	Specifies the number of snapshot slots that are reserved for only First-In-First-Out (FIFO) snapshots. Since FIFO and non-FIFO snapshots share the same pool, the minFifoSize parameter reduces the total number of possible non-FIFO snapshots by the same amount. Note that you cannot modify this value so that it conflicts with the current non-FIFO snapshot count.	integer	None	No

mode	Volume replication mode. Possible values: • `asynch`: Waits for system to acknowledge that data is stored on source before writing to the target. • `sync`: Does not wait for data transmission acknowledgment from source to begin writing data to the target.	string	None	No
qos	The new quality of service settings for the volumes. If not specified, the QoS settings are not changed. Possible values: • `minIOPS` • `maxIOPS` • `burstIOPS`	QoS	None	No
qosPolicyID	The ID for the policy whose QoS settings should be applied to the specified volumes. This parameter is mutually exclusive with the qos parameter.	integer	None	No
setCreateTime	Set to true to change the recorded date of volume creation.	boolean	None	No

totalSize	The new size of the volumes in bytes. 1000000000 is equal to 1GB. Size is rounded up to the nearest megabyte in size. This parameter can only be used to increase the size of a volume.	integer	None	No
volumeIDs	A list of volumeIDs for the volumes to be modified.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volume	An array of objects containing information about each newly modified volume.	volume array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyVolumes",
  "params": {
    "volumeIDs": [2,3],
    "attributes": {
      "name1": "value1",
      "name2": "value2",
      "name3": "value3"
    },
    "qos": {
      "minIOPS": 50,
      "maxIOPS": 100,
      "burstIOPS": 150,
      "burstTime": 60
    },
    "access" : "replicationTarget"
  },
  "totalSize": 80000000000,
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumes": [
      {
        "access": "replicationTarget",
        "accountID": 1,
        "attributes": {
          "name1": "value1",
          "name2": "value2",
          "name3": "value3"
        },
        "blockSize": 4096,
        "createTime": "2016-04-06T17:25:13Z",
        "deleteTime": "",
        "enable512e": false,
        "iqn": "iqn.2010-01.com.solidfire:jo73.2",
        "name": "doctest1",
        "purgeTime": "",
        "qos": {
```

```
        "burstIOPS": 150,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 100,
        "minIOPS": 50
      },
      "scsiEUIDeviceID": "6a6f373300000002f47acc0100000000",
      "scsiNAADeviceID": "6f47acc1000000006a6f373300000002",
      "sliceCount": 1,
      "status": "active",
      "totalSize": 1000341504,
      "virtualVolumeID": null,
      "volumeAccessGroups": [],
      "volumeID": 2,
      "volumePairs": []
    },
    {
      "access": "replicationTarget",
      "accountID": 1,
      "attributes": {
        "name1": "value1",
        "name2": "value2",
        "name3": "value3"
      },
      "blockSize": 4096,
      "createTime": "2016-04-06T17:26:31Z",
      "deleteTime": "",
      "enable512e": false,
      "iqn": "iqn.2010-01.com.solidfire:jo73.3",
      "name": "doctest2",
      "purgeTime": "",
      "qos": {
        "burstIOPS": 150,
        "burstTime": 60,
        "curve": {
          "4096": 100,
```

```
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 100,
          "minIOPS": 50
        },
        "scsiEUIDeviceID": "6a6f373300000003f47acc0100000000",
        "scsiNAADeviceID": "6f47acc1000000006a6f373300000003",
        "sliceCount": 1,
        "status": "active",
        "totalSize": 1000341504,
        "virtualVolumeID": null,
        "volumeAccessGroups": [],
        "volumeID": 3,
        "volumePairs": []
      }
    ]
  }
}
```


New since version

9.6

Find more information

GetDefaultQoS

PurgeDeletedVolume

You can use the `PurgeDeletedVolume` method to immediately and permanently purge a volume that has been deleted. You must delete a volume using `DeleteVolume` before it can be purged.

Volumes are purged automatically after a period of time, so usage of this method is not typically required.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	The volumeID of the volume to be purged.	integer	No	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "PurgeDeletedVolume",
   "params": {
      "volumeID"  : 5
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {}
}
```

New since version

9.6

Find more information

DeleteVolume

PurgeDeletedVolumes

You can use the `PurgeDeletedVolumes` method to immediately and permanently purge volumes that have been deleted; you can use this method to purge up to 500 volumes at one time.

You must delete volumes using `DeleteVolumes` before they can be purged. Volumes are purged automatically after a period of time, so usage of this method is not typically required.

If you purge a large number of volumes at one time, or if the volumes you purge each have many associated snapshots, the method might fail and return the error "xDBConnectionLoss". If this happens, retry the method call again with fewer volumes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeIDs	A list of volumeIDs of volumes to be purged from the system.	integer array	No	No
accountIDs	A list of accountIDs. All of the volumes from all of the specified accounts are purged from the system.	integer array	No	No
volumeAccessGroupIDs	A list of volumeAccessGroupIDs. All of the volumes from all of the specified volume access groups are purged from the system.	integer array	No	No

Note: You can specify only one of the above parameters per method call. Specifying more than one, or none, results in an error.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "PurgeDeletedVolumes",
   "params": {
      "accountIDs" : [1, 2, 3]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {}
}
```

New since version

9.6

Find more information

DeleteVolumes

RemoveBackupTarget

You can use the `RemoveBackupTarget` method to remove backup targets.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
backupTargetID	Unique target ID of the target to remove.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveBackupTarget",
   "params": {
        "backupTargetID" : 1
   },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {}
}
```

New since version

9.6

RestoreDeletedVolume

You can use the `RestoreDeletedVolume` method to mark a deleted volume as active again. This action makes the volume immediately available for iSCSI connection.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeID	The volumeID of the deleted volume to restore.	integer	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RestoreDeletedVolume",
   "params": {
      "volumeID"  : 5
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result": {}
}
```

New since version

9.6

SetDefaultQoS

You can use the `SetDefaultQoS` method to configure the default Quality of Service (QoS) values (measured in inputs and outputs per second, or IOPS) for a volume.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
minIOPS	The minimum number of sustained IOPS that are provided by the cluster to a volume.	integer	None	No
maxIOPS	The maximum number of sustained IOPS that are provided by the cluster to a volume.	integer	None	No
burstIOPS	The maximum number of IOPS allowed in a short burst scenario.	integer	None	No

Return values

This method has the following return values:

Name	Description	Type
minIOPS	The minimum number of sustained IOPS that are provided by the cluster to a volume.	integer

Name	Description	Type
maxIOPS	The maximum number of sustained IOPS that are provided by the cluster to a volume.	integer
burstIOPS	The maximum number of IOPS allowed in a short burst scenario.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "SetDefaultQoS",
    "params": {
        "burstIOPS":8000,
        "maxIOPS":1000,
        "minIOPS":200
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id":1,
    "result": {
        "burstIOPS":8000,
        "maxIOPS":1000,
        "minIOPS":200
    }
}
```

New since version

9.6

StartBulkVolumeRead

You can use the `StartBulkVolumeRead` method to start a bulk volume read session on a specified volume.

Only two bulk volume processes can run simultaneously on a volume. When you initialize the session, data is

read from a SolidFire storage volume to be stored on an external backup source. The external data is accessed by a web server running on an Element storage node. Server interaction information for external data access is passed by a script running on the storage system.

At the start of a bulk volume read operation, a snapshot of the volume is made and the snapshot is deleted when the read has completed. You can also read a snapshot of the volume by entering the ID of the snapshot as a parameter. When you read a previous snapshot, the system does not create a new snapshot of the volume, nor does it delete the previous snapshot when the read completes.

This process creates a new snapshot if the ID of an existing snapshot is not provided. Snapshots can be created if cluster fullness is at stage 2 or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
format	The format of the volume data. Can be either: • `uncompressed`: Every byte of the volume is returned without any compression. • `native`: Opaque data is returned that is smaller and more efficiently stored and written on a subsequent bulk volume write.	string	None	Yes
volumeID	The ID of the volume to be read.	integer	None	Yes
snapshotID	The ID of a previously created snapshot used for bulk volume reads. If no ID is entered, a snapshot of the current active volume image is made.	integer	None	No

Name	Description	Type	Default value	Required
script	The name of an executable script. If no script name is given, the key and URL are necessary to access Element storage nodes. The script is run on the primary node, and the key and URL are returned to the script so the local web server can be contacted.	string	None	No
scriptParameters	JSON parameters to pass to the script.	JSON object	None	No
attributes	List of name-value pairs in the JSON object format. Learn more.	JSON object	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	The ID of the asynchronous process to be checked for completion.	integer
key	Opaque key uniquely identifying the session.	string
url	URL to access the node's web server.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "StartBulkVolumeRead",
   "params": {
      "volumeID" : 5,
      "format"  : "native",
      "snapshotID" : 2
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
      "id" : 1,
   "result" : {
      "asyncHandle" : 1,
      "key" : "11eed8f086539205beeaadd981aad130",
      "url" : "https://127.0.0.1:44000/"
   }
}
```

New since version

9.6

StartBulkVolumeWrite

You can use the `StartBulkVolumeWrite` method to start a bulk volume write session on a specified volume.

Only two bulk volume processes can run simultaneously on a volume. When you initialize the session, data is written to an Element storage volume from an external backup source. The external data is accessed by a web server running on an Element storage node. Server interaction information for external data access is passed by a script running on the storage system.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
format	The format of the volume data. Can be either: • `uncompressed`: Every byte of the volume is returned without any compression. • `native`: Opaque data is returned that is smaller and more efficiently stored and written on a subsequent bulk volume write.	string	None	Yes
volumeID	The ID of the volume to be written to.	integer	None	Yes
script	The name of an executable script. If no script name is given, the key and URL are necessary to access Element storage nodes. The script is run on the primary node, and the key and URL are returned to the script so the local web server can be contacted.	string	None	No
scriptParameters	JSON parameters to pass to the script.	JSON object	None	No
attributes	List of name-value pairs in the JSON object format. Learn more.	JSON object	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	The ID of the asynchronous process to be checked for completion.	integer
key	Opaque key uniquely identifying the session.	string
url	URL to access the node's web server.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "StartBulkVolumeWrite",
   "params": {
      "volumeID" : 5,
      "format"  : "native",
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "asyncHandle" : 1,
      "key" : "11eed8f086539205beeaadd981aad130",
      "url" : "https://127.0.0.1:44000/"
   }
}
```

New since version

9.6

UpdateBulkVolumeStatus

You can use the `UpdateBulkVolumeStatus` method to update the status of a bulk volume job that you started with the `StartBulkVolumeRead` or `StartBulkVolumeWrite` methods.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
key	The key assigned during initialization of a StartBulkVolumeRead or StartBulkVolumeWrite session.	string	None	Yes
status	The system sets the status of the given bulk volume job. Possible values: • running: Jobs that are still active. • complete: Jobs that are done. • failed: Jobs that have failed.	string	None	Yes
percentComplete	The completed progress of the bulk volume job as a percentage.	string	None	No
message	Returns the status of the bulk volume job when the job has completed.	string	None	No
attributes	JSON attributes; updates what is on the bulk volume job.	JSON object	None	No

Return values

This method has the following return values:

Name	Description	Type
status	Status of the session requested. Returned status: • preparing • active • done • failed	string
attributes	Returns attributes that were specified in the method call. Values are returned whether they have changed or not.	string
url	The URL to access the node's web server; provided only if the session is still active.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "UpdateBulkVolumeStatus",
   "params": {
          "key": "0b2f532123225febda2625f55dcb0448",
          "status": "running"
      },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result": {
      "status" : "running",
      "url" : "https://10.10.23.47:8443/"
    }
}
```

New since version

9.6

Find more information

- StartBulkVolumeRead
- StartBulkVolumeWrite

Volume access group API methods

Volume access group methods enable you to add, remove, view, and modify volume access groups, which are collections of volumes that users can access using either iSCSI or Fibre Channel initiators.

- AddInitiatorsToVolumeAccessGroup
- AddVolumesToVolumeAccessGroup
- CreateVolumeAccessGroup
- DeleteVolumeAccessGroup
- ListVolumeAccessGroups
- RemoveVolumesFromVolumeAccessGroup
- RemoveInitiatorsFromVolumeAccessGroup
- ModifyVolumeAccessGroup
- GetVolumeAccessGroupEfficiency

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

AddInitiatorsToVolumeAccessGroup

You can use the `AddInitiatorsToVolumeAccessGroup` method to add initiators to a specified volume access group.

The accepted format of an initiator IQN is iqn.yyyy-mm, where y and m are digits, followed by text which must only contain digits, lower-case alphabetic characters, a period (.), colon (:) or dash (-). See the following example:

```
iqn.2010-01.com.solidfire:17oi.solidfire-0.1
```

The accepted format of a Fibre Channel initiator WWPN is Aa:bB:CC:dd:11:22:33:44, or AabBCCdd11223344. See the following example:

```
21:00:00:0e:1e:11:f1:81
```

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
initiators	List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the method returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.	integer array or string array (deprecated)		Yes
volumeAccessGroupID	The ID of the volume access group to add the initiator.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumeAccessGroup	An object containing information about the newly modified volume access group.	volumeAccessGroup

Request example

Requests for this method are similar to the following example:

```
{
  "id": 13171,
  "method": "AddInitiatorsToVolumeAccessGroup",
  "params": {
    "initiators": [116,117],
    "volumeAccessGroupID": 96
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 13171,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        327
      ],
      "initiatorIDs": [
        116,
        117
      ],
      "initiators": [
        "iqn.1993-08.org.debian:01:181324777",
        "iqn.1993-08.org.debian:01:181324888"
      ],
      "name": "northbanktest",
      "volumeAccessGroupID": 96,
      "volumes": [
        346
      ]
    }
  }
}
```

New since version

9.6

AddVolumesToVolumeAccessGroup

You can use the `AddVolumesToVolumeAccessGroup` method to add volumes to a

specified volume access group.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumes	List of volumeIDs to add to the volume access group.	integer array	None	Yes
volumeAccessGroupID	VolumeAccessGroupID of the volume access group to which volumes are added.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumeAccessGroup	An object containing information about the newly modified volume access group.	volumeAccessGroup

Request example

Requests for this method are similar to the following example:

```
{
   "method": "AddVolumesToVolumeAccessGroup",
   "params": {
      "volumeAccessGroupID": 96,
      "volumes": [1,2]
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        346
      ],
      "initiatorIDs": [
        116,
        117
      ],
      "initiators": [
        "iqn.1993-08.org.debian:01:181324777",
        "iqn.1993-08.org.debian:01:181324888"
      ],
      "name": "northbanktest",
      "volumeAccessGroupID": 96,
      "volumes": [
        1,
        2
      ]
    }
  }
}
```


New since version

9.6

CreateVolumeAccessGroup

You can use `CreateVolumeAccessGroup` to create a new volume access group. When you create the volume access group, you need to give it a name, and you can optionally enter initiators and volumes.

Any initiator IQN that you add to the volume access group is able to access any volume in the group without CHAP authentication.

Cloned volumes do not inherit volume access group membership from the source volume.

Consider the following when you create volume access groups:

- A volume access group can contain up to 64 initiator IQNs.
- An initiator can only belong to one volume access group.
- A volume access group can contain up to 2000 volumes.

- Each volume access group can belong to a maximum of four volume access groups.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
initiators	List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the method returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.	integer array or string array (deprecated)		No
name	Name of the volume access group. Not required to be unique, but recommended. Must be 1 to 64 characters in length.	string	None	Yes
volumes	List of volumeIDs to include in the volume access group.	integer array		No
attributes	List of name-value pairs in JSON object format.	JSON object	{}	No

Return values

This method has the following return values:

Name	Description	Type

volumeAccessGroup	An object containing information about the newly created volume access group.	volumeAccessGroup
volumeAccessGroupID	The ID of the newly created volume access group.	integer

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CreateVolumeAccessGroup",
   "params": {
      "name": "myaccessgroup",
      "initiators": ["iqn.1993-08.org.debian: 01: a31b1d799d5c"],
      "volumes": [327],
      "attributes": {}
   }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": null,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [],
      "initiatorIDs": [
        95
      ],
      "initiators": [
        "iqn.1993-08.org.debian: 01: a31b1d799d5c"
      ],
      "name": "myaccessgroup",
      "volumeAccessGroupID": 96,
      "volumes": [
        327
      ]
    },
    "volumeAccessGroupID": 96
  }
}
```


New since version

9.6

Find more information

- GetAsyncResult
- ListSyncJobs
- ModifyVolume

DeleteVolumeAccessGroup

You can use `DeleteVolumeAccessGroup` to delete a volume access group.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeAccessGroupID	The ID of the volume access group to be deleted.	integer	None	Yes

Name	Description	Type	Default value	Required
deleteOrphanInitiators	Specifies whether to delete initiator objects or not. Possible values: • true: Delete initiator objects after they are removed from a volume access group. • false: Do not delete initiator objects after they are removed from a volume access group. This is the default.	boolean	false	No
force	Adding this flag will force the volume access group to be deleted even though it has a Virtual Network ID or Tag. Possible values: • true: Volume access group will be deleted. • false: Default. Do not delete the volume access group if it has a Virtual Network ID or Tag.	boolean	false	No

Return values

This method does not have return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteVolumeAccessGroup",
   "params": {
          "force": true,
      "volumeAccessGroupID"  : 3
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result": {}
}
```

New since version

9.6

ListVolumeAccessGroups

You can use the `ListVolumeAccessGroups` method to get information about the volume access groups that are currently in the system.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
limit	Maximum number of volumeAccessGroup objects to return. Mutually exclusive with the volumeAccessGroups parameter.	integer	Unlimited	No

Name	Description	Type	Default value	Required
startVolumeAccessGroupID	The volume access group ID at which to begin the listing. Mutually exclusive with the volumeAccessGroups parameter.	integer	0	No
volumeAccessGroups	List of volumeAccessGroupID values to retrieve. Mutually exclusive with the startVolumeAccessGroupID and limit parameters.	integer array		No

Return values

This method has the following return values:

Name	Description	Type
volumeAccessGroups	A list of objects describing each volume access group.	volumeAccessGroup array
volumeAccessGroupsNotFound	A list of volume access groups not found by the system. Present if you used the volumeAccessGroups parameter and the system was unable to find one or more volume access groups that you specified.	integer array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListVolumeAccessGroups",
   "params": {
      "startVolumeAccessGroupID": 3,
      "limit"   : 1
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeAccessGroups": [
      {
        "attributes": {},
        "deletedVolumes": [],
        "initiatorIDs": [],
        "initiators": [],
        "name": "example1",
        "volumeAccessGroupID": 3,
        "volumes": []
      }
    ]
  }
}
```

New since version

9.6

RemoveVolumesFromVolumeAccessGroup

You can use the `RemoveVolumesFromVolumeAccessGroup` method to remove volumes from a specified volume access group.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeAccessGroupID	VolumeAccessGroupID to remove volumes from.	integer	None	Yes
volumes	VolumeIDs of volumes to remove from the volume access group.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
volumeAccessGroup	An object containing information about the newly modified volume access group.	volumeAccessGroup

Request example

Requests for this method are similar to the following example:

```
{
   "method": "RemoveVolumesFromVolumeAccessGroup",
   "params": {
      "volumeAccessGroupID": 96,
      "volumes": [1,2]
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        346
      ],
      "initiatorIDs": [
        116,
        117
      ],
      "initiators": [
        "iqn.1993-08.org.debian:01:181324777",
        "iqn.1993-08.org.debian:01:181324888"
      ],
      "name": "northbanktest",
      "volumeAccessGroupID": 96,
      "volumes": []
    }
  }
}
```

New since version

9.6

RemoveInitiatorsFromVolumeAccessGroup

You can use the `RemoveInitiatorsFromVolumeAccessGroup` method to remove initiators from a specified volume access group.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeAccessGroupID	The ID of the volume access group from which initiators are removed.	integer	None	Yes
initiators	List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the method returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.	integer array (recommended) or string array (deprecated)	None	No

Name	Description	Type	Default value	Required
deleteOrphanInitiators	Specifies whether to delete initiator objects after they are removed from a volume access group or not. Possible values: • true: Delete initiator objects after they are removed from a volume access group. • false: Do not delete initiator objects after they are removed from a volume access group. This is the default.	boolean	false	No

Return value

This method has the following return value:

Name	Description	Type
volumeAccessGroup	An object containing information about the newly modified volume access group.	volumeAccessGroup

Request example

Requests for this method are similar to the following example:

```
{
  "id": 13171,
  "method": "RemoveInitiatorsFromVolumeAccessGroup",
  "params": {
    "initiators": [114,115],
    "volumeAccessGroupID": 96
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 13171,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        327
      ],
      "initiatorIDs": [],
      "initiators": [],
      "name": "test",
      "volumeAccessGroupID": 96,
      "volumes": [
        346
      ]
    }
  }
}
```

New since version

9.6

ModifyVolumeAccessGroup

You can use the `ModifyVolumeAccessGroup` method to update initiators and add or remove volumes from a volume access group.

If a specified initiator or volume is a duplicate of what currently exists, the volume access group is left as-is. If you do not specify a value for volumes or initiators, the current list of initiators and volumes is not changed.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeAccessGroupID	The ID of the volume access group to modify.	integer	None	Yes
name	The new name for this volume access group.	string	None	No

attributes	List of name-value pairs in JSON object format.	JSON object	None	No
initiators	List of initiator IDs or names (IQNs and WWPNs) to include in the volume access group. If you pass a list of initiator names, the initiators are created if they do not already exist. If you pass a list of initiator IDs, the method returns an error if any of the initiators does not already exist. Passing initiator names is deprecated; you should use initiator IDs whenever possible.	integer array (recommended) or string array (deprecated)	None	No
deleteOrphanInitiators	Specifies whether to delete initiator objects after they are removed from a volume access group or not. Possible values: • true: Delete initiator objects after they are removed from a volume access group. • false: Do not delete initiator objects after they are removed from a volume access group. This is the default.	boolean	false	No

volumes	A list of volume IDs of volumes to modify.	integer array	None	volumeAccessGroup

Return value

This method has the following return value:

Name	Description	Type
volumeAccessGroup	An object containing information about the newly modified volume access group.	volumeAccessGroup

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ModifyVolumeAccessGroup",
   "params": {
      "volumeAccessGroupID": 96,
      "name": "accessgrouptest",
      "initiators": [115,114],
      "volumes": [
         346
     ],
      "attributes": {}
   }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": null,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        327
      ],
      "initiatorIDs": [
        114,
        115
      ],
      "initiators": [
        "iqn.1998-01.com.vmware:desk1-esx1-577b283a",
        "iqn.1998-01.com.vmware:donesq-esx1-421b281b"
      ],
      "name": "accessgrouptest",
      "volumeAccessGroupID": 96,
      "volumes": [
        346
      ]
    }
  }
}
```


New since version

9.6

Find more information

- AddInitiatorsToVolumeAccessGroup
- AddVolumesToVolumeAccessGroup
- RemoveInitiatorsFromVolumeAccessGroup
- RemoveVolumesFromVolumeAccessGroup

GetVolumeAccessGroupEfficiency

You can use the `GetVolumeAccessGroupEfficiency` method to get efficiency information about a volume access group. Only the volume access group you provide as the parameter in this API method is used to compute the capacity.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
volumeAccessGroupID	Specifies the volume access group for which capacity is computed.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
compression	The amount of space saved by data compression for all volumes in the volume access group. Stated as a ratio where a value of 1 means data has been stored with no compression.	float
deduplication	The amount of space saved by not duplicating data for all volumes in the volume access group. Stated as a ratio.	float
thinProvisioning	The ratio of space used to the amount of space allocated for storing data. Stated as a ratio.	float
timestamp	The last time efficiency data was collected after garbage collection.	ISO 8601 data string
missingVolumes	The volumes that could not be queried for efficiency data. Missing volumes can be caused by a recent garbage collection, temporary network loss or restarted services since the garbage collection cycle.	integer array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetVolumeAccessGroupEfficiency",
   "params": {
      "volumeAccessGroupID": 1
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id": 1,
   "result": {
       "compression": 2.006012925331075,
       "deduplication": 1,
       "missingVolumes": [],
       "thinProvisioning": 1.009861932938856,
       "timestamp": "2014-03-10T17:05:27Z"
   }
}
```

New since version

9.6

Volume snapshot API methods

Element software volume snapshot API methods enable you to manage volume snapshots. You can create, modify, clone, and delete volume snapshots using the volume snapshot API methods.

- Snapshots overview
- CreateGroupSnapshot
- CreateSchedule
- CreateSnapshot
- DeleteGroupSnapshot
- DeleteSnapshot
- GetSchedule
- ListGroupSnapshots
- ListSchedules

- ListSnapshots
- ModifyGroupSnapshot
- ModifySchedule
- ModifySnapshot
- RollbackToGroupSnapshot
- RollbackToSnapshot

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

Snapshots overview

A volume snapshot is a point-in-time copy of a volume. You can use snapshots to roll a volume back to the state it was in at the time the snapshot was created.

You can group volume snapshots together so that related volumes can be backed up or rolled back in a consistent manner. A group snapshot captures a point-in-time image of all volume slice files. You can then use the image to roll back a group of volumes to a point-in-time state and ensure that all data is consistent across all volumes in the group.

You can schedule volume snapshots to occur autonomously at defined intervals. You can define intervals by time, days of the week, or days of the month. You can also use scheduled snapshots to ensure snapshots are backed up to remote storage for archiving purposes.

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

CreateGroupSnapshot

You can use `CreateGroupSnapshot` to create a point-in-time copy of a group of volumes.

You can use this snapshot later as a backup or rollback to ensure the data on the group of volumes is consistent for the point in time that you created the snapshot.

CLUSTER_FULLNESS

> You can create snapshots if cluster fullness is at stage 1, 2, or 3. You cannot create snapshots when cluster fullness reaches stage 4 or 5.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`attributes`	List of name-value pairs in JSON object format.	JSON object	None	No
`enableRemoteReplication`	Specifies whether the snapshot will be replicated to remote storage or not. Possible values: • `true`: The snapshot will be replicated to remote storage. • `false`: The snapshot will not be replicated to remote storage.	boolean	false	No
`ensureSerialCreation`	Specifies that the snapshot should not be created if a previous snapshot replication is in progress. Possible values are: • `true`: This ensures that only one snapshot is being replicated at a time. The creation of a new snapshot will fail if a previous snapshot replication is still in progress. • `false`: Default. This snapshot creation is allowed if another snapshot replication is still in progress.	boolean	`false`	No

Name	Description	Type	Default value	Required
`expirationTime`	Specify the time after which the snapshot can be removed. Cannot be used with `retention`. If neither `expirationTime`, or `retention` are specified, the snapshot will not expire. The time format is an ISO 8601 date string for time based expiration, otherwise it will not expire. A value of `null` causes the snapshot to be retained permanently. A value of `fifo` causes the snapshot to be preserved on a First-In-First-Out (FIFO) basis, relative to other FIFO snapshots on the volume. The API will fail if no FIFO space is available.	ISO 8601 date string	None	No
`name`	The name of the group snapshot. If no name is entered, the date and time the group snapshot was taken is used. The maximum name length allowed is 255 characters.	string	None	No

Name	Description	Type	Default value	Required
`retention`	This parameter is same as the `expirationTime` parameter, except the time format is HH:mm:ss. If neither `expirationTime` nor `retention` are specified, the snapshot will not expire.	string	None	No
`snapMirrorLabel`	The label used by SnapMirror software to specify the snapshot retention policy on a SnapMirror endpoint.	string	None	No
`volumes`	Unique ID of the volume image from which to copy.	volumeID array	None	Yes

Return values

This method has the following return values:

Name	Description	Type
members	List of checksum, volumeIDs, and snapshotIDs for each member of the group. Valid values: • checksum: A small string representation of the data in the stored snapshot. This checksum can be used later to compare other snapshots to detect errors in the data. (string) • snapshotID: Unique ID of a snapshot from which the new snapshot is made. The snapshotID must be from a snapshot on the given volume. (integer) • volumeID: The source volume ID for the snapshot. (integer)	JSON object array

groupSnapshotID	Unique ID of the new group snapshot.	groupSnapshot ID
groupSnapshot	Object containing information about the newly created group snapshot.	groupSnapshot

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CreateGroupSnapshot",
   "params": {
      "volumes": [1,2]
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "groupSnapshot": {
      "attributes": {},
      "createTime": "2016-04-04T22:43:29Z",
      "groupSnapshotID": 45,
      "groupSnapshotUUID": "473b78a3-ef85-4541-9438-077306b2d3ca",
      "members": [
        {
          "attributes": {},
          "checksum": "0x0",
          "createTime": "2016-04-04T22:43:29Z",
          "enableRemoteReplication": false,
          "expirationReason": "None",
          "expirationTime": null,
          "groupID": 45,
          "groupSnapshotUUID": "473b78a3-ef85-4541-9438-077306b2d3ca",
          "name": "2016-04-04T22:43:29Z",
          "snapshotID": 3323,
          "snapshotUUID": "7599f200-0092-4b41-b362-c431551937d1",
          "status": "done",
          "totalSize": 5000658944,
```

```
          "virtualVolumeID": null,
          "volumeID": 1
        },
        {
          "attributes": {},
          "checksum": "0x0",
          "createTime": "2016-04-04T22:43:29Z",
          "enableRemoteReplication": false,
          "expirationReason": "None",
          "expirationTime": null,
          "groupID": 45,
          "groupSnapshotUUID": "473b78a3-ef85-4541-9438-077306b2d3ca",
          "name": "2016-04-04T22:43:29Z",
          "snapshotID": 3324,
          "snapshotUUID": "a0776a48-4142-451f-84a6-5315dc37911b",
          "status": "done",
          "totalSize": 6001000448,
          "virtualVolumeID": null,
          "volumeID": 2
        }
      ],
      "name": "2016-04-04T22:43:29Z",
      "status": "done"
    },
    "groupSnapshotID": 45,
    "members": [
      {
        "checksum": "0x0",
        "snapshotID": 3323,
        "snapshotUUID": "7599f200-0092-4b41-b362-c431551937d1",
        "volumeID": 1
      },
      {
        "checksum": "0x0",
        "snapshotID": 3324,
        "snapshotUUID": "a0776a48-4142-451f-84a6-5315dc37911b",
        "volumeID": 2
      }
    ]
  }
}
```


New since version

9.6

CreateSchedule

You can use `CreateSchedule` to schedule an automatic snapshot of a volume at a defined interval.

You can use the created snapshot later as a backup or rollback to ensure the data on a volume or group of volumes is consistent for the point in time in which the snapshot was created. If you schedule a snapshot to run at a time period that is not divisible by 5 minutes, the snapshot will run at the next time period that is divisible by 5 minutes. For example, if you schedule a snapshot to run at 12:42:00 UTC, it will run at 12:45:00 UTC. You cannot schedule a snapshot to run at intervals of less than 5 minutes.

You can create snapshots if cluster fullness is at stage 1, 2, or 3. You cannot create snapshots when cluster fullness reaches stage 4 or 5.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`attributes`	Use the “frequency” string to indicate the frequency of the snapshot. Possible values: • `Days of Week` • `Days of Month` • `Time Interval`	JSON object	None	No
`hasError`	**Help with description needed**	boolean	`false`	No
`hours`	Number of hours between recurring snapshots or hour in GMT time that the snapshot will occur in Days of Week or Days of Month mode. Valid values are 0 through 23.	integer	None	No
`lastRunStatus`	The result or status of the last scheduled snapshot creation.	string	None	No

Name	Description	Type	Default value	Required
`name`	The name of the snapshot. If no name is entered, the date and time the group snapshot was taken is used. The maximum name length allowed is 244 characters.	string	None	No
`minutes`	Number of minutes between recurring snapshots or the minute in GMT time that the snapshot will occur in Days of Week or Days of Month mode. Valid values are 5 through 59.	integer	None	No
`paused`	Indicates if the schedule should be paused or not. Valid values: • `true` • `false`	boolean	None	No
`recurring`	Indicates if the schedule will be recurring or not. Valid values are: • `true` • `false`	boolean	None	No

Name	Description	Type	Default value	Required
`runNextInterval`	Specifies whether or not to run the snapshot the next time the scheduler is active. When set to true, the scheduled snapshot runs the next time the scheduler is active and resets back to false. Valid values are: • `true` • `false`	boolean	`false`	No
`scheduleName`	Unique name for the schedule. The maximum schedule name length allowed is 244 characters.	string	None	Yes
`scheduleType`	Indicates the type of schedule to create. Valid value is snapshot.	string	None	Yes

Name	Description	Type	Default value	Required
`scheduleInfo`	The unique name given to the schedule, the retention period for the snapshot that was created, and the volume ID of the volume from which the snapshot was created. Valid values: • `volumeID`: The ID of the volume to be included in the snapshot. (integer) • `volumes`: A list of volume IDs to be included in the group snapshot. (integer array) • `name`: The snapshot name to be used. (string) • `enableRemoteReplication`: Indicates if the snapshot should be included in remote replication. (boolean) • `retention`: The amount of time the snapshot will be retained in HH:mm:ss. If empty, the snapshot is retained forever. (string) • `fifo`: The snapshot is retained on a First-In-First-Out (FIFO) basis. (string) • `ensureSerialCreation`:	JSON object	None	Yes

Name	Description	Type	Default value	Required
`snapMirrorLabel`	The label used by SnapMirror software to specify the snapshot retention policy on a SnapMirror endpoint.	string	None	No
`startingDate`	Time after which the schedule will be run. If not set, the schedule starts immediately. Formatted in UTC time.	ISO 8601 date string	None	No
`toBeDeleted`	Specifies that this snapshot schedule should be deleted after snapshot creation is completed.	boolean	`false`	No
`monthdays`	The days of the month that a snapshot will be made. Valid values are 1 through 31.	integer array	None	Yes (if scheduling for days of the month)

Name	Description	Type	Default value	Required
`weekdays`	Day of the week the snapshot is to be created. Required values (if used): • `Day`: 0 through 6 (Sunday through Saturday) • `Offset`: for each possible week in a month, 1 through 6 (If greater than 1, only matched on the Nth-1 day of the week. For example, offset:3 for Sunday means the third Sunday of the month, while offset:4 for Wednesday means the fourth Wednesday of the month. Offset:0 means no action is taken. Offset:1 (default) means that the snapshot is created for this day of the week, regardless of where it falls in the month)	JSON object array	None	Yes (if scheduling for days of the week)

Return values

This method has the following return values:

Name	Description	Type
scheduleID	ID of the schedule created.	integer
schedule	An object containing information about the newly created schedule.	schedule

Request example 1

The following example schedule has the following parameters:

- No start hours or minutes are specified so the schedule starts as closely as possible to midnight (00:00:00Z).
- It is not recurring (will only run once).
- It runs once on either the first Sunday or Wednesday following June 1, 2015, UTC 19:17:15Z (whichever day comes first).
- It includes only one volume (volumeID = 1).

```
{
  "method":"CreateSchedule",
  "params":{
    "hours":0,
    "minutes":0,
    "paused":false,
    "recurring":false,
    "scheduleName":"MCAsnapshot1",
    "scheduleType":"snapshot",
    "attributes":{
      "frequency":"Days Of Week"
    },
    "scheduleInfo":{
      "volumeID":"1",
      "name":"MCA1"
    },
    "monthdays":[],
    "weekdays":[
      {
        "day":0,
        "offset":1
      },
      {
        "day":3,
        "offset":1
      }
    ],
    "startingDate":"2015-06-01T19:17:54Z"
  },
   "id":1
}
}
}
```

Response example 1

The above request returns a response similar to the following example:


```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Days Of Week"
      },
      "hasError": false,
      "hours": 0,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": null,
      "minutes": 0,
      "monthdays": [],
      "paused": false,
      "recurring": false,
      "runNextInterval": false,
      "scheduleID": 4,
      "scheduleInfo": {
        "name": "MCA1",
        "volumeID": "1"
      },
      "scheduleName": "MCAsnapshot1",
      "scheduleType": "Snapshot",
      "startingDate": "2015-06-01T19:17:54Z",
      "toBeDeleted": false,
      "weekdays": [
        {
          "day": 0,
          "offset": 1
        },
        {
          "day": 3,
          "offset": 1
        }
      ]
    },
    "scheduleID": 4
  }
}
```

Request example 2

The following example schedule has the following parameters:

- It is recurring (will run at each scheduled interval of the month at the specified time).
- It runs on the 1st, 10th, 15th and 30th of each month following the starting date.
- It runs at 12:15 PM on each day it is scheduled to occur.
- It includes only one volume (volumeID = 1).

```
{
  "method":"CreateSchedule",
    "params":{
      "hours":12,
      "minutes":15,
      "paused":false,
      "recurring":true,
      "scheduleName":"MCASnapshot1",
      "scheduleType":"snapshot",
      "attributes":{
        "frequency":"Days Of Month"
      },
      "scheduleInfo":{
        "volumeID":"1"
      },
      "weekdays":[
      ],
      "monthdays":[
        1,
        10,
        15,
        30
      ],
      "startingDate":"2015-04-02T18:03:15Z"
    },
    "id":1
}
```

Response example 2

The above request returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Days Of Month"
      },
      "hasError": false,
      "hours": 12,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": null,
      "minutes": 15,
      "monthdays": [
        1,
        10,
        15,
        30
      ],
      "paused": false,
      "recurring": true,
      "runNextInterval": false,
      "scheduleID": 5,
      "scheduleInfo": {
        "volumeID": "1"
      },
      "scheduleName": "MCASnapshot1",
      "scheduleType": "Snapshot",
      "startingDate": "2015-04-02T18:03:15Z",
      "toBeDeleted": false,
      "weekdays": []
    },
      "scheduleID": 5
  }
}
```


Request example 3

The following example schedule has the following parameters:

- It starts within 5 minutes of the scheduled interval on April 2, 2015.
- It is recurring (will run at each scheduled interval of the month at the specified time).
- It runs on the second, third, and fourth of each month following the starting date.
- It runs at 14:45 PM on each day it is scheduled to occur.
- It includes a group of volumes (volumes = 1 and 2).

```
{
  "method":"CreateSchedule",
  "params":{
    "hours":14,
    "minutes":45,
    "paused":false,
    "recurring":true,
    "scheduleName":"MCASnapUser1",
    "scheduleType":"snapshot",
    "attributes":{
      "frequency":"Days Of Month"
    },
    "scheduleInfo":{
      "volumes":[1,2]
    },
    "weekdays":[],
    "monthdays":[2,3,4],
    "startingDate":"2015-04-02T20:38:23Z"
  },
  "id":1
}
```

Response example 3

The above request returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Days Of Month"
      },
      "hasError": false,
      "hours": 14,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": null,
      "minutes": 45,
      "monthdays": [
        2,
        3,
        4
      ],
      "paused": false,
      "recurring": true,
      "runNextInterval": false,
      "scheduleID": 6,
      "scheduleInfo": {
        "volumes": [
          1,
          2
        ]
      },
      "scheduleName": "MCASnapUser1",
      "scheduleType": "Snapshot",
      "startingDate": "2015-04-02T20:38:23Z",
      "toBeDeleted": false,
      "weekdays": []
    },
    "scheduleID": 6
  }
}
```


New since version

9.6

CreateSnapshot

You can use `CreateSnapshot` to create a point-in-time copy of a volume. You can create a snapshot from any volume or from an existing snapshot.

If you do not provide a SnapshotID with this API method, a snapshot is created from the volume's active branch. If the volume from which the snapshot is created is being replicated to a remote cluster, the snapshot can also be replicated to the same target. Use the enableRemoteReplication parameter to enable snapshot replication.

> You can create snapshots if cluster fullness is at stage 1, 2, or 3. You cannot create snapshots when cluster fullness reaches stage 4 or 5.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`attributes`	List of name-value pairs in JSON object format.	JSON object	None	No
`enableRemoteReplication`	Specifies whether the snapshot will be replicated to remote storage or not. Possible values: • `true`: The snapshot will be replicated to remote storage. • `false`: The snapshot will not be replicated to remote storage.	boolean	false	No

Name	Description	Type	Default value	Required
`ensureSerialCreation`	Specifies that the snapshot should not be created if a previous snapshot replication is in progress. Possible values are: • `true`: This ensures that only one snapshot is being replicated at a time. The creation of a new snapshot will fail if a previous snapshot replication is still in progress. • `false`: Default. This snapshot creation is allowed if another snapshot replication is still in progress.	boolean	`false`	No

Name	Description	Type	Default value	Required
expirationTime	Specify the time after which the snapshot can be removed. Cannot be used with `retention`. If neither expirationTime or retention are specified the snapshot will not expire. The time format is an ISO 8601 date string for time based expiration, otherwise it will not expire. A value of `null` causes the snapshot to be retained permanently. A value of `fifo` causes the snapshot to be preserved on a First-In-First-Out basis, relative to other FIFO snapshots on the volume. The API will fail if no FIFO space is available.	string	None	No
`name`	The name of the snapshot. If no name is entered, the date and time the snapshot was taken is used. The maximum name length allowed is 255 characters.	string	None	No

Name	Description	Type	Default value	Required
`retention`	This parameter is same as the `expirationTime` parameter, except the time format is HH:mm:ss. If neither `expirationTime` nor `retention` are specified, the snapshot will not expire.	string	None	No
`snapMirrorLabel`	The label used by SnapMirror software to specify the snapshot retention policy on a SnapMirror endpoint.	string	None	No
`snapshotID`	Unique ID of a snapshot from which the new snapshot is made. The snapshotID passed must be a snapshot on the given volume.	integer	None	No
`volumeID`	Unique ID of the volume image from which to copy.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
checksum	A string that represents the correct digits in the stored snapshot. This checksum can be used later to compare other snapshots to detect errors in the data.	string
snapshotID	Unique ID of the new snapshot.	Snapshot ID
snapshot	An object containing information about the newly created snapshot.	snapshot

Request example

Requests for this method are similar to the following example:

```
{
   "method": "CreateSnapshot",
   "params": {
      "volumeID": 1
   },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "checksum": "0x0",
      "snapshot": {
        "attributes": {},
        "checksum": "0x0",
        "createTime": "2016-04-04T17:14:03Z",
        "enableRemoteReplication": false,
        "expirationReason": "None",
        "expirationTime": null,
        "groupID": 0,
        "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",
        "name": "2016-04-04T17:14:03Z",
        "snapshotID": 3110,
        "snapshotUUID": "6f773939-c239-44ca-9415-1567eae79646",
        "status": "done",
        "totalSize": 5000658944,
        "virtualVolumeID": null,
        "volumeID": 1
      },
        "snapshotID": 3110
  }
}
```

Exception

An xNotPrimary exception is displayed when the `CreateSnapshot` API is called and the snapshot fails to get created. This is expected behavior. Retry the `CreateSnapshot` API call.

New since version

9.6

DeleteGroupSnapshot

You can use `DeleteGroupSnapshot` to delete a group snapshot.

You can use the saveMembers parameter to preserve all the snapshots that were made for the volumes in the group, but the group association will be removed.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
groupSnapshotID	Unique ID of the group snapshot.	integer	None	Yes
saveMembers	Specifies what to delete when you delete a group snapshot. Valid values: • true: Snapshots are kept, but the group association is removed. • false: The group and snapshots are deleted.	boolean	false	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteGroupSnapshot",
     "params": {
       "groupSnapshotID": 10,
       "saveMembers" : true
        },
     "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {}
}
```

New since version

9.6

DeleteSnapshot

You can use the `DeleteSnapshot` method to delete a snapshot.

A snapshot that is currently the active snapshot cannot be deleted. You must rollback and make another snapshot active before the current snapshot can be deleted.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
snapshotID	The ID of the snapshot to delete.	integer	None	Yes

Name	Description	Type	Default value	Required
overrideSnapMirrorHold	Override the lock placed on snapshots during replication. You can use this parameter to delete stale SnapMirror snapshots after the associated SnapMirror relationship has been deleted.	boolean	false	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "DeleteSnapshot",
"params": {
"snapshotID": 8,
"overrideSnapMirrorHold": true
},
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {}
}
```

New since version

9.6

Find more information

RollbackToSnapshot

GetSchedule

You can use `GetSchedule` to get information about a scheduled snapshot.

You can see information about a specific schedule if there are many snapshot schedules in the system. You also retrieve information about more than one schedule with this method by specifying additional IDs in the scheduleID parameter.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
scheduleID	Unique ID of the schedule or multiple schedules to display.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
schedule	An array of schedule attributes.	schedule array

Request example

Requests for this method are similar to the following example:

```
{
"method": "GetSchedule",
  "params": {
              "scheduleID" : 2
      },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Time Interval"
      },
       "hasError": false,
       "hours": 0,
       "lastRunStatus": "Success",
       "lastRunTimeStarted": "2015-03-23T21:25:00Z",
       "minutes": 2,
       "monthdays": [],
       "paused": false,
       "recurring": true,
       "runNextInterval": false,
       "scheduleID": 2,
       "scheduleInfo": {
          "name": "MCA2",
          "volumeID": "3"
       },
       "scheduleName": "MCAsnapshot2",
       "scheduleType": "Snapshot",
       "startingDate": "2015-03-23T19:28:57Z",
       "toBeDeleted": false,
       "weekdays": []
      }
   }
}
```


New since version

9.6

ListGroupSnapshots

You can use `ListGroupSnapshots` method to return information about all group snapshots that have been created.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
groupSnapshotID	Retrieve information for an individual group snapshot ID.	integer	None	No
volumes	An array of unique volume IDs to query. If you do not specify this parameter, all group snapshots on the cluster are included.	volumeID array	None	No

Return value

This method has the following return value:

Name	Description	Type
groupSnapshots	A list of objects containing information about each group snapshot.	groupSnapshot array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListGroupSnapshots",
  "params": {
  "volumes": [
     31,
     49
    ]
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "groupSnapshots": [
      {
          "status": "Done",
```

```
          "remoteStatuses": [
              {
                "volumePairUUID": "abcdef-1234-5678-90ab-cdef0123",
                "remoteStatus": "Present"
              }
          ],
          "attributes": {},
          "groupSnapshotID": 1,
          "createTime": "2014-06-17T17:35:05Z",
          "members": [
              {
                "snapshotUUID": "abcdef-1234-5678-90ab-cdef0123",
                "expirationReason": "None",
                "virtualVolumeID": "abcdef-1234-5678-90ab-cdef0123",
                "groupID": 1,
                "createTime": "2014-06-17T17:35:05Z",
                "totalSize": 1,
                "snapMirrorLabel": "test1",
                "volumeName": "test1",
                "instanceCreateTime": "2014-06-17T17:35:05Z",
                "volumeID": 1,
                "checksum": "0x0",
                "attributes": {},
                "instanceSnapshotUUID": "abcdef-1234-5678-90ab-cdef0123",
                "snapshotID": 1,
                "status": "Done",
                "groupSnapshotUUID": "abcdef-1234-5678-90ab-cdef0123",
                "expirationTime": "2014-06-17T17:35:05Z",
                "enableRemoteReplication": true,
                "name": "test1",
                "remoteStatuses": [
                      {
                        "volumePairUUID": "abcdef-1234-5678-90ab-
cdef0123",
                        "remoteStatus": "Present"
                      }
                  ]
              }
          ],
          "enableRemoteReplication": true,
          "name": "test1",
          "groupSnapshotUUID": "abcdef-1234-5678-90ab-cdef0123"
      }
    ]
}
```

New since version

9.6

ListSchedules

You can use `ListSchedules` to get information about all scheduled snapshots that have been created.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
schedules	A list of the schedules currently on the cluster.	schedule array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListSchedules",
    "params": {},
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "schedules": [
      {
       "attributes": {
         "frequency": "Days Of Week"
       },
       "hasError": false,
       "hours": 0,
       "lastRunStatus": "Success",
```

```
      "lastRunTimeStarted": null,
      "minutes": 1,
      "monthdays": [],
      "paused": false,
      "recurring": false,
      "runNextInterval": false,
      "scheduleID": 3,
      "scheduleInfo": {
           "name": "Wednesday Schedule",
           "retention": "00:02:00",
           "volumeID": "2"
    },
    "scheduleName": "Vol2Schedule",
    "scheduleType": "Snapshot",
    "startingDate": "2015-03-23T20:08:33Z",
    "toBeDeleted": false,
    "weekdays": [
        {
          "day": 3,
          "offset": 1
        }
   ]
  },
  {
    "attributes": {
        "frequency": "Time Interval"
    },
     "hasError": false,
     "hours": 0,
     "lastRunStatus": "Success",
     "lastRunTimeStarted": "2015-03-23T21:40:00Z",
     "minutes": 2,
     "monthdays": [],
     "paused": false,
     "recurring": true,
     "runNextInterval": false,
     "scheduleID": 2,
     "scheduleInfo": {
          "name": "MCA2",
          "volumeID": "3"
     },
     "scheduleName": "MCAsnapshot2",
     "scheduleType": "Snapshot",
     "startingDate": "2015-03-23T19:28:57Z",
     "toBeDeleted": false,
     "weekdays": []
```

```
          }
        ]
    }
}
```

New since version

9.6

ListSnapshots

You can use `ListSnapshots` to return the attributes of each snapshot taken on the volume.

Information about snapshots that reside on the target cluster will be displayed on the source cluster when this method is called from the source cluster.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeID	Retrieves snapshots for a volume. If volumeID is not provided, all snapshots for all volumes are returned.	integer	None	No
snapshotID	Retrieves information for an individual snapshot ID.	integer	None	No

Return value

This method has the following return value:

Name	Description	Type
snapshots	Information about each snapshot for each volume. If volumeID is not provided, all snapshots for all volumes are returned. Snapshots that are in a group are returned with a group ID.	snapshot array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListSnapshots",
   "params": {
             "volumeID": "1"
       },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "snapshots": [
        {
          "attributes": {},
          "checksum": "0x0",
          "createTime": "2015-05-08T13:15:00Z",
          "enableRemoteReplication": true,
          "expirationReason": "None",
          "expirationTime": "2015-05-08T21:15:00Z",
          "groupID": 0,
          "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",
          "name": "Hourly",
          "remoteStatuses": [
              {
                "remoteStatus": "Present",
                "volumePairUUID": "237e1cf9-fb4a-49de-a089-a6a9a1f0361e"
              }
         ],
          "snapshotID": 572,
          "snapshotUUID": "efa98e40-cb36-4c20-a090-a36c48296c14",
          "status": "done",
          "totalSize": 10000269312,
          "volumeID": 1
        }
      ]
    }
  }
```

New since version

9.6

ModifyGroupSnapshot

You can use `ModifyGroupSnapshot` to change the attributes of a group of snapshots. You can also use this method to enable snapshots created on the read/write (source) volume to be remotely replicated to a target storage system.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
enableRemoteReplication	Use to enable the snapshot created to be replicated to a remote cluster. Possible values: • `true`: The snapshot will be replicated to remote storage. • `false`: The snapshot will not be replicated to remote storage.	boolean	false	No

expirationTime	Specify the time after which the snapshot can be removed. Cannot be used with retention. If neither expirationTime, or retention on the original snapshot, are specified, the snapshot will not expire. The time format is an ISO 8601 date string for time based expiration, otherwise it will not expire. A value of `null` causes the snapshot to be retained permanently. A value of fifo causes the snapshot to be preserved on a First-In-First-Out (FIFO) basis, relative to other FIFO snapshots on the volume. The API will fail if no FIFO space is available.	ISO 8601 date string	None	No
name	The name of the group snapshot. If no name is entered, the date and time the group snapshot was taken is used. The maximum name length allowed is 255 characters.	string	None	No
groupSnapshotID	The ID of the group of snapshots.	string	None	Yes
snapMirrorLabel	The label used by SnapMirror software to specify the snapshot retention policy on a SnapMirror endpoint.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
groupSnapshot	Object containing information about the newly modified group snapshot.	groupSnapshot

Request example

Requests for this method are similar to the following example:

```
{
  "id": 695,
  "method": "ModifyGroupSnapshot",
  "params": {
    "groupSnapshotID": 3,
    "enableRemoteReplication": true,
    "expirationTime": "2016-04-08T22:46:25Z"
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 695,
  "result": {
    "groupSnapshot": {
      "attributes": {},
      "createTime": "2016-04-06T17:31:41Z",
      "groupSnapshotID": 3,
      "groupSnapshotUUID": "8b2e101d-c5ab-4a72-9671-6f239de49171",
      "members": [
        {
          "attributes": {},
          "checksum": "0x0",
          "createTime": "2016-04-06T17:31:41Z",
          "enableRemoteReplication": true,
          "expirationReason": "None",
          "expirationTime": "2016-04-08T22:46:25Z",
          "groupID": 3,
          "groupSnapshotUUID": "8b2e101d-c5ab-4a72-9671-6f239de49171",
          "name": "grpsnap1-2",
          "snapshotID": 2,
          "snapshotUUID": "719b162c-e170-4d80-b4c7-1282ed88f4e1",
          "status": "done",
          "totalSize": 1000341504,
          "virtualVolumeID": null,
          "volumeID": 2
        }
      ],
      "name": "grpsnap1",
      "status": "done"
    }
  }
}
```


New since version

9.6

ModifySchedule

You can use `ModifySchedule` to change the intervals at which a scheduled snapshot occurs. You can also delete or pause a schedule by using this method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
attributes	Use to change the frequency of the snapshot occurrence. Possible values: • `Days of Week` • `Days of Month` • `Time Interval`	JSON object	None	No
hours	Number of hours between snapshots or hour at which the snapshot will occur in Days of Week or Days of Month mode. Valid values are 0 through 24.	string	None	No
name	The name of the snapshot. If no name is entered, the date and time the group snapshot was taken is used. The maximum name length allowed is 244 characters.	string	None	No
minutes	Number of minutes between snapshots or minute at which snapshot will occur in Days of Week or Days of Month mode. Valid values are 0 through 59.	integer	None	No
lastRunStatus	The result or status of the last scheduled snapshot creation.	string	None	No

paused	Indicates if the schedule should be paused or not. Valid values: • `true` • `false`	boolean	None	No
recurring	Indicates if the schedule will be recurring or not. Valid values are: • `true` • `false`	boolean	None	No
runNextInterval	Use to choose whether or not to run the snapshot the next time the scheduler is active. Valid values: • `true` • `false` When set to true, the scheduled snapshot runs the next time the scheduler is active, and then resets back to false.	boolean	false	No
scheduleID	Unique ID of the schedule.	integer	None	Yes
scheduleName	Unique name for the schedule. The maximum schedule name length allowed is 244 characters.	string	None	No
scheduleType	Indicates the type of schedule to create. The only supported value is `snapshot`.	string	None	Yes

`scheduleInfo`	The unique name given to the schedule, the retention period for the snapshot that was created, and the volume ID of the volume from which the snapshot was created. Valid values: • `enableRemoteReplication`: Indicates if the snapshot should be included in remote replication. (boolean) • `ensureSerialCreation`: Specifies whether a new snapshot creation should be allowed if a previous snapshot replication is in progress. (boolean) • `name`: The snapshot name to be used. (string) • `retention`: The amount of time the snapshot is retained. Depending on the time, it displays in one of the following formats: ◦ `fifo`: The snapshot is retained on a First-In-First-Out (FIFO) basis. If empty, the snapshot is retained	schedule	None	No

snapMirrorLabel	The label used by SnapMirror software to specify the snapshot retention policy on a SnapMirror endpoint.	string	None	No
toBeDeleted	Indicates if the schedule is marked for deletion. Valid values: • `true` • `false`	boolean	None	No
startingDate	Indicates the date the first time the schedule began or will begin.	ISO 8601 date string	None	No
monthdays	The days of the month that a snapshot will be made. Valid values are 1 through 31.	integer array	None	Yes
weekdays	Day of the week the snapshot is to be created. The day of the week starts at Sunday with the value of 0 and an offset of 1.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
schedule	An object containing the modified schedule attributes.	schedule

Request example

```
{
  "method": "ModifySchedule",
  "params": {
    "scheduleName" : "Chicago",
    "scheduleID" : 3
    },
  "id": 1
}
```

Response example

```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Days Of Week"
            },
      "hasError": false,
      "hours": 5,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": null,
      "minutes": 0,
      "monthdays": [],
      "paused": false,
      "recurring": true,
      "runNextInterval": false,
      "scheduleID": 3,
      "scheduleInfo": {
        "volumeID": "2"
            },
      "scheduleName": "Chicago",
      "scheduleType": "Snapshot",
      "startingDate": null,
      "toBeDeleted": false,
      "weekdays": [
        {
          "day": 2,
          "offset": 1
      }
      ]
    }
  }
}
```

New since version

9.6

ModifySnapshot

You can use `ModifySnapshot` to change the attributes currently assigned to a snapshot. You can also use this method to enable snapshots created on the read/write (source) volume to be remotely replicated to a target storage cluster running Element software.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
enableRemoteReplication	Use to enable the snapshot created to be replicated to a remote storage cluster. Possible values: • `true`: The snapshot will be replicated to remote storage. • `false`:The snapshot will not be replicated to remote storage.	boolean	false	No

expirationTime	Specify the time after which the snapshot can be removed. Cannot be used with retention. If neither expirationTime, or retention on the original snapshot, are specified, the snapshot will not expire. The time format is an ISO 8601 date string for time based expiration, otherwise it will not expire. A value of null causes the snapshot to be retained permanently. A value of fifo causes the snapshot to be preserved on a First-In-First-Out (FIFO) basis, relative to other FIFO snapshots on the volume. The API will fail if no FIFO space is available.	ISO 8601 date string	None	No
name	The name of the snapshot. If no name is entered, the date and time the snapshot was taken is used. The maximum name length allowed is 255 characters.	string	None	No
snapMirrorLabel	The label used by SnapMirror software to specify the snapshot retention policy on a SnapMirror endpoint.	string	None	No
snapshotID	Identifier of the snapshot.	string	None	Yes

Return value

This method has the following return value:

Name	Description	Type
snapshot	An object containing information about the newly modified snapshot.	snapshot

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifySnapshot",
  "params": {
    "snapshotID": 3114,
    "enableRemoteReplication": "true",
    "name" : "Chicago"
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "snapshot": {
      "attributes": {},
      "checksum": "0x0",
      "createTime": "2016-04-04T17:26:20Z",
      "enableRemoteReplication": true,
      "expirationReason": "None",
      "expirationTime": null,
      "groupID": 0,
      "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",
      "name": "test1",
      "snapshotID": 3114,
      "snapshotUUID": "5809a671-4ad0-4a76-9bf6-01cccf1e65eb",
      "status": "done",
      "totalSize": 5000658944,
      "virtualVolumeID": null,
      "volumeID": 1
    }
  }
}
```


New since version

9.6

RollbackToGroupSnapshot

You can use `RollbackToGroupSnapshot` to roll back all individual volumes in a snapshot group to each volume's individual snapshot.

Rolling back to a group snapshot creates a temporary snapshot of each volume within the group snapshot.

- Creating a snapshot is allowed if cluster fullness is at stage 1, 2, or 3. Snapshots are not created when cluster fullness is at stage 4 or 5.
- Rolling back volumes to a group snapshot might fail when slice synchronization is in progress. Retry `RollbackToGroupSnapshot` after syncing completes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
`groupSnapshotID`	Unique ID of the group snapshot.	integer	None	Yes
`attributes`	List of name-value pairs in JSON object format.	JSON object	None	No
`name`	Name for the group snapshot of the volume's current state that is created if `saveCurrentState` is set to true. If you do not give a name, then the name of the snapshots (group and individual volume) are set to a timestamp of the time that the rollback occurred.	string	None	No
`saveCurrentState`	Specifies whether to save the previous active volume image or not. Valid values: • `true`: The previous active volume image is kept. • `false`: The previous active volume image is deleted.	boolean	false	No

Return values

This method has the following return values:

Name	Description	Type

members	An array containing volumeIDs and snapshotIDs of members of the group snapshot. Values: • checksum: A small string representation of the data in the stored snapshot. This checksum can be used later to compare other snapshots to detect errors in the data. (string) • snapshotID: Unique ID of a snapshot from which the new snapshot is made. The snapshotID must be a snapshot on the given volume. (integer) • volumeID: The source volume ID for the snapshot. (integer)	JSON object array
groupSnapshotID	If `saveCurrentState` was set to false, this value is null. If `saveCurrentState` was set to true, the unique ID of the newly created group snapshot.	integer
groupSnapshot	If `saveCurrentState` was set to false, this value is null. If `saveCurrentState` was set to true, an object containing information about the group snapshot which `RollbackToGroupSnapshot` just rolled back to.	groupSnapshot

Request example

Requests for this method are similar to the following example:

```
{
  "id": 438,
  "method": "RollbackToGroupSnapshot",
  "params": {
    "groupSnapshotID": 1,
    "name": "grpsnap1",
    "saveCurrentState": true
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 438,
  "result": {
    "groupSnapshot": {
      "attributes": {},
      "createTime": "2016-04-06T17:27:17Z",
      "groupSnapshotID": 1,
      "groupSnapshotUUID": "468fe181-0002-4b1d-ae7f-8b2a5c171eee",
      "members": [
        {
          "attributes": {},
          "checksum": "0x0",
          "createTime": "2016-04-06T17:27:17Z",
          "enableRemoteReplication": false,
          "expirationReason": "None",
          "expirationTime": null,
          "groupID": 1,
          "groupSnapshotUUID": "468fe181-0002-4b1d-ae7f-8b2a5c171eee",
          "name": "2016-04-06T17:27:17Z",
          "snapshotID": 4,
          "snapshotUUID": "03563c5e-51c4-4e3b-a256-a4d0e6b7959d",
          "status": "done",
          "totalSize": 1000341504,
          "virtualVolumeID": null,
          "volumeID": 2
        }
      ],
      "name": "2016-04-06T17:27:17Z",
      "status": "done"
    },
    "groupSnapshotID": 3,
    "members": [
      {
        "checksum": "0x0",
        "snapshotID": 2,
        "snapshotUUID": "719b162c-e170-4d80-b4c7-1282ed88f4e1",
        "volumeID": 2
      }
    ]
  }
}
```


New since version

9.6

RollbackToSnapshot

You can use the `RollbackToSnapshot` method to make an existing snapshot of the active volume image. This method creates a new snapshot from an existing snapshot.

The new snapshot becomes active and the existing snapshot is preserved until it is manually deleted. The previously active snapshot is deleted unless you set the saveCurrentState parameter to true.

CLUSTER_FULLNESS

- You can create snapshots if cluster fullness is at stage 1, 2, or 3. You cannot create snapshots when cluster fullness reaches stage 4 or 5.
- Rolling back a volume to a snapshot might fail when slice synchronization is in progress. Retry `RollbackToSnapshot` after syncing completes.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
volumeID	VolumeID for the volume.	integer	None	Yes
attributes	List of name-value pairs in JSON object format.	JSON attributes	None	No
name	Name for the snapshot. If no name is given, the name of the snapshot being rolled back to is used with "- copy" appended to the end of the name.	string	None	No
snapshotID	ID of a previously created snapshot on the given volume.	integer	None	Yes

Name	Description	Type	Default value	Required
saveCurrentState	Specifies whether to save previous active volume image or not. Valid values: • true: The previous active volume image is kept. • false: The previous active volume image is deleted.	boolean	false	No

Return values

This method has the following return values:

Name	Description	Type
checksum	A small string representation of the data in the stored snapshot.	string
snapshotID	If saveCurrentState was set to false, this value is null. If saveCurrentState was set to true, the unique ID of the newly created snapshot.	integer
snapshot	If saveCurrentState was set to false, this value is null. If saveCurrentState was set to true, an object containing information about the newly created snapshot.	snapshot

Request example

Requests for this method are similar to the following example:

```
{
 "method": "RollbackToSnapshot",
 "params": {
    "volumeID": 1,
    "snapshotID": 3114,
    "saveCurrentState": true
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "checksum": "0x0",
    "snapshot": {
      "attributes": {},
      "checksum": "0x0",
      "createTime": "2016-04-04T17:27:32Z",
      "enableRemoteReplication": false,
      "expirationReason": "None",
      "expirationTime": null,
      "groupID": 0,
      "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",
      "name": "test1-copy",
      "snapshotID": 1,
      "snapshotUUID": "30d7e3fe-0570-4d94-a8d5-3cc8097a6bfb",
      "status": "done",
      "totalSize": 5000658944,
      "virtualVolumeID": null,
      "volumeID": 1
    },
    "snapshotID": 1
  }
}
```

New since version

9.6

Virtual volume API methods

Element software virtual volume API methods enable you to manage virtual volumes (VVols). You can view existing VVols with these API methods as well as create, modify, and delete virtual volume storage containers. Although you cannot use these methods to operate on normal volumes, you can use the normal volume API methods to list information about VVols.

- CreateStorageContainer
- DeleteStorageContainers
- GetStorageContainerEfficiency
- GetVirtualVolumeCount
- ListProtocolEndpoints
- ListStorageContainers
- ListVirtualVolumeBindings
- ListVirtualVolumeHosts
- ListVirtualVolumes
- ListVirtualVolumeTasks
- ModifyStorageContainer

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

CreateStorageContainer

You can use the `CreateStorageContainer` method to create a Virtual Volume (VVol) storage container. You can use storage containers for reporting and resource allocation. You need to create at least one storage container to use the Virtual Volumes feature.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	Name of the storage container. Follows Element software account naming restrictions.	string	None	Yes

Name	Description	Type	Default value	Required
accountID	Non-storage container account that will become a storage container.	integer	None	No
initiatorSecret	The secret for CHAP authentication for the initiator.	string	None	No
targetSecret	The secret for CHAP authentication for the target.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
storageContainer	Object containing Information about the newly created storage container.	storageContainer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CreateStorageContainer",
    "params": {
        "name" : "example"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "storageContainer": {
      "accountID": 8,
      "initiatorSecret": "rVTOi25^H.d;cP}l",
      "name": "example",
      "protocolEndpointType": "SCSI",
      "status": "active",
      "storageContainerID": "a9ec1138-e386-4a44-90d7-b9acbbc05176",
      "targetSecret": "6?AEIxWpvo6,!boM"
    }
  }
}
```


New since version

9.6

DeleteStorageContainers

You can use the `DeleteStorageContainers` method to remove up to 2000 Virtual Volume (VVol) storage containers from the system at one time. The storage containers you remove must not contain any VVols.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
storageContainerIDs	A list of IDs of the storage containers to delete. You can specify up to 2000 IDs in the list.	UUID array	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DeleteStorageContainers",
    "params": {
        "storageContainerIDs" : ["a9ec1138-e386-4a44-90d7-b9acbbc05176"]
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {}
}
```

New since version

9.6

GetStorageContainerEfficiency

You can use the `GetStorageContainerEfficiency` method to retrieve efficiency information about a virtual volume storage container.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
storageContainerID	The ID of the storage container for which to retrieve efficiency information.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type

compression	The amount of space saved by data compression for all virtual volumes in the storage container. Stated as a ratio where a value of 1 means data has been stored with no compression.	float
deduplication	The amount of space saved by not duplicating data for all virtual volumes in the storage container. Stated as a ratio.	float
missingVolumes	The virtual volumes that could not be queried for efficiency data. Missing volumes can be caused by the Garbage Collection (GC) cycle being less than an hour old, temporary loss of network connectivity, or restarted services since the GC cycle.	integer array
thinProvisioning	The ratio of space used to the amount of space allocated for storing data. Stated as a ratio.	float
timestamp	The last time efficiency data was collected after GC.	ISO 8601 data string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetStorageContainerEfficiency",
  "params": {
    "storageContainerID" : "6c95e24f-9f0b-4793-affb-5a4bc6c3d7e1"
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "compression": 1,
    "deduplication": 1,
    "missingVolumes": [],
    "thinProvisioning": 1,
    "timestamp": "2016-04-12T15:39:49Z"
  }
}
```

New since version

9.6

GetVirtualVolumeCount

You can use the `GetVirtualVolumeCount` method to retrieve the number of virtual volumes currently in the system.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
count	The number of virtual volumes currently in the system.	integer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetVirtualVolumeCount",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "count": 5
  }
}
```

New since version

9.6

ListProtocolEndpoints

You can use the `ListProtocolEndpoints` method to retrieve information about all protocol endpoints in the cluster. Protocol endpoints govern access to their associated virtual volume storage containers.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
protocolEndpointIDs	A list of protocol endpoint IDs for which to retrieve information. If you omit this parameter, the method returns information about all protocol endpoints.	protocolEndpointID UUID array	None	No

Return values

This method has the following return value:

Name	Description	Type
protocolEndpoints	List of objects containing information about each protocol endpoint in the system.	protocolEndpoint array

Request example

Requests for this method are similar to the following example:

```
{
  "id": 1,
  "method": "ListProtocolEndpoints",
  "params": {}
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "protocolEndpoints": [
      {
        "primaryProviderID": 1,
        "protocolEndpointID": "1387e257-d2e3-4446-be6d-39db71583e7b",
        "protocolEndpointState": "Active",
        "providerType": "Primary",
        "scsiNAADeviceID": "6f47acc2000000016970687200000000",
        "secondaryProviderID": 2
      },
      {
        "primaryProviderID": 2,
        "protocolEndpointID": "1f16ed86-3f31-4c76-b004-a1251187700b",
        "protocolEndpointState": "Active",
        "providerType": "Primary",
        "scsiNAADeviceID": "6f47acc2000000026970687200000000",
        "secondaryProviderID": 3
      },
      {
        "primaryProviderID": 4,
        "protocolEndpointID": "c6458dfe-9803-4350-bb4e-68a3feb7e830",
        "protocolEndpointState": "Active",
        "providerType": "Primary",
        "scsiNAADeviceID": "6f47acc2000000046970687200000000",
        "secondaryProviderID": 1
      },
      {
        "primaryProviderID": 3,
        "protocolEndpointID": "f3e7911d-0e86-4776-97db-7468c272213f",
        "protocolEndpointState": "Active",
        "providerType": "Primary",
        "scsiNAADeviceID": "6f47acc2000000036970687200000000",
        "secondaryProviderID": 4
      }
    ]
  }
}
```


New since version

9.6

ListStorageContainers

You can use the `ListStorageContainers` method to retrieve information about all virtual volume storage containers known to the system.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
storageContainerIDs	A list of storage container IDs for which to retrieve information. If you omit this parameter, the method returns information about all storage containers in the system.	UUID array	None	No

Return value

This method has the following return value:

Name	Description	Type
storageContainers	List of objects containing information about all storage containers in the system.	storageContainer array

Request example

Requests for this method are similar to the following example:

```
{
   "method": "ListStorageContainers",
   "params": {
      "storageContainerIDs": ["efda8307-b916-4424-979e-658a3f16894d"]
   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 6395,
  "result": {
    "storageContainers": [
      {
        "accountID": 64,
        "initiatorSecret": "EJ:08An1MyNQmL!7",
        "name": "VvolContainer",
        "protocolEndpointType": "SCSI",
        "status": "active",
        "storageContainerID": "efda8307-b916-4424-979e-658a3f16894d",
        "targetSecret": "g38}zWBK%206jQr~",
        "virtualVolumes": []
      }
    ]
  }
}
```


New since version

9.6

ListVirtualVolumeBindings

You can use the `ListVirtualVolumeBindings` method to get a list of all virtual volumes in the cluster that are bound to protocol endpoints.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
virtualVolumeBindingIDs	A list of virtual volume binding IDs for which to retrieve information. If you omit this parameter, the method returns information about all virtual volume bindings.	integer array	None	No

Return value

This method has the following return value:

Name	Description	Type
bindings	A list of objects describing all virtual volumes in the cluster that are bound to protocol endpoints.	binding

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVirtualVolumeBindings",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "bindings": [
      {
        "protocolEndpointID": "5dd53da0-b9b7-43f9-9b7e-b41c2558e92b",
        "protocolEndpointInBandID":
"naa.6f47acc2000000016a67746700000000",
        "protocolEndpointType": "SCSI",
        "virtualVolumeBindingID": 177,
        "virtualVolumeHostID": "564de1a4-9a99-da0f-8b7c-3a41dfd64bf1",
        "virtualVolumeID": "269d3378-1ca6-4175-a18f-6d4839e5c746",
        "virtualVolumeSecondaryID": "0xe200000000a6"
      }
    ]
  }
}
```

New since version

9.6

ListVirtualVolumeHosts

You can use the `ListVirtualVolumeHosts` method to get a list of all virtual volume hosts known to the cluster. A virtual volume host is a VMware ESX host that has initiated a session with the VASA API provider.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
virtualVolumeHostIDs	A list of virtual volume host IDs for which to retrieve information. If you omit this parameter, the method returns information about all virtual volume hosts.	virtualVolumeHostID UUID array	None	No

Return value

This method has the following return value:

Name	Description	Type
hosts	A list of objects describing the virtual volume hosts in the cluster.	host array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVirtualVolumeHosts",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "hosts": [
      {
        "bindings": [],
        "clusterID": "5ebdb4ad-9617-4647-adfd-c1013578483b",
        "hostAddress": "172.30.89.117",
        "initiatorNames": [
          "iqn.1998-01.com.vmware:zdc-dhcp-0-c-29-d6-4b-f1-1a0cd614",
          "iqn.1998-01.com.vmware:zdc-dhcp-0-c-29-d6-4b-f1-5bcf9254"
        ],
        "virtualVolumeHostID": "564de1a4-9a99-da0f-8b7c-3a41dfd64bf1",
        "visibleProtocolEndpointIDs": [
          "5dd53da0-b9b7-43f9-9b7e-b41c2558e92b"
        ]
      }
    ]
  }
}
```


New since version

9.6

ListVirtualVolumes

You can use the `ListVirtualVolumes` method to list the virtual volumes currently in the system. You can use this method to list all virtual volumes, or only list a subset.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
details	The level of detail in the response. Possible values: • true: Include more details about each VVol in the response. • false: Include the standard level of detail about each VVol in the response.	boolean	False	No
limit	The maximum number of virtual volumes to list.	integer	10000	No
recursive	Specifies whether to include information about the children of each VVol in the response or not. Possible values: • true: Include information about the children of each VVol in the response. • false: Do not include information about the children of each VVol in the response.	boolean	False	No
startVirtualVolumeID	The ID of the virtual volume at which to begin the list in the response.	UUIDType	None	No

Name	Description	Type	Default value	Required
virtualVolumeIDs	A list of virtual volume IDs for which to retrieve information. If you omit this parameter, the method returns information about only these virtual volumes.	virtualVolumeID UUID array	None	No

Return values

This method has the following return values:

Name	Description	Type
nextVirtualVolumeID	The ID of the next virtual volume in the list.	UUID
virtualVolumes	A list of objects describing the virtual volumes currently in the system.	virtualVolume array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVirtualVolumes",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "nextVirtualVolumeID": "00000000-0000-0000-0000-000000000000",
    "virtualVolumes": [
      {
        "bindings": [
          177
        ],
        "children": [],
        "metadata": {
          "SFProfileId": "f4e5bade-15a2-4805-bf8e-52318c4ce443",
          "SFgenerationId": "0",
          "VMW_ContainerId": "abaab415-bedc-44cd-98b8-f37495884db0",
          "VMW_VVolName": "asdf",
          "VMW_VVolType": "Config",
          "VMW_VmID": "502e0676-e510-ccdd-394c-667f6867fcdf",
          "VMW_VvolProfile": "f4e5bade-15a2-4805-bf8e-52318c4ce443:0"
        },
        "parentVirtualVolumeID": "00000000-0000-0000-0000-000000000000",
        "snapshotID": 0,
        "snapshotInfo": null,
        "status": "done",
        "storageContainer": {
          "accountID": 1,
          "initiatorSecret": "B5)D1y10K)8IDN58",
          "name": "test",
          "protocolEndpointType": "SCSI",
          "status": "active",
          "storageContainerID": "abaab415-bedc-44cd-98b8-f37495884db0",
          "targetSecret": "qgae@{o{~8\"2U)U^"
        },
        "virtualVolumeID": "269d3378-1ca6-4175-a18f-6d4839e5c746",
        "virtualVolumeType": "config",
        "volumeID": 166,
        "volumeInfo": null
      }
    ]
  }
}
```


New since version

9.6

ListVirtualVolumeTasks

You can use the `ListVirtualVolumeTasks` method to get a list of virtual volume tasks in the system.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
virtualVolumeTaskIDs	A list of virtual volume task IDs for which to retrieve information. If you omit this parameter, the method returns information about all virtual volume tasks.	UUID array	None	No

Return value

This method has the following return value:

Name	Description	Type
tasks	A list of objects describing the virtual volume tasks in the cluster.	task array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListVirtualVolumeTasks",
    "params": {
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "tasks": [
      {
        "cancelled": false,
        "cloneVirtualVolumeID": "fafeb3a0-7dd9-4c9f-8a07-80e0bbf6f4d0",
        "operation": "clone",
        "parentMetadata": {
          "SFProfileId": "f4e5bade-15a2-4805-bf8e-52318c4ce443",
          "SFgenerationId": "0",
          "VMW_ContainerId": "abaab415-bedc-44cd-98b8-f37495884db0",
          "VMW_GosType": "windows7Server64Guest",
          "VMW_VVolName": "asdf.vmdk",
          "VMW_VVolNamespace": "/vmfs/volumes/vvol:abaab415bedc44cd-
98b8f37495884db0/rfc4122.269d3378-1ca6-4175-a18f-6d4839e5c746",
          "VMW_VVolType": "Data",
          "VMW_VmID": "502e0676-e510-ccdd-394c-667f6867fcdf",
          "VMW_VvolAllocationType": "4",
          "VMW_VvolProfile": "f4e5bade-15a2-4805-bf8e-52318c4ce443:0"
        },
        "parentTotalSize": 42949672960,
        "parentUsedSize": 0,
        "status": "success",
        "virtualVolumeHostID": "564de1a4-9a99-da0f-8b7c-3a41dfd64bf1",
        "virtualVolumeTaskID": "a1b72df7-66a6-489a-86e4-538d0dbe05bf",
        "virtualvolumeID": "fafeb3a0-7dd9-4c9f-8a07-80e0bbf6f4d0"
      }
    ]
  }
}
```


New since version

9.6

ModifyStorageContainer

You can use the `ModifyStorageContainer` method to make changes to an existing virtual volume storage container.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required

storageContainerID	The unique ID of the virtual volume storage container to modify.	UUID	None	Yes
initiatorSecret	The new secret for CHAP authentication for the initiator.	string	None	No
targetSecret	The new secret for CHAP authentication for the target.	string	None	No

Return values

This method has the following return value:

Name	Description	Type
storageContainer	Information about the newly created storage container.	storageContainer

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ModifyStorageContainer",
    "params": {
      "storageContainerID": "6c95e24f-9f0b-4793-affb-5a4bc6c3d7e1",
      "targetSecret": "O,IM;tOQdn9$JJ*8"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "storageContainer": {
      "accountID": 8,
      "initiatorSecret": "T$|5TO>2IY5sk4@k",
      "name": "doctest1",
      "protocolEndpointType": "SCSI",
      "status": "active",
      "storageContainerID": "6c95e24f-9f0b-4793-affb-5a4bc6c3d7e1",
      "targetSecret": "O,IM;tOQdn9$JJ*8"
    }
  }
}
```


New since version

9.6

Access control

The Element API methods available vary based on the type of access you set.

accounts

The following methods are available to the accounts access type:

AddAccount
GetAccountByID
ModifyAccount
GetAccountByName
ListAccounts
GetAccountEfficiency
RemoveAccount

administrator

All methods are available to the administrator access type.

clusterAdmin

The following methods are available to the cluster admin access type:

AddClusterAdmin
ListBackupTargets
AddInitiatorsToVolumeAccessGroup
ListBulkVolumeJobs
AddLdapClusterAdmin
ListClusterAdmins
AddVirtualNetwork
ListClusterPairs
AddVirtualNetwork
ListNodeFibreChannelPortInfo
AddVolumetoVolumeAccessGroup
ListBackupTargets
CloneMultipleVolumes
ListDriveHardware
CompleteClusterPairing
ListFibreChannelSessions
CompleteVolumePairing
ListFibreChannelPortInfo
CreateBackupTarget
ListGroupSnapshots
CreateSchedule

ListActivePairedVolumes
CreateSnapshot
ModifyBackupTarget
CreateSupportBundle
ModifyClusterAdmin
CreateClusterSupportBundle
ModifyGroupSnapshot
CreateGroupSnapshot
ModifyClusterFullThreshold
CreateVolumeAccessGroup
ModifyVolumeAccessGroup
DeleteAllSupportBundles
ModifyVolumeAccessGroupLunAssignments
DeleteSnapshot
ModifyVolumePair
DeleteGroupSnapshot
ModifyVirtualNetwork
DeleteVolumeAccessGroup
RemoveClusterAdmin
DisableEncryptionAtRest
RemoveVolumePair
DisableLdapAuthentication

RemoveVirtualNetwork
DisableSnmp
RemoveVolumesFromVolumeAccessGroup
EnableEncryptionAtRest
RemoveInitiatorsFromVolumeAccessGroup
EnableLdapAuthentication
RollbackToSnapshot
EnableSnmp
RollbackToGroupSnapshot
GetBackupTarget
SetLoginSessionInfo
GetClusterFullThreshold
SetNtpInfo
GetClusterMasterNodeID
SetSnmpACL
GetHardwareConfig
SetSnmpInfo
GetLdapConfiguration
SetSnmpTrapInfo
GetLoginSessionInfo
SetRemoteLoggingHosts
GetNtpInfo

Shutdown
GetNvramInfo
StartBulkVolumeRead
GetRawStats
StartBulkVolumeWrite
GetSnmpACL
StartClusterPairing
GetVolumeAccessGroupEfficiency
StartVolumePairing
GetVolumeAccessLunAssignments
TestLdapAuthentication
GetVirtualNetwork

drives

The following methods are available to the drives access type:

ListDrives
RemoveDrives
AddDrives
SecureEraseDrives

nodes

The following methods are available to the nodes access type:

AddNodes
ListPendingNodes

ListActiveNodes
RemoveNodes

read

The following methods are available to the read access type:

GetAccountByID
ListCloneJobs
GetAccountByName
ListDeletedVolumes
GetAsyncResult
ListDriveHardware
GetClusterCapacity
ListDrives
GetDefaultQoS
ListEvents
GetDriveStats
ListISCSISessions
GetSoftwareUpgrade
ListPendingNodes
GetVolumeStats
ListSyncJobs
ListAccounts
ListVolumeAccessGroups

ListActiveNodes
ListVolumeStatsByAccount
ListActiveNodes
ListVolumeStatsByVolume
ListActiveVolumes
ListVolumeStatsByVolumeAccessGroup
ListAllNodes
ListVolumesForAccount
ListBackupTargets

reporting

The following methods are available to the reporting access type:

ClearClusterFaults
GetVolumeEfficiency
GetAccountEfficiency
GetVolumeStats
GetClusterCapacity
ListCloneJobs
GetClusterHardwareInfo
ListClusterFaults
GetClusterInfo
ListClusterPairs
GetClusterMasterNodeID

ListDriveHardware
GetClusterStats
ListEvents
GetDriveHardwareInfo
ListISCSISessions
GetDriveStats
ListSchedules
GetNetworkConfig
ListServices
GetNodeHardwareInfo
ListSyncJobs
GetNodeStats
ListVirtualNetworks
GetSnmpInfo
ListVolumeStatsByAccount
GetSnmpTrapInfo
ListVolumeStatsByVolume
GetVolumeAccessGroupEfficiency
ListVolumeStatsByVolumeAccessGroup

repositories

The ListAllNodes method is available to the repositories access type.

volumes

The following methods are available to the volumes access type:

CreateVolume
DeleteVolume
ModifyBackupTarget
CloneVolume
DeleteVolumePairing
ModifyVolumes
CloneMultipleVolumes
GetBackupTarget
ModifyVolumePair
CreateBackupTarget
GetDefaultQoS
PurgeDeletedVolume
CreateSnapshot
ListActiveVolumes
RemoveBackupTarget
CreateGroupSnapshot
ListBackupTarget
RemoveVolumePair
CompleteVolumePairing
ListGroupSnapshots
RestoreDeletedVolume
CloneMultipleVolumes

ListVolumesForAccount
RollbackToGroupSnapshot
DeleteGroupSnapshot
ListDeletedVolumes
RollbackToSnapshot
DeleteSnapshot
ListGroupSnapshots
StartBulkVolumeRead
StartBulkVolumeWrite
StartVolumePairing
UpdateBulkVolumeStatus

write

The following methods are available to the write access type:

AddDrives
RemoveNodes
AddNodes
RemoveAccount
AddAccount
RemoveVolumesFromVolumeAccessGroup
AddVolumeToVolumeAccessGroup
RemoveInitiatorsFromVolumeAccessGroup
AddInitiatorsToVolumeAccessGroup

DeleteVolumeAccessGroup
CreateVolumeAccessGroup
DeleteVolume
ModifyVolumeAccessGroup
RestoreDeletedVolume
ModifyAccount
PurgeDeletedVolume
CreateVolume
ModifyVolume
CloneVolume
GetAsyncResult
RemoveDrives

Response examples

Complete response examples are provided here.

- GetConfig
- GetClusterHardwareInfo
- GetLldpInfo
- GetNetworkConfig
- GetNodeHardwareInfo (output for iSCSI)
- GetNodeHardwareInfo (output for Fibre Channel nodes)
- GetNvramInfo
- ListActiveNodes
- ListActiveVolumes
- TestHardwareConfig

Find more information

- SolidFire and Element Software Documentation
- Documentation for earlier versions of NetApp SolidFire and Element products

GetConfig

The `GetConfig` method returns a response similar to the following example. Due to length, the response contains information for one node of the cluster only.


```
{
    "id": 1,
    "result": {
        "config": {
            "cluster": {
                "cipi": "Bond10G",
                "cluster": "AutoTest2-Fjqt",
                "encryptionCapable": true,
                "ensemble": [
                    "1:10.1.1.0",
                    "3:10.1.1.0",
                    "4:10.1.1.0"
                ],
                "mipi": "Bond1G",
                "name": "NLABP2605",
                "nodeID": 1,
                "pendingNodeID": 0,
                "role": "Storage",
                "sipi": "Bond10G",
                "state": "Active",
                "version": "11.0"
            },
            "network": {
            "Bond10G": {
                "#default": false,
                "address": "10.1.1.0",
                "auto": true,
                "bond-downdelay": "0",
                "bond-fail_over_mac": "None",
                "bond-miimon": "100",
                "bond-mode": "ActivePassive",
                "bond-primary_reselect": "Failure",
                "bond-slaves": "eth0 eth1",
                "bond-updelay": "200",
                "dns-nameservers": "10.1.1.0, 10.1.1.0",
                "dns-search": "ten.test.company.net., company.net.",
                "family": "inet",
                "gateway": "10.1.1.0",
                "linkSpeed": 10000,
                "macAddress": "c8:1f:66:ee:59:b9",
                "macAddressPermanent": "00:00:00:00:00:00",
```

```
                "method": "static",
                "mtu": "9000",
                "netmask": "255.255.240.0",
                "network": "10.1.1.0",
                "physical": {
                    "address": "10.1.1.0",
                    "macAddress": "c8:1f:66:ee:59:b9",
                    "macAddressPermanent": "00:00:00:00:00:00",
                    "mtu": "9000",
                    "netmask": "255.255.240.0",
                    "network": "10.1.1.0",
                    "upAndRunning": true
                },
                "routes": [],
                "status": "UpAndRunning",
                "symmetricRouteRules": [
                    "ip route add 10.1.1.1/20 dev Bond1G src 10.1.2.2
table Bond1G",
                    "ip rule add from 10.1.1.1 table Bond1G",
                    "ip route add default via 10.1.1.254"
                ],
                "upAndRunning": true,
                "virtualNetworkTag": "0"
            },
            "eth0": {
                "auto": true,
                "bond-master": "Bond10G",
                "family": "inet",
                "linkSpeed": 10000,
                "macAddress": "c8:1f:66:ee:59:b9",
                "macAddressPermanent": "c8:1f:66:ee:59:b9",
                "method": "bond",
                "physical": {
                    "address": "0.0.0.0",
                    "macAddress": "c8:1f:66:ee:59:b9",
                    "macAddressPermanent": "c8:1f:66:ee:59:b9",
                    "netmask": "N/A",
                    "network": "N/A",
                    "upAndRunning": true
                },
                "status": "UpAndRunning",
                "upAndRunning": true
            },
            "lo": {
                "auto": true,
                "family": "inet",
```

```
                "linkSpeed": 0,
                "macAddress": "00:00:00:00:00:00",
                "macAddressPermanent": "00:00:00:00:00:00",
                "method": "loopback",
                "physical": {
                    "address": "0.0.0.0",
                    "macAddress": "00:00:00:00:00:00",
                    "macAddressPermanent": "00:00:00:00:00:00",
                    "netmask": "N/A",
                    "network": "N/A",
                    "upAndRunning": true
                },
                "status": "UpAndRunning",
                "upAndRunning": true
            }
        }
        }
    }
}
```

GetClusterHardwareInfo

The `GetClusterHardwareInfo` method returns a response similar to the following example.

```
{
 "id": null,
  "result": {
   "clusterHardwareInfo": {
     "drives": {
      "1": {
        "description": "ATA      Drive",
        "dev": "8:0",
        "devpath": "/dev/disk/by-id/scsi-SATA_VRFSD3400GNCVMT205121562-
part4",
        "driveSecurityAtMaximum": false,
        "driveSecurityFrozen": true,
        "driveSecurityLocked": false,
        "logicalname": "/dev/sda",
        "product": "VRFSD3400GNCVMTJS1",
        "securityFeatureEnabled": false,
        "securityFeatureSupported": true,
        "serial": "205121562",
        "size": 299988156416,
        "uuid": "febe39ae-4984-edc0-e3a7-3c47608cface",
```

```
        "version": "515ABBF0"
      },
      "2": {...
      },
      "3": {...
      },
      "4": {...
      },
      "5": {...
      },
      "6": {...
      },
         .
         .
         .
    "44": {...
      }
      },
"nodes":{
  "1":{                        Storage Node
    "core_DMI:0200": {
    "description": "Motherboard",
    "physid": "0",
    "vendor": "SolidFire"
  },
    "fiber:0_PCI:0000:04:00.0": {
      "businfo": "pci@0000:04:00.0",
      "clock": "33000000",
      "description": "Fibre Channel",
      "physid": "0",
      "product": "ISP8324-based 16Gb Fibre Channel to PCI Express
Adapter",
      "vendor": "QLogic Corp.",
      "version": "02",
      "width": "64"
  },
    "Repeat fiber information": {...}
    "Repeat fiber": {...},
    "Repeat fiber": {...},
   }
 },
   "fans": {
     "Fan1A RPM": {
     "baseUnit": "RPM",
     "threshold": 840,
     "value": 4800
```

```
  },
      "Fan1B RPM": {...},
          .
          .
          .
      "Fan7B RPM": {...
      },
      "fibreChannelPorts": [
        {
         "firmware": "7.04.00 (d0d5)",
         "hbaPort": 1,
         "model": "QLE2672",
         "nPortID": "0x110c36",
         "pciSlot": 3,
         "serial": "BFE1341E09329",
         "speed": "8 Gbit",
         "state": "Online",
         "switchWwn": "20:01:00:2a:6a:a0:25:01",
         "wwnn": "5f:47:ac:c8:82:23:e0:00",
         "wwpn": "5f:47:ac:c0:82:23:e0:02"
        },
        {
         "firmware": "7.04.00 (d0d5)", {...}
         "firmware": "7.04.00 (d0d5)", {...}
         "firmware": "7.04.00 (d0d5)", {...}
        }
      ],
      "hardwareConfig": {
        "BIOS_REVISION": {
         "Passed": true,
         "actual": "1.1",
         "comparator": ">=",
         "expected": "1.0"
        },
        "BIOS_VENDOR": {
         "Passed": true,
         "actual": "SolidFire",
         "comparator": "==",
         "expected": "SolidFire"
        },
        "BIOS_VERSION": {
         "Passed": true,
         "actual": "1.1.2",
         "comparator": ">=",
         "expected": "1.1.2"
        },
```

```
      "BMC_FIRMWARE_REVISION": {
       "Passed": true,
       "actual": "1.6",
       "comparator": ">=",
       "expected": "1.6"
      },
      "BMC_IPMI_VERSION": {
       "Passed": true,
       "actual": "2.0",
       "comparator": ">=",
       "expected": "2.0"
      },
      "CHASSIS_TYPE": {
       "Passed": true,
       "actual": "R620",
       "comparator": "==",
       "expected": "R620"
      },
      "CPU_CORES_00": {
       "Passed": true,
       "actual": "6",
       "comparator": "==",
       "expected": "6"
      },
      "CPU_CORES_01": {
       "Passed": true,
       "actual": "6",
       "comparator": "==",
       "expected": "6"
      },
      "CPU_CORES_ENABLED_00": {
       "Passed": true,
       "actual": "6",
       "comparator": "==",
       "expected": "6"
      },
      "CPU_CORES_ENABLED_01": {
       "Passed": true,
       "actual": "6",
       "comparator": "==",
       "expected": "6"
      },
      "CPU_MODEL_00": {
       "Passed": true,
       "actual": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
       "comparator": "==",
```

```
 "expected": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz"
},
"CPU_MODEL_01": {
 "Passed": true,
 "actual": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
 "comparator": "==",
 "expected": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz"
},
"CPU_THREADS_00": {
 "Passed": true,
 "actual": "12",
 "comparator": "==",
 "expected": "12"
},
"CPU_THREADS_01": {
 "Passed": true,
 "actual": "12",
 "comparator": "==",
 "expected": "12"
},
"DRIVE_SIZE_BYTES_SDIMM0": {
 "Passed": true,
 "actual": "100030242816",
 "comparator": ">=",
 "expected": "100030242816"
},
"FIBRE_CHANNEL_FIRMWARE_REVISION": {
 "Passed": true,
 "actual": "FW:v7.04.00",
 "comparator": "==",
 "expected": "FW:v7.04.00"
},
"FIBRE_CHANNEL_MODEL": {
 "Passed": true,
 "actual": "QLE2672",
 "comparator": "==",
 "expected": "QLE2672"
},
"IDRAC_VERSION": {
 "Passed": true,
 "actual": "1.06.06",
 "comparator": ">=",
 "expected": "1.06.06"
},
"LIFECYCLE_VERSION": {
 "Passed": true,
```

```
        "actual": "1.0.0.5747",
        "comparator": ">=",
        "expected": "1.0.0.5747"
       },
       "MEMORY_GB": {
        "Passed": true,
        "actual": "32",
        "comparator": ">=",
        "expected": "32"
       },
       "MEMORY_MHZ_00": {
        "Passed": true,
        "actual": "1333",
        "comparator": ">=",
        "expected": "1333"
       },
       "MEMORY_MHZ_01": {
        "Passed": true,
        "actual": "1333",
        "comparator": ">=",
        "expected": "1333"
       },
       "MEMORY_MHZ_02": {
       "Passed": true,
       "actual": "1333",
       "comparator": ">=",
       "expected": "1333"
      },
      "MEMORY_MHZ_03": {
       "Passed": true,
       "actual": "1333",
       "comparator": ">=",
       "expected": "1333"
      },
      "NETWORK_DRIVER_ETH0": {
       "Passed": true,
       "actual": "bnx2x",
       "comparator": "=~",
       "expected": "^bnx2x$"
      },
      {
       "NETWORK_DRIVER_ETH1":, {...
      },
       "NETWORK_DRIVER_ETH2":, {...
      },
       "NETWORK_DRIVER_ETH3":, {...
```

```
      },
       "NETWORK_DRIVER_ETH4":, {...
      },
       "NETWORK_DRIVER_ETH5":, {...
      },
       "NODE_TYPE": {
       "Passed": true,
       "actual": "FC0025",
       "comparator": "==",
       "expected": "FC0025"
      },
      "NUM_CPU": {
       "Passed": true,
       "actual": "2",
       "comparator": "==",
       "expected": "2"
      },
      "NUM_DRIVES": {
       "Passed": true,
       "actual": "0",
       "comparator": "==",
       "expected": "0"
      },
      "NUM_DRIVES_INTERNAL": {
       "Passed": true,
       "actual": "1",
       "comparator": "==",
       "expected": "1"
      },
      "NUM_FIBRE_CHANNEL_PORTS": {
       "Passed": true,
       "actual": "4",
       "comparator": "==",
       "expected": "4"
      },
      "NVRAM_VENDOR": {
       "Passed": true,
       "actual": "",
       "comparator": "==",
       "expected": ""
      },
      "ROOT_DRIVE_REMOVABLE": {
       "Passed": true,
       "actual": "false",
       "comparator": "==",
       "expected": "false"
```

```
      }
      },
      "memory": {
        "firmware_": {
         "capacity": "8323072",
         "date": "03/08/2012",
         "description": "BIOS",
         "physid": "0",
         "size": "65536",
         "vendor": "SolidFire",
         "version": "1.1.2"
      },
      "memory_DMI:1000": {
        "description": "System Memory",
        "physid": "1000",
        "size": "34359738368",
        "slot": "System board or motherboard"
      }
      },
      "network": {
       "network:0_PCI:0000:01:00.0": {
        "businfo": "pci@0000:01:00.0",
        "capacity": "1000000000",
        "clock": "33000000",
        "description": "Ethernet interface",
        "logicalname": "eth0",
        "physid": "0",
        "product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
        "serial": "c8:1f:66:e0:97:2a",
        "vendor": "Broadcom Corporation",
        "version": "10",
        "width": "64"
      },
       "network:0_PCI:0000:41:00.0": {...
      },
       "network:1_PCI:0000:01:00.1": {...
      },
       "network:1_PCI:0000:41:00.1": {...
      },
       "network:2_PCI:0000:01:00.2": {...
      },
       "network:3_PCI:0000:01:00.3": {...
      }
      },
      "networkInterfaces": {
       "Bond10G": {
```

```
        "isConfigured": true,
        "isUp": true
      },
      "Bond1G": {
       "isConfigured": true,
       "isUp": true
      },
      "eth0": {
       "isConfigured": true,
       "isUp": true
      },
      "eth1": {...
      },
      "eth2": {...
      },
      "eth3": {...
      },
      "eth4": {...
      },
      "eth5": {...
      }
     },
     "nvram": {
       "errors": {
        "numOfErrorLogEntries": "0"
       },
       "extended": {
        "dialogVersion": "4",
        "event": [
        {
         "name": "flushToFlash",
         "time": "2015-08-06 01:19:39",
         "value": "0"
        },
        {
         "name": "flushToFlash",
         "time": "2015-08-06 01:26:44",
         "value": "0"
        },
        {... next "flushToFlash"
        },
        {... next "flushToFlash"
        },
        {... next "flushToFlash"
        },
        {... next "flushToFlash"
```

```
        },
        {... next "flushToFlash"
        },
        {... next "flushToFlash"
        },
        {... next "flushToFlash"
        }
      ],
      "eventOccurrences": [
        {
          "count": "740",
          "name": "flushToFlash"
        },
        {
          "count": "1",
          "name": "excessiveCurrent"
        }
      ],
      "initialCapacitance": "6.630 F",
      "initialEsr": "0.101 Ohm",
      "measurement": [
       {
         "level_0": " 0",
         "level_1": " 3969",
         "level_2": " 4631",
         "level_3": " 12875097",
         "level_4": " 1789948",
         "level_5": " 0",
         "level_6": " 0",
         "level_7": " 0",
         "level_8": " 0",
         "level_9": " 0",
         "name": "enterpriseFlashControllerTemperature",
         "recent": "66 C"
      },
      {
         "level_0": " 0",
         "level_1": " 58",
         "level_2": " 1479058",
         "level_3": " 12885356",
         "level_4": " 308293",
         "level_5": " 851",
         "level_6": " 29",
         "level_7": " 0",
         "level_8": " 0",
         "level_9": " 0",
```

```
        "name": "capacitor1And2Temperature",
        "recent": "30.69 C"
      },
      {...next temp measurement
      },
      {...next temp measurement
      },
      {...next temp measurement
      },
      }
       "name": "voltageOfCapacitor1",
       "recent": "2.198 V"
      },
      {
       "name": "voltageOfCapacitor2",
       "recent": "2.181 V"
      },
      {
       "name": "voltageOfCapacitor3",
       "recent": "2.189 V"
      },
      {
       "name": "voltageOfCapacitor4",
       "recent": "2.195 V"
      },
      {
       "level_0": " 4442034",
       "level_1": " 6800018",
       "level_2": " 2846869",
       "level_3": " 119140",
       "level_4": " 29506",
       "level_5": " 428935",
       "level_6": " 7143",
       "level_7": " 0",
       "level_8": " 0",
       "level_9": " 0",
       "name": "capacitorPackVoltage",
       "recent": "8.763 V"
      },
      {
       "level_0": " 0",
       "level_1": " 0",
       "level_2": " 0",
       "level_3": " 0",
       "level_4": " 189",
       "level_5": " 17",
```

```
      "level_6": " 36",
      "level_7": " 0",
      "level_8": " 2",
      "level_9": " 490",
      "name": "capacitorPackVoltageAtEndOfFlushToFlash",
      "recent": "4.636 V"
    },
    {
      "name": "currentDerivedFromV3V4",
      "recent": "-0.004 A"
    },
    {
      "level_0": " 230",
      "level_1": " 482",
      "level_2": " 22",
      "level_3": " 0",
      "level_4": " 0",
      "level_5": " 0",
      "level_6": " 0",
      "level_7": " 0",
      "level_8": " 0",
      "level_9": " 0",
      "name": "derivedEnergy",
      "recent": "172 Joules"
    },
    {...next voltage measurement
    },
    {...next voltage measurement
    },
    {...next voltage measurement
    },
  ],
  "smartCounters": [
    {
      "name": "numberOf512ByteBlocksReadFromDdr",
      "value": "10530088847"
    },
    {
      "name": "numberOf512ByteBlocksWrittenToDdr",
      "value": "1752499453837"
    },
    {
      "name": "numberOfHostReadCommands",
      "value": "235317769"
    },
    {...next smartCounters measurement
```

```
      },
      {...next smartCounters measurement
      },
      {...next smartCounters measurement
      },
    ],
    "snapshotTime": "2015-08-20 16:30:01"
  },
  "firmware": {
    "activeSlotNumber": "2",
    "slot1Version": "1e5817bc",
    "slot2Version": "5fb7565c",
    "slot3Version": "1e5817bc",
    "slot4Version": "1e5817bc"
  },
  "identify": {
    "firmwareVersion": "5fb7565c on slot 2",
    "hardwareRevision": "B04",
    "modelNumber": "RMS-200",
    "serialNumber": "0000862"
  },
  "smart": {
    "availableSpace": "0%",
    "availableSpaceThreshold": "0%",
    "controllerBusyTimeMinutes": "6793",
    "criticalErrorVector": "0x0",
    "mediaErrors": "0",
    "numberOf512ByteBlocksRead": "10530088847",
    "numberOf512ByteBlocksWritten": "1752499439063",
    "numberOfErrorInfoLogs": "1",
    "numberOfHostReadCommands": "235317769",
    "numberOfHostWriteCommands": "126030374065",
    "numberOfPowerCycles": "709",
    "powerOnHours": "11223",
    "temperature": "324 Kelvin",
    "unsafeShutdowns": "357"
    }
    },
    "origin": null,
    "platform": {
     "chassisType": "R620",
     "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
     "nodeMemoryGB": 32,
     "nodeType": "FC0025"
    },
    "powerSupplies": {
```

```
      "PS1 status": {
       "powerSupplyFailureDetected": false,
       "powerSupplyHasAC": true,
       "powerSupplyPredictiveFailureDetected": false,
       "powerSupplyPresent": true,
       "powerSupplyPresentLastCheck": true
      },
      "PS2 status": {
       "powerSupplyFailureDetected": false,
       "powerSupplyHasAC": true,
       "powerSupplyPredictiveFailureDetected": false,
       "powerSupplyPresent": true,
       "powerSupplyPresentLastCheck": true
      }
     },
     "storage": {
      "storage_PCI:0000:00:1f.2": {
       "businfo": "pci@0000:00:1f.2",
       "clock": "66000000",
       "description": "SATA controller",
       "physid": "1f.2",
       "product": "C600/X79 series chipset 6-Port SATA AHCI Controller",
       "vendor": "Intel Corporation",
       "version": "05",
       "width": "32"
      }
     },
     "system": {
      "ubuntu_DMI:0100": {
       "description": "Rack Mount Chassis",
       "product": "SFx010 ()",
       "serial": "HTW1DZ1",
       "vendor": "SolidFire",
       "width": "64"
      }
     },
     "temperatures": {
      "Exhaust Temp": {
       "baseUnit": "C",
       "threshold": 70,
       "value": 41
      },
      "Inlet Temp": {
       "baseUnit": "C",
       "threshold": 42,
       "value": 18
```

```
      }
      },
      "uuid": "4C4C4544-0054-5710-8031-C8C04F445A31"
    },
  "2": {...},                    Storage Node "2"
  "3": {...},                    Storage Node "3"
  "4": {...},                    Storage Node "4"
  "5": {                      Fibre Channel Node
       }
      }
    }
  }
```

GetLldpInfo

The `GetLldpInfo` method returns a response similar to the following example.

```
{
  "id": null,
  "result": {
    "lldpInfo": {
        "lldpChassis": {
        "local-chassis": [
        {
          "chassis": [
            {
              "capability": [
                {
                  "enabled": false,
                  "type": "Bridge"
                },
                {
                  "enabled": false,
                  "type": "Router"
                },
                {
                  "enabled": false,
                  "type": "Wlan"
                },
                {
                  "enabled": true,
                  "type": "Station"
                }
              ],
              "descr": [
```

```
            {
              "value": "Element OS 11.0"
            }
          ],
          "id": [
            {
              "type": "mac",
              "value": "08:00:27:3c:0a:f4"
            }
          ],
          "mgmt-ip": [
            {
              "value": "10.0.2.15"
            },
            {
              "value": "fe80::a00:27ff:fe3c:af4"
            }
          ],
          "name": [
            {
              "value": "SF-93FF"
            }
          ]
        }
      ],
      "lldp-med": [
        {
          "capability": [
            {
              "available": true,
              "type": "Capabilities"
            },
            {
              "available": true,
              "type": "Policy"
            },
            {
              "available": true,
              "type": "Location"
            },
            {
              "available": true,
              "type": "MDI/PSE"
            },
            {
              "available": true,
```

```
        "type": "MDI/PD"
      },
      {
        "available": true,
        "type": "Inventory"
      }
    ],
    "device-type": [
      {
        "value": "Generic Endpoint (Class I)"
      }
    ],
    "inventory": [
      {
        "firmware": [
          {
            "value": "VirtualBox"
          }
        ],
        "hardware": [
          {
            "value": "1.2"
          }
        ],
        "manufacturer": [
          {
            "value": "innotek GmbH"
          }
        ],
        "model": [
          {
            "value": "VirtualBox"
          }
        ],
        "serial": [
          {
            "value": "0"
          }
        ],
        "software": [
          {
            "value": "4.14.27-solidfire2"
          }
        ]
      }
    ]
```

```
            }
          ]
        }
      ]
    },
    "lldpInterfaces": {
      "lldp": [
        {
          "interface": [
            {
              "age": "0 day, 00:01:04",
              "chassis": [
                {
                  "capability": [
                    {
                      "enabled": false,
                      "type": "Bridge"
                    },
                    {
                      "enabled": false,
                      "type": "Router"
                    },
                    {
                      "enabled": false,
                      "type": "Wlan"
                    },
                    {
                      "enabled": true,
                      "type": "Station"
                    }
                  ],
                  "descr": [
                    {
                      "value": "Element OS 11.0"
                    }
                  ],
                  "id": [
                    {
                      "type": "mac",
                      "value": "08:00:27:3c:0a:f4"
                    }
                  ],
                  "mgmt-ip": [
                    {
                      "value": "10.0.2.15"
                    },
```

```
                {
                  "value": "fe80::a00:27ff:fe3c:af4"
                }
              ],
              "name": [
                {
                  "value": "SF-93FF"
                }
              ]
            }
          ],
          "lldp-med": [
            {
              "capability": [
                {
                  "available": true,
                  "type": "Capabilities"
                },
                {
                  "available": true,
                  "type": "Policy"
                },
                {
                  "available": true,
                  "type": "Location"
                },
                {
                  "available": true,
                  "type": "MDI/PSE"
                },
                {
                  "available": true,
                  "type": "MDI/PD"
                },
                {
                  "available": true,
                  "type": "Inventory"
                }
              ],
              "device-type": [
                {
                  "value": "Generic Endpoint (Class I)"
                }
              ],
              "inventory": [
                {
```

```
                "firmware": [
                  {
                    "value": "VirtualBox"
                  }
                ],
                "hardware": [
                  {
                    "value": "1.2"
                  }
                ],
                "manufacturer": [
                  {
                    "value": "innotek GmbH"
                  }
                ],
                "model": [
                  {
                    "value": "VirtualBox"
                  }
                ],
                "serial": [
                  {
                    "value": "0"
                  }
                ],
                "software": [
                  {
                    "value": "4.14.27-solidfire2"
                  }
                ]
              }
            ]
          }
        ],
        "name": "eth0",
        "port": [
          {
            "aggregation": [
              {
                "value": "7"
              }
            ],
            "auto-negotiation": [
              {
                "advertised": [
                  {
```

```
                    "fd": true,
                    "hd": true,
                    "type": "10Base-T"
                  },
                  {
                    "fd": true,
                    "hd": true,
                    "type": "100Base-TX"
                  },
                  {
                    "fd": true,
                    "hd": false,
                    "type": "1000Base-T"
                  }
                ],
                "current": [
                  {
                    "value": "full duplex mode"
                  }
                ],
                "enabled": true,
                "supported": true
              }
            ],
            "descr": [
              {
                "value": "eth0"
              }
            ],
            "id": [
              {
                "type": "mac",
                "value": "08:00:27:3c:0a:f4"
              }
            ]
          }
        ],
        "ttl": [
          {
            "ttl": "120"
          }
        ],
        "via": "unknown"
      },
      {
        "age": "17722 days, 17:14:28",
```

```
            "chassis": [
              {
                "capability": [
                  {
                    "enabled": false,
                    "type": "Bridge"
                  },
                  {
                    "enabled": false,
                    "type": "Router"
                  },
                  {
                    "enabled": false,
                    "type": "Wlan"
                  },
                  {
                    "enabled": true,
                    "type": "Station"
                  }
                ],
                "descr": [
                  {
                    "value": "Element OS 11.0"
                  }
                ],
                "id": [
                  {
                    "type": "mac",
                    "value": "08:00:27:3c:0a:f4"
                  }
                ],
                "mgmt-ip": [
                  {
                    "value": "10.0.2.15"
                  },
                  {
                    "value": "fe80::a00:27ff:fe3c:af4"
                  }
                ],
                "name": [
                  {
                    "value": "SF-93FF"
                  }
                ]
              }
            ],
```

```
            "lldp-med": [
              {
                "capability": [
                  {
                    "available": true,
                    "type": "Capabilities"
                  },
                  {
                    "available": true,
                    "type": "Policy"
                  },
                  {
                    "available": true,
                    "type": "Location"
                  },
                  {
                    "available": true,
                    "type": "MDI/PSE"
                  },
                  {
                    "available": true,
                    "type": "MDI/PD"
                  },
                  {
                    "available": true,
                    "type": "Inventory"
                  }
                ],
                "device-type": [
                  {
                    "value": "Generic Endpoint (Class I)"
                  }
                ],
                "inventory": [
                  {
                    "firmware": [
                      {
                        "value": "VirtualBox"
                      }
                    ],
                    "hardware": [
                      {
                        "value": "1.2"
                      }
                    ],
                    "manufacturer": [
```

```
                    {
                      "value": "innotek GmbH"
                    }
                  ],
                  "model": [
                    {
                      "value": "VirtualBox"
                    }
                  ],
                  "serial": [
                    {
                      "value": "0"
                    }
                  ],
                  "software": [
                    {
                      "value": "4.14.27-solidfire2"
                    }
                  ]
                }
              ]
            }
          ],
          "name": "eth1",
          "port": [
            {
              "aggregation": [
                {
                  "value": "7"
                }
              ],
              "auto-negotiation": [
                {
                  "advertised": [
                    {
                      "fd": true,
                      "hd": true,
                      "type": "10Base-T"
                    },
                    {
                      "fd": true,
                      "hd": true,
                      "type": "100Base-TX"
                    },
                    {
                      "fd": true,
```

```
                    "hd": false,
                    "type": "1000Base-T"
                  }
                ],
                "current": [
                  {
                    "value": "unknown"
                  }
                ],
                "enabled": true,
                "supported": true
              }
            ],
            "descr": [
              {
                "value": "eth1"
              }
            ],
            "id": [
              {
                "type": "mac",
                "value": "08:00:27:36:79:78"
              }
            ]
          }
        ],
        "ttl": [
          {
            "ttl": "120"
          }
        ],
        "via": "unknown"
      },
      {
        "age": "0 day, 00:01:01",
        "chassis": [
          {
            "capability": [
              {
                "enabled": false,
                "type": "Bridge"
              },
              {
                "enabled": false,
                "type": "Router"
              },
```

```
            {
              "enabled": false,
              "type": "Wlan"
            },
            {
              "enabled": true,
              "type": "Station"
            }
          ],
          "descr": [
            {
              "value": "Element OS 11.0"
            }
          ],
          "id": [
            {
              "type": "mac",
              "value": "08:00:27:3c:0a:f4"
            }
          ],
          "mgmt-ip": [
            {
              "value": "10.0.2.15"
            },
            {
              "value": "fe80::a00:27ff:fe3c:af4"
            }
          ],
          "name": [
            {
              "value": "SF-93FF"
            }
          ]
        }
      ],
      "lldp-med": [
        {
          "capability": [
            {
              "available": true,
              "type": "Capabilities"
            },
            {
              "available": true,
              "type": "Policy"
            },
```

```
      {
        "available": true,
        "type": "Location"
      },
      {
        "available": true,
        "type": "MDI/PSE"
      },
      {
        "available": true,
        "type": "MDI/PD"
      },
      {
        "available": true,
        "type": "Inventory"
      }
    ],
    "device-type": [
      {
        "value": "Generic Endpoint (Class I)"
      }
    ],
    "inventory": [
      {
        "firmware": [
          {
            "value": "VirtualBox"
          }
        ],
        "hardware": [
          {
            "value": "1.2"
          }
        ],
        "manufacturer": [
          {
            "value": "innotek GmbH"
          }
        ],
        "model": [
          {
            "value": "VirtualBox"
          }
        ],
        "serial": [
          {
```

```
                    "value": "0"
                  }
                ],
                "software": [
                  {
                    "value": "4.14.27-solidfire2"
                  }
                ]
              }
            ]
          }
        ],
        "name": "eth2",
        "port": [
          {
            "aggregation": [
              {
                "value": "6"
              }
            ],
            "auto-negotiation": [
              {
                "advertised": [
                  {
                    "fd": true,
                    "hd": true,
                    "type": "10Base-T"
                  },
                  {
                    "fd": true,
                    "hd": true,
                    "type": "100Base-TX"
                  },
                  {
                    "fd": true,
                    "hd": false,
                    "type": "1000Base-T"
                  }
                ],
                "current": [
                  {
                    "value": "full duplex mode"
                  }
                ],
                "enabled": true,
                "supported": true
```

```
            }
          ],
          "descr": [
            {
              "value": "eth2"
            }
          ],
          "id": [
            {
              "type": "mac",
              "value": "08:00:27:fc:f0:a9"
            }
          ]
        }
      ],
      "ttl": [
        {
          "ttl": "120"
        }
      ],
      "via": "LLDP"
    },
    {
      "age": "0 day, 00:01:01",
      "chassis": [
        {
          "capability": [
            {
              "enabled": false,
              "type": "Bridge"
            },
            {
              "enabled": false,
              "type": "Router"
            },
            {
              "enabled": false,
              "type": "Wlan"
            },
            {
              "enabled": true,
              "type": "Station"
            }
          ],
          "descr": [
            {
```

```
                "value": "Element OS 11.0"
              }
            ],
            "id": [
              {
                "type": "mac",
                "value": "08:00:27:3c:0a:f4"
              }
            ],
            "mgmt-ip": [
              {
                "value": "10.0.2.15"
              },
              {
                "value": "fe80::a00:27ff:fe3c:af4"
              }
            ],
            "name": [
              {
                "value": "SF-93FF"
              }
            ]
          }
        ],
        "lldp-med": [
          {
            "capability": [
              {
                "available": true,
                "type": "Capabilities"
              },
              {
                "available": true,
                "type": "Policy"
              },
              {
                "available": true,
                "type": "Location"
              },
              {
                "available": true,
                "type": "MDI/PSE"
              },
              {
                "available": true,
                "type": "MDI/PD"
```

```
      },
      {
        "available": true,
        "type": "Inventory"
      }
    ],
    "device-type": [
      {
        "value": "Generic Endpoint (Class I)"
      }
    ],
    "inventory": [
      {
        "firmware": [
          {
            "value": "VirtualBox"
          }
        ],
        "hardware": [
          {
            "value": "1.2"
          }
        ],
        "manufacturer": [
          {
            "value": "innotek GmbH"
          }
        ],
        "model": [
          {
            "value": "VirtualBox"
          }
        ],
        "serial": [
          {
            "value": "0"
          }
        ],
        "software": [
          {
            "value": "4.14.27-solidfire2"
          }
        ]
      }
    ]
  }
```

```
            ],
            "name": "eth3",
            "port": [
              {
                "aggregation": [
                  {
                    "value": "6"
                  }
                ],
                "auto-negotiation": [
                  {
                    "advertised": [
                      {
                        "fd": true,
                        "hd": true,
                        "type": "10Base-T"
                      },
                      {
                        "fd": true,
                        "hd": true,
                        "type": "100Base-TX"
                      },
                      {
                        "fd": true,
                        "hd": false,
                        "type": "1000Base-T"
                      }
                    ],
                    "current": [
                      {
                        "value": "full duplex mode"
                      }
                    ],
                    "enabled": true,
                    "supported": true
                  }
                ],
                "descr": [
                  {
                    "value": "eth3"
                  }
                ],
                "id": [
                  {
                    "type": "mac",
                    "value": "08:00:27:2c:e4:f8"
```

```
                    }
                  ]
                }
              ],
              "ttl": [
                {
                  "ttl": "120"
                }
              ],
              "via": "LLDP"
            }
          ]
        }
      ]
    },
    "lldpNeighbors": {
      "lldp": [
        {
          "interface": [
            {
              "age": "0 day, 00:04:34",
              "chassis": [
                {
                  "capability": [
                    {
                      "enabled": true,
                      "type": "Bridge"
                    },
                    {
                      "enabled": true,
                      "type": "Router"
                    },
                    {
                      "enabled": true,
                      "type": "Wlan"
                    },
                    {
                      "enabled": false,
                      "type": "Station"
                    }
                  ],
                  "descr": [
                    {
                      "value": "x86_64"
                    }
                  ],
```

```
            "id": [
              {
                "type": "mac",
                "value": "50:7b:9d:2b:36:84"
              }
            ],
            "mgmt-ip": [
              {
                "value": "192.168.100.1"
              },
              {
                "value": "fe80::a58e:843:952e:d8eb"
              }
            ],
            "name": [
              {
                "value": "ConventionalWisdom.wlan.netapp.com"
              }
            ]
          }
        ],
        "name": "eth2",
        "port": [
          {
            "auto-negotiation": [
              {
                "current": [
                  {
                    "value": "full duplex mode"
                  }
                ],
                "enabled": false,
                "supported": false
              }
            ],
            "descr": [
              {
                "value": "vboxnet1"
              }
            ],
            "id": [
              {
                "type": "mac",
                "value": "0a:00:27:00:00:01"
              }
            ],
```

```
            "ttl": [
              {
                "value": "120"
              }
            ]
          }
        ],
        "rid": "2",
        "via": "LLDP"
      },
      {
        "age": "0 day, 00:01:01",
        "chassis": [
          {
            "capability": [
              {
                "enabled": false,
                "type": "Bridge"
              },
              {
                "enabled": false,
                "type": "Router"
              },
              {
                "enabled": false,
                "type": "Wlan"
              },
              {
                "enabled": true,
                "type": "Station"
              }
            ],
            "descr": [
              {
                "value": "Element OS 11.0"
              }
            ],
            "id": [
              {
                "type": "mac",
                "value": "08:00:27:3c:0a:f4"
              }
            ],
            "mgmt-ip": [
              {
                "value": "10.0.2.15"
```

```
                    },
                    {
                      "value": "fe80::a00:27ff:fe3c:af4"
                    }
                  ],
                  "name": [
                    {
                      "value": "SF-93FF"
                    }
                  ]
                }
              ],
              "lldp-med": [
                {
                  "capability": [
                    {
                      "available": true,
                      "type": "Capabilities"
                    },
                    {
                      "available": true,
                      "type": "Policy"
                    },
                    {
                      "available": true,
                      "type": "Location"
                    },
                    {
                      "available": true,
                      "type": "MDI/PSE"
                    },
                    {
                      "available": true,
                      "type": "MDI/PD"
                    },
                    {
                      "available": true,
                      "type": "Inventory"
                    }
                  ],
                  "device-type": [
                    {
                      "value": "Generic Endpoint (Class I)"
                    }
                  ],
                  "inventory": [
```

```
                {
                  "firmware": [
                    {
                      "value": "VirtualBox"
                    }
                  ],
                  "hardware": [
                    {
                      "value": "1.2"
                    }
                  ],
                  "manufacturer": [
                    {
                      "value": "innotek GmbH"
                    }
                  ],
                  "model": [
                    {
                      "value": "VirtualBox"
                    }
                  ],
                  "serial": [
                    {
                      "value": "0"
                    }
                  ],
                  "software": [
                    {
                      "value": "4.14.27-solidfire2"
                    }
                  ]
                }
              ]
            }
          ],
          "name": "eth2",
          "port": [
            {
              "aggregation": [
                {
                  "value": "6"
                }
              ],
              "auto-negotiation": [
                {
                  "advertised": [
```

```
                    {
                      "fd": true,
                      "hd": true,
                      "type": "10Base-T"
                    },
                    {
                      "fd": true,
                      "hd": true,
                      "type": "100Base-TX"
                    },
                    {
                      "fd": true,
                      "hd": false,
                      "type": "1000Base-T"
                    }
                  ],
                  "current": [
                    {
                      "value": "full duplex mode"
                    }
                  ],
                  "enabled": true,
                  "supported": true
                }
              ],
              "descr": [
                {
                  "value": "eth3"
                }
              ],
              "id": [
                {
                  "type": "mac",
                  "value": "08:00:27:2c:e4:f8"
                }
              ],
              "ttl": [
                {
                  "value": "120"
                }
              ]
            }
          ],
          "rid": "1",
          "via": "LLDP"
        },
```

```
            {
              "age": "0 day, 00:04:34",
              "chassis": [
                {
                  "capability": [
                    {
                      "enabled": true,
                      "type": "Bridge"
                    },
                    {
                      "enabled": true,
                      "type": "Router"
                    },
                    {
                      "enabled": true,
                      "type": "Wlan"
                    },
                    {
                      "enabled": false,
                      "type": "Station"
                    }
                  ],
                  "descr": [
                    {
                      "value": "x86_64"
                    }
                  ],
                  "id": [
                    {
                      "type": "mac",
                      "value": "50:7b:9d:2b:36:84"
                    }
                  ],
                  "mgmt-ip": [
                    {
                      "value": "192.168.100.1"
                    },
                    {
                      "value": "fe80::a58e:843:952e:d8eb"
                    }
                  ],
                  "name": [
                    {
                      "value": ""
                    }
                  ]
```

```
      }
    ],
    "name": "eth3",
    "port": [
      {
        "auto-negotiation": [
          {
            "current": [
              {
                "value": "full duplex mode"
              }
            ],
            "enabled": false,
            "supported": false
          }
        ],
        "descr": [
          {
            "value": "vboxnet1"
          }
        ],
        "id": [
          {
            "type": "mac",
            "value": "0a:00:27:00:00:01"
          }
        ],
        "ttl": [
          {
            "value": "120"
          }
        ]
      }
    ],
    "rid": "2",
    "via": "LLDP"
  },
  {
    "age": "0 day, 00:01:01",
    "chassis": [
      {
        "capability": [
          {
            "enabled": false,
            "type": "Bridge"
          },
```

```
            {
              "enabled": false,
              "type": "Router"
            },
            {
              "enabled": false,
              "type": "Wlan"
            },
            {
              "enabled": true,
              "type": "Station"
            }
          ],
          "descr": [
            {
              "value": "Element OS 11.0"
            }
          ],
          "id": [
            {
              "type": "mac",
              "value": "08:00:27:3c:0a:f4"
            }
          ],
          "mgmt-ip": [
            {
              "value": "10.0.2.15"
            },
            {
              "value": "fe80::a00:27ff:fe3c:af4"
            }
          ],
          "name": [
            {
              "value": "SF-93FF"
            }
          ]
        }
      ],
      "lldp-med": [
        {
          "capability": [
            {
              "available": true,
              "type": "Capabilities"
            },
```

```
        {
          "available": true,
          "type": "Policy"
        },
        {
          "available": true,
          "type": "Location"
        },
        {
          "available": true,
          "type": "MDI/PSE"
        },
        {
          "available": true,
          "type": "MDI/PD"
        },
        {
          "available": true,
          "type": "Inventory"
        }
      ],
      "device-type": [
        {
          "value": "Generic Endpoint (Class I)"
        }
      ],
      "inventory": [
        {
          "firmware": [
            {
              "value": "VirtualBox"
            }
          ],
          "hardware": [
            {
              "value": "1.2"
            }
          ],
          "manufacturer": [
            {
              "value": "innotek GmbH"
            }
          ],
          "model": [
            {
              "value": "VirtualBox"
```

```
                    }
                  ],
                  "serial": [
                    {
                      "value": "0"
                    }
                  ],
                  "software": [
                    {
                      "value": "4.14.27-solidfire2"
                    }
                  ]
                }
              ]
            }
          ],
          "name": "eth3",
          "port": [
            {
              "aggregation": [
                {
                  "value": "6"
                }
              ],
              "auto-negotiation": [
                {
                  "advertised": [
                    {
                      "fd": true,
                      "hd": true,
                      "type": "10Base-T"
                    },
                    {
                      "fd": true,
                      "hd": true,
                      "type": "100Base-TX"
                    },
                    {
                      "fd": true,
                      "hd": false,
                      "type": "1000Base-T"
                    }
                  ],
                  "current": [
                    {
                      "value": "1000BaseTFD"
```

```
                        }
                      ],
                      "enabled": true,
                      "supported": true
                    }
                  ],
                  "descr": [
                    {
                      "value": "eth2"
                    }
                  ],
                  "id": [
                    {
                      "type": "mac",
                      "value": "08:00:27:fc:f0:a9"
                    }
                  ],
                  "ttl": [
                    {
                      "value": "120"
                    }
                  ]
                }
              ],
              "rid": "1",
              "via": "LLDP"
            }
          ]
        }
      ]
    }
    }
 }
}
```

GetNetworkConfig

The `GetNetworkConfig` method returns a response similar to the following example.

```
{
    "id": 1,
    "result": {
        "network": {
            "Bond10G": {
                "#default": false,
```

```
                "address": "10.1.1.0",
                "auto": true,
                "bond-downdelay": "0",
                "bond-fail_over_mac": "None",
                "bond-miimon": "100",
                "bond-mode": "ActivePassive",
                "bond-primary_reselect": "Failure",
                "bond-slaves": "eth0 eth1",
                "bond-updelay": "200",
                "dns-nameservers": "10.1.1.0, 10.1.1.0",
                "dns-search": "ten.test.company.net., company.net.",
                "family": "inet",
                "gateway": "10.1.1.0",
                "linkSpeed": 10000,
                "macAddress": "c8:1f:66:ee:59:b9",
                "macAddressPermanent": "00:00:00:00:00:00",
                "method": "static",
                "mtu": "9000",
                "netmask": "255.255.240.0",
                "network": "10.1.1.0",
                "physical": {
                    "address": "10.1.1.0",
                    "macAddress": "c8:1f:66:ee:59:b9",
                    "macAddressPermanent": "00:00:00:00:00:00",
                    "mtu": "9000",
                    "netmask": "255.255.240.0",
                    "network": "10.1.1.0",
                    "upAndRunning": true
                },
                "routes": [],
                "status": "UpAndRunning",
                "symmetricRouteRules": [
                    "ip route add 10.1.1.1/20 dev Bond1G src 10.1.2.2
table Bond1G",
                    "ip rule add from 10.1.1.1 table Bond1G",
                    "ip route add default via 10.1.1.254"
                ],
                "upAndRunning": true,
                "virtualNetworkTag": "0"
            },
            "Bond1G": {
                "#default": true,
                "address": "10.1.1.0",
                "addressV6": "",
                "auto": true,
                "bond-downdelay": "0",
```

```
                "bond-fail_over_mac": "None",
                "bond-miimon": "100",
                "bond-mode": "ActivePassive",
                "bond-primary_reselect": "Failure",
                "bond-slaves": "eth2 eth3",
                "bond-updelay": "200",
                "dns-nameservers": "10.1.1.0, 10.1.1.0",
                "dns-search": "ten.test.company.net., company.net.",
                "family": "inet",
                "gateway": "10.1.1.254",
                "gatewayV6": "",
                "linkSpeed": 1000,
                "macAddress": "c8:1f:66:ee:59:bd",
                "macAddressPermanent": "00:00:00:00:00:00",
                "method": "static",
                "mtu": "1500",
                "netmask": "255.255.240.0",
                "network": "10.1.1.0",
                "physical": {
                    "address": "10.1.1.0",
                    "macAddress": "c8:1f:66:ee:59:bd",
                    "macAddressPermanent": "00:00:00:00:00:00",
                    "mtu": "1500",
                    "netmask": "255.255.240.0",
                    "network": "10.1.1.0",
                    "upAndRunning": true
                },
                "routes": [],
                "status": "UpAndRunning",
                "symmetricRouteRules": [
                    "ip route add 10.1.1.1/20 dev Bond1G src 10.1.2.2
table Bond1G",
                    "ip rule add from 10.1.1.1 table Bond1G",
                    "ip route add default via 10.1.1.254"
                ],
                "upAndRunning": true,
                "virtualNetworkTag": "0"
            },
            "eth0": {
                "auto": true,
                "bond-master": "Bond10G",
                "family": "inet",
                "linkSpeed": 10000,
                "macAddress": "c8:1f:66:ee:59:b9",
                "macAddressPermanent": "c8:1f:66:ee:59:b9",
                "method": "bond",
```

```
                "physical": {
                    "address": "0.0.0.0",
                    "macAddress": "c8:1f:66:ee:59:b9",
                    "macAddressPermanent": "c8:1f:66:ee:59:b9",
                    "netmask": "N/A",
                    "network": "N/A",
                    "upAndRunning": true
                },
                "status": "UpAndRunning",
                "upAndRunning": true
            },
            "eth1": {
                "auto": true,
                "bond-master": "Bond10G",
                "family": "inet",
                "linkSpeed": 10000,
                "macAddress": "c8:1f:66:ee:59:b9",
                "macAddressPermanent": "c8:1f:66:ee:59:bb",
                "method": "bond",
                "physical": {
                    "address": "0.0.0.0",
                    "macAddress": "c8:1f:66:ee:59:b9",
                    "macAddressPermanent": "c8:1f:66:ee:59:bb",
                    "netmask": "N/A",
                    "network": "N/A",
                    "upAndRunning": true
                },
                "status": "UpAndRunning",
                "upAndRunning": true
            },
            "eth2": {
                "auto": true,
                "bond-master": "Bond1G",
                "family": "inet",
                "linkSpeed": 1000,
                "macAddress": "c8:1f:66:ee:59:bd",
                "macAddressPermanent": "c8:1f:66:ee:59:bd",
                "method": "bond",
                "physical": {
                    "address": "0.0.0.0",
                    "macAddress": "c8:1f:66:ee:59:bd",
                    "macAddressPermanent": "c8:1f:66:ee:59:bd",
                    "netmask": "N/A",
                    "network": "N/A",
                    "upAndRunning": true
                },
```

```
                "status": "UpAndRunning",
                "upAndRunning": true
            },
            "eth3": {
                "auto": true,
                "bond-master": "Bond1G",
                "family": "inet",
                "linkSpeed": 1000,
                "macAddress": "c8:1f:66:ee:59:bd",
                "macAddressPermanent": "c8:1f:66:ee:59:bf",
                "method": "bond",
                "physical": {
                    "address": "0.0.0.0",
                    "macAddress": "c8:1f:66:ee:59:bd",
                    "macAddressPermanent": "c8:1f:66:ee:59:bf",
                    "netmask": "N/A",
                    "network": "N/A",
                    "upAndRunning": true
                },
                "status": "UpAndRunning",
                "upAndRunning": true
            },
            "lo": {
                "auto": true,
                "family": "inet",
                "linkSpeed": 0,
                "macAddress": "00:00:00:00:00:00",
                "macAddressPermanent": "00:00:00:00:00:00",
                "method": "loopback",
                "physical": {
                    "address": "0.0.0.0",
                    "macAddress": "00:00:00:00:00:00",
                    "macAddressPermanent": "00:00:00:00:00:00",
                    "netmask": "N/A",
                    "network": "N/A",
                    "upAndRunning": true
                },
                "status": "UpAndRunning",
                "upAndRunning": true
            }
        }
    }
}
```

GetNodeHardwareInfo (output for iSCSI)

The `GetNodeHardwareInfo` method for iSCSI returns a response similar to the following example.


```
{
    "id": 1,
    "result": {
        "nodeHardwareInfo": {
            "bus": {
                "core_DMI:0200": {
                    "description": "Motherboard",
                    "physid": "0",
                    "product": "0H47HH",
                    "serial": "..CN7475141I0271.",
                    "vendor": "SolidFire",
                    "version": "A07"
                }
            },
            "driveHardware": [
                {
                "canonicalName": "sda",
                "connected": true,
                "dev": 2048,
                "devPath": "/dev/slot0",
                "driveEncryptionCapability": "fips",
                "driveType": "Slice",
                "lifeRemainingPercent": 98,
                "lifetimeReadBytes": 0,
                "lifetimeWriteBytes": 14012129342144,
                "name": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
                "path": "/dev/sda",
                "pathLink": "/dev/slot0",
                "powerOnHours": 15489,
                "product": "SAMSUNG MZ7GE240HMGR-00003",
                "reallocatedSectors": 0,
                "reserveCapacityPercent": 100,
                "scsiCompatId": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
                "scsiState": "Running",
                "securityAtMaximum": false,
                "securityEnabled": true,
                "securityFrozen": false,
                "securityLocked": false,
                "securitySupported": true,
                "serial": "S1M9NWAG501251",
                "size": 240057409536,
```

```
                "slot": 0,
                "uncorrectableErrors": 0,
                "uuid": "789aa05d-e49b-ff4f-f821-f60eed8e43bd",
                "vendor": "Samsung",
                "version": "EXT1303Q"
                },
                {
                "canonicalName": "sda",
                "connected": true,
                "dev": 2048,
                "devPath": "/dev/slot1",
                "driveEncryptionCapability": "fips",
                "driveType": "Slice",
                "lifeRemainingPercent": 98,
                "lifetimeReadBytes": 0,
                "lifetimeWriteBytes": 14112129567184,
                "name": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
                "path": "/dev/sda",
                "pathLink": "/dev/slot0",
                "powerOnHours": 15489,
                "product": "SAMSUNG MZ7GE240HMGR-00003",
                "reallocatedSectors": 0,
                "reserveCapacityPercent": 100,
                "scsiCompatId": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
                "scsiState": "Running",
                "securityAtMaximum": false,
                "securityEnabled": true,
                "securityFrozen": false,
                "securityLocked": false,
                "securitySupported": true,
                "serial": "S1M9NWAG501252",
                "size": 240057409536,
                "slot": 0,
                "uncorrectableErrors": 0,
                "uuid": "789aa05d-e49b-ff4f-f821-f60eed8e43bd",
                "vendor": "Samsung",
                "version": "EXT1303Q"
                }
    }
```


GetNodeHardwareInfo (output for Fibre Channel nodes)

The `GetNodeHardwareInfo` method for Fibre Channel nodes returns a response similar to the following example.


```
```

```
{
"id": null,
"result": {
"nodeHardwareInfo": {
"bus": {
"core_DMI:0200": {
"description": "Motherboard",
"physid": "0",
"product": "0H47HH",
"serial": "..CN747513AA0541.",
"version": "A07"
},
"fiber:0_PCI:0000:04:00.0": {
"businfo": "pci@0000:04:00.0",
"clock": "33000000",
"description": "Fibre Channel",
"physid": "0",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
},
"fiber:0_PCI:0000:42:00.0": {
"businfo": "pci@0000:42:00.0",
"clock": "33000000",
"description": "Fibre Channel",
"physid": "0",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
},
"fiber:1_PCI:0000:04:00.1": {
"businfo": "pci@0000:04:00.1",
"clock": "33000000",
"description": "Fibre Channel",
"physid": "0.1",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
},
"fiber:1_PCI:0000:42:00.1": {
"businfo": "pci@0000:42:00.1",
"clock": "33000000",
"description": "Fibre Channel",
```

```
"physid": "0.1",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
}
},
"fans": {
"Fan1A RPM": {
"baseUnit": "RPM",
"threshold": 840,
"value": 3360
},
"Fan1B RPM": {
"baseUnit": "RPM",
"threshold": 840,
"value": 3120
}
},
"fibreChannelPorts": [
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 1,
"internalPortID": 2,
"model": "QLE2672",
"nPortID": "0x060019",
"nodeID": 6,
"pciSlot": 3,
"serial": "BFE1335E04217",
"speed": "8 Gbit",
"state": "Online",
"switchWwn": "20:01:00:2a:6a:9c:71:01",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:0a"
},
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 2,
"internalPortID": 3,
"model": "QLE2672",
"nPortID": "0xc70019",
"nodeID": 6,
"pciSlot": 3,
"serial": "BFE1335E04217",
"speed": "8 Gbit",
"state": "Online",
```

```
"switchWwn": "20:01:00:2a:6a:98:a3:41",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:0b"
},
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 1,
"internalPortID": 0,
"model": "QLE2672",
"nPortID": "0xc70017",
"nodeID": 6,
"pciSlot": 2,
"serial": "BFE1341E09515",
"speed": "8 Gbit",
"state": "Online",
"switchWwn": "20:01:00:2a:6a:98:a3:41",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:08"
},
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 2,
"internalPortID": 1,
"model": "QLE2672",
"nPortID": "0x060017",
"nodeID": 6,
"pciSlot": 2,
"serial": "BFE1341E09515",
"speed": "8 Gbit",
"state": "Online",
"switchWwn": "20:01:00:2a:6a:9c:71:01",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:09"
}
],
"memory": {
"firmware_": {
"capacity": "8323072",
"date": "08/29/2013",
"description": "BIOS",
"physid": "0",
"size": "65536",
"version": "2.0.19"
},
"memory_DMI:1000": {
"description": "System Memory",
```

```
"physid": "1000",
"size": "34359738368",
"slot": "System board or motherboard"
}
},
"network": {
"network:0_": {
"description": "Ethernet interface",
"logicalname": "Bond1G",
"physid": "1",
"serial": "c8:1f:66:df:04:da"
},
"network:0_PCI:0000:01:00.0": {
"businfo": "pci@0000:01:00.0",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth0",
"physid": "0",
"product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
"serial": "c8:1f:66:df:04:d6",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
},
"network:0_PCI:0000:41:00.0": {
"businfo": "pci@0000:41:00.0",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth4",
"physid": "0",
"product": "NetXtreme II BCM57810 10 Gigabit Ethernet",
"serial": "00:0a:f7:41:7a:30",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
},
"network:1_": {
"description": "Ethernet interface",
"logicalname": "Bond10G",
"physid": "2",
"serial": "c8:1f:66:df:04:d6"
},
"network:1_PCI:0000:01:00.1": {
"businfo": "pci@0000:01:00.1",
```

```
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth1",
"physid": "0.1",
"product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
"serial": "c8:1f:66:df:04:d8",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
},
"network:1_PCI:0000:41:00.1": {
"businfo": "pci@0000:41:00.1",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth5",
"physid": "0.1",
"product": "NetXtreme II BCM57810 10 Gigabit Ethernet",
"serial": "00:0a:f7:41:7a:32",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
},
"network:2_PCI:0000:01:00.2": {
"businfo": "pci@0000:01:00.2",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth2",
"physid": "0.2",
"product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
"serial": "c8:1f:66:df:04:da",
"size": "1000000000",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
},
"network:3_PCI:0000:01:00.3": {
"businfo": "pci@0000:01:00.3",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth3",
"physid": "0.3",
"product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
```

```
"serial": "c8:1f:66:df:04:dc",
"size": "1000000000",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
}
},
"networkInterfaces": {
"Bond10G": {
"isConfigured": true,
"isUp": true
},
"Bond1G": {
"isConfigured": true,
"isUp": true
},
"eth0": {
"isConfigured": true,
"isUp": true
},
"eth1": {
"isConfigured": true,
"isUp": true
},
"eth2": {
"isConfigured": true,
"isUp": true
},
"eth3": {
"isConfigured": true,
"isUp": true
},
"eth4": {
"isConfigured": true,
"isUp": true
},
"eth5": {
"isConfigured": true,
"isUp": true
}
},
"platform": {
"chassisType": "R620",
"cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
"nodeMemoryGB": 32,
"nodeType": "SFFC"
```

```
},
"powerSupplies": {
"PS1 status": {
"powerSupplyFailureDetected": false,
"powerSupplyHasAC": true,
"powerSupplyPredictiveFailureDetected": false,
"powerSupplyPresent": true
},
"PS2 status": {
"powerSupplyFailureDetected": false,
"powerSupplyHasAC": true,
"powerSupplyPredictiveFailureDetected": false,
"powerSupplyPresent": true
}
},
"storage": {
"storage_PCI:0000:00:1f.2": {
"businfo": "pci@0000:00:1f.2",
"clock": "66000000",
"description": "SATA controller",
"physid": "1f.2",
"product": "C600/X79 series chipset 6-Port SATA AHCI Controller",
"vendor": "Intel Corporation",
"version": "05",
"width": "32"
}
},
"system": {
"fcn-2_DMI:0100": {
"description": "Rack Mount Chassis",
"product": "(SKU=NotProvided;ModelName=)",
"serial": "HTX1DZ1",
"width": "64"
}
},
"temperatures": {
"Exhaust Temp": {
"baseUnit": "C",
"threshold": 70,
"value": 38
},
"Inlet Temp": {
"baseUnit": "C",
"threshold": 42,
"value": 13
  },
```

```
"uuid": "4C4C4544-004D-5310-8052-C4C04F335431"
    }
   }
  }
 }
}
```

GetNvramInfo

The `GetNvramInfo` method returns a response similar to the following example.

```
{
  id: 1,
  result: {
    nvramInfo: {
      details: {
         errors: {
           numOfErrorLogEntries: "0"
         },
         extended: {
            dialogVersion: "4",
            event: [
               {
                    name: "flushToFlash",
                    time: "2014-02-24 20:30:28",
                    value: "0"
         },
         {
                    name: "flushToFlash",
                    time: "1946-02-06 17:16:42",
                    value: "0"
         },
         {
                    name: "flushToFlash",
                    time: "2014-02-25 00:48:06",
                    value: "0"
         },
         {
                    name: "flushToFlash",
                    time: "2014-02-25 15:44:07",
                    value: "0"
         },
         {
                    name: "flushToFlash",
                    time: "2014-03-17 17:21:46",
```

```
                value: "0"
        },
        {
                name: "flushToFlash",
                time: "2014-03-17 17:59:30",
                value: "0"
        },
        {
                name: "flushToFlash",
                time: "2014-03-17 18:06:27",
                value: "0"
        },
        {
                name: "flushToFlash",
                time: "2014-03-17 21:43:17",
                value: "0"
        },
        {
                name: "excessiveCurrent",
                time: "2014-02-25 00:00:29",
                value: "39"
        },
         {
                name: "excessiveCurrent",
                time: "2014-03-01 00:00:24",
                value: "23"
         }
      ],
          eventOccurrences: [
        {
                count: "15",
                name: "flushToFlash"
        },
        {
                count: "2",
                name: "excessiveCurrent"
        }
                initialCapacitance: "6.653 F",
                initialEsr: "0.097 Ohm",
          measurement: [
        {
                level_0: " 0",
                level_1: " 112",
                level_2: " 670919",
                level_3: " 455356",
                level_4: " 90215",
```

```
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "enterpriseFlashControllerTemperature",
                recent: "64 C"
        },
        {
                level_0: " 0",
                level_1: " 27",
                level_2: " 456896",
                level_3: " 717565",
                level_4: " 39422",
                level_5: " 2692",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "capacitor1And2Temperature",
                recent: "28.64 C"
        },
        {
                level_0: " 0",
                level_1: " 2080",
                level_2: " 907196",
                level_3: " 280178",
                level_4: " 26539",
                level_5: " 609",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "capacitor3And4Temperature",
                recent: "28.60 C"
        },
        {
                errorPeriod: {
                    duration: "24",
                    startTime: "2014-02-06 00:23:54",
                    worst: "8"
                },
                level_0: " 0",
                level_1: " 839",
                level_2: " 272794",
                level_3: " 404758",
```

```
                level_4: " 35216",
                level_5: " 377818",
                level_6: " 103891",
                level_7: " 21274",
                level_8: " 12",
                level_9: " 0",
                name: "rearVentAmbientTemperature",
                recent: "46.82 C"
        },
        {
                level_0: " 0",
                level_1: " 742749",
                level_2: " 460016",
                level_3: " 13837",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "rms200BoardTemperature",
                recent: "50.62 C"
        },
        {
                name: "voltageOfCapacitor1",
                recent: "2.308 V"
        },
        {
                name: "voltageOfCapacitor2",
                recent: "2.305 V"},
        {
                name: "voltageOfCapacitor3",
                recent: "2.314 V"
        },
        {
                name: "voltageOfCapacitor4",
                recent: "2.307 V"
        },
        {
               level_0: " 175052",
               level_1: " 51173",
               level_2: " 435788",
               level_3: " 12766",
               level_4: " 4",
               level_5: " 6",
               level_6: " 541813",
```

```
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "capacitorPackVoltage",
                recent: "9.233 V"
         },
         {
                level_0: " 0",
                level_1: " 0",
                level_2: " 0",
                level_3: " 0",
                level_4: " 0",
                level_5: " 0",
                level_6: " 4",
                level_7: " 1",
                level_8: " 4",
                level_9: " 6",
                name: "capacitorPackVoltageAtEndOfFlushToFlash",
                recent: "5.605 V"
        },
        {
                name: "currentDerivedFromV3V4",
                recent: "0.000 A"
        },
        {
                level_0: " 7",
                level_1: " 4",
                level_2: " 3",
                level_3: " 1",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "derivedEnergy",
                recent: "175 Joules"
        },
        {
                level_0: " 0",
                level_1: " 0",
                level_2: " 0",
                level_3: " 0",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
```

```
                level_7: " 17",
                level_8: " 19",
                level_9: " 7",
                name: "derivedCapacitanceOfThePack",
                recent: "5.959 F"
         },
         {
                level_0: " 0",
                level_1: " 43",
                level_2: " 0",
                level_3: " 0",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "derivedEsrOfCapacitorPack",
                recent: "0.104 Ohm"
        },
        {
                level_0: " 0",
                level_1: " 0",
                level_2: " 0",
                level_3: " 0",
                level_4: " 15",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "timeToRunFlushToFlash",
                recent: "22.40 Seconds"
        },
        {
                level_0: " 0",
                level_1: " 0",
                level_2: " 7",
                level_3: " 0",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "timeToRunRestore",
```

```
                recent: "20.44 Seconds"
        },
        {
                level_0: " 0",
                level_1: " 1",
                level_2: " 3",
                level_3: " 2",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 1",
                name: "timeToChargeCapacitors",
                recent: "48 Seconds"
        },
        {
                level_0: " 448586",
                level_1: " 2998",
                level_2: " 0",
                level_3: " 0",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name: "correctableBitsInErrorOnReadingAPage"
        },
        {
                level_0: " 2998",
                level_1: " 0",
                level_2: " 0",
                level_3: " 0",
                level_4: " 0",
                level_5: " 0",
                level_6: " 0",
                level_7: " 0",
                level_8: " 0",
                level_9: " 0",
                name:
"correctableBitsInErrorOnReadingTheWorstBchRegionOfAPage"
        },
        {
                level_0: " 0",
                level_1: " 37",
```

```
                level_2: " 280274",
                level_3: " 422999",
                level_4: " 245814",
                level_5: " 242470",
                level_6: " 24447",
                level_7: " 561",
                level_8: " 0",
                level_9: " 0",
                name: "fanInletAmbientTemperature",
                recent: "41.74 C"
        }
        ],
                predictedCapacitanceDepletion: "504328 uF",
                smartCounters: [
        {
                name: "numberOf512ByteBlocksReadFromDdr",
                value: "218284648"
        },
        {
                name: "numberOf512ByteBlocksWrittenToDdr",
                value: "12031567354"
        },
        {
                name: "numberOfHostReadCommands",
                value: "5366315"
        },
        {
                name: "numberOfHostWriteCommands",
                value: "1266099334"
        },
        {
                name: "controllerBusyTimeMinutes",
                value: "0"
        },
        {
                name: "numberOfPowerCycles",
                value: "13"
        },
        {
                name: "powerOnHours",
                value: "1009"
        },
        {
                name: "unsafeShutdowns",
                value: "5"
        },
```

```
        {
                name: "mediaErrors",
                value: "0"
        },
        {
                name: "numberOfErrorLogs",
                value: "2"
        }
        ],
         snapshotTime: "2014-03-20 16:43:49"
     },
     firmware: {
         activeSlotNumber: "2",
         slot1Version: "1e5817bc",
         slot2Version: "1e0d70ac",
         slot3Version: "1e5817bc",
         slot4Version: "1e5817bc"
    },
     smart: {
         availableSpace: "0%",
         availableSpaceThreshold: "0%",
         controllerBusyTimeMinutes: "0",
         criticalErrorVector: "0x0",
         mediaErrors: "0",
         numberOf512ByteBlocksRead: "218284648",
         numberOf512ByteBlocksWritten: "12031567354",
         numberOfErrorInfoLogs: "2",
         numberOfHostReadCommands: "5366315",
         numberOfHostWriteCommands: "1266099334",
         numberOfPowerCycles: "13",
         powerOnHours: "1009",
         temperature: "323 Kelvin",
         unsafeShutdowns: "5"
      }
     },
     status: "Warning",
     statusInfo: {
     warning: [
         "excessiveCurrent (2x)"
       ]
     },
     type: "RMS-200"
    }
  }
}
```

ListActiveNodes

The `ListActiveNodes` method returns a response similar to the following example.

```
 {
    "id": 1,
    "result": {
        "nodes": [
            {
                "associatedFServiceID": 0,
                "associatedMasterServiceID": 1,
                "attributes": {},
                "cip": "172.27.21.23",
                "cipi": "Bond10G",
                "fibreChannelTargetPortGroup": null,
                "mip": "172.27.1.23",
                "mipi": "Bond1G",
                "name": "PSN-1-23",
                "nodeID": 1,
                "platformInfo": {
                    "chassisType": "R620",
                    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @
2.50GHz",
                    "nodeMemoryGB": 72,
                    "nodeType": "SF3010"
                },
                "sip": "172.27.21.23",
                "sipi": "Bond10G",
                "softwareVersion": "9.0.0.1298",
                "uuid": "4C4C4544-0056-3810-804E-B5C04F4C5631",
                "virtualNetworks": [
                    {
                        "address": "10.1.2.4",
                        "virtualNetworkID": 1
                    },
                    {
                        "address": "10.2.2.10",
                        "virtualNetworkID": 2
                    }
                ]
            },
            {
                "associatedFServiceID": 0,
                "associatedMasterServiceID": 4,
                "attributes": {},
                "cip": "172.27.21.24",
```

```
                "cipi": "Bond10G",
                "fibreChannelTargetPortGroup": null,
                "mip": "172.27.1.24",
                "mipi": "Bond1G",
                "name": "PSN-1-24",
                "nodeID": 2,
                "platformInfo": {
                    "chassisType": "R620",
                    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @
2.50GHz",
                    "nodeMemoryGB": 72,
                    "nodeType": "SF3010"
                },
                "sip": "172.27.21.24",
                "sipi": "Bond10G",
                "softwareVersion": "9.0.0.1298",
                "uuid": "4C4C4544-0042-4210-804E-C3C04F4C5631",
                "virtualNetworks": [
                    {
                        "address": "10.1.2.5",
                        "virtualNetworkID": 1
                    },
                    {
                        "address": "10.2.2.11",
                        "virtualNetworkID": 2
                    }
                ]
            },
            {
                "associatedFServiceID": 0,
                "associatedMasterServiceID": 2,
                "attributes": {},
                "cip": "172.27.21.25",
                "cipi": "Bond10G",
                "fibreChannelTargetPortGroup": null,
                "mip": "172.27.1.25",
                "mipi": "Bond1G",
                "name": "PSN-1-25",
                "nodeID": 3,
                "platformInfo": {
                    "chassisType": "R620",
                    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @
2.50GHz",
                    "nodeMemoryGB": 72,
                    "nodeType": "SF3010"
                },
```

```
                "sip": "172.27.21.25",
                "sipi": "Bond10G",
                "softwareVersion": "9.0.0.1298",
                "uuid": "4C4C4544-0053-4210-8051-C6C04F515631",
                "virtualNetworks": [
                    {
                        "address": "10.1.2.6",
                        "virtualNetworkID": 1
                    },
                    {
                        "address": "10.2.2.12",
                        "virtualNetworkID": 2
                    }
                ]
            },
            {
                "associatedFServiceID": 0,
                "associatedMasterServiceID": 3,
                "attributes": {},
                "cip": "172.27.21.26",
                "cipi": "Bond10G",
                "fibreChannelTargetPortGroup": null,
                "mip": "172.27.1.26",
                "mipi": "Bond1G",
                "name": "PSN-1-26",
                "nodeID": 4,
                "platformInfo": {
                    "chassisType": "R620",
                    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @
2.50GHz",
                    "nodeMemoryGB": 72,
                    "nodeType": "SF3010"
                },
                "sip": "172.27.21.26",
                "sipi": "Bond10G",
                "softwareVersion": "9.0.0.1298",
                "uuid": "4C4C4544-0056-3810-804E-B4C04F4C5631",
                "virtualNetworks": [
                    {
                        "address": "10.1.2.7",
                        "virtualNetworkID": 1
                    },
                    {
                        "address": "10.2.2.13",
                        "virtualNetworkID": 2
                    }
```

```
                ]
            }
        ]
    }
 }
```

ListActiveVolumes

The `ListActiveVolumes` method returns a response similar to the following example.

```
{
    "id": 1,
    "result": {
        "volumes": [
            {
                "access": "readWrite",
                "accountID": 1,
                "attributes": {},
                "blockSize": 4096,
                "createTime": "2016-06-23T14:19:12Z",
                "deleteTime": "",
                "enable512e": false,
                "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo1.1",
                "name": "HulkDemo1",
                "purgeTime": "",
                "qos": {
                    "burstIOPS": 1500,
                    "burstTime": 60,
                    "curve": {
                        "4096": 100,
                        "8192": 160,
                        "16384": 270,
                        "32768": 500,
                        "65536": 1000,
                        "131072": 1950,
                        "262144": 3900,
                        "524288": 7600,
                        "1048576": 15000
                    },
                    "maxIOPS": 1000,
                    "minIOPS": 100
                },
                "scsiEUIDeviceID": "306f746f00000001f47acc0100000000",
                "scsiNAADeviceID": "6f47acc100000000306f746f00000001",
                "sliceCount": 1,
```

```
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 1,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:14Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo6.6",
            "name": "HulkDemo6",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
            },
            "scsiEUIDeviceID": "306f746f00000006f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000006",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
```

```
            "volumeID": 6,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:14Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo7.7",
            "name": "HulkDemo7",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
            },
            "scsiEUIDeviceID": "306f746f00000007f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000007",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 7,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
```

```
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:15Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo8.8",
            "name": "HulkDemo8",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
            },
            "scsiEUIDeviceID": "306f746f00000008f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000008",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 8,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:15Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo9.9",
```

```
            "name": "HulkDemo9",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
            },
            "scsiEUIDeviceID": "306f746f00000009f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000009",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 9,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:16Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo12.12",
            "name": "HulkDemo12",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
```

```
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
            },
            "scsiEUIDeviceID": "306f746f0000000cf47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f0000000c",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 12,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:18Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo16.16",
            "name": "HulkDemo16",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
```

```
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
            },
            "scsiEUIDeviceID": "306f746f00000010f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000010",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 16,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:18Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo17.17",
            "name": "HulkDemo17",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
```

```
            },
            "scsiEUIDeviceID": "306f746f00000011f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000011",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 17,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-23T14:19:18Z",
            "deleteTime": "",
            "enable512e": false,
            "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo18.18",
            "name": "HulkDemo18",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 1500,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 1000,
                "minIOPS": 100
            },
            "scsiEUIDeviceID": "306f746f00000012f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000012",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 53687091200,
```

```
            "virtualVolumeID": null,
            "volumeAccessGroups": [
                1
            ],
            "volumeID": 18,
            "volumePairs": []
        },
        {
            "access": "readWrite",
            "accountID": 1,
            "attributes": {},
            "blockSize": 4096,
            "createTime": "2016-06-24T15:21:59Z",
            "deleteTime": "",
            "enable512e": true,
            "iqn": "iqn.2010-01.com.solidfire:0oto.bk.24",
            "name": "BK",
            "purgeTime": "",
            "qos": {
                "burstIOPS": 15000,
                "burstTime": 60,
                "curve": {
                    "4096": 100,
                    "8192": 160,
                    "16384": 270,
                    "32768": 500,
                    "65536": 1000,
                    "131072": 1950,
                    "262144": 3900,
                    "524288": 7600,
                    "1048576": 15000
                },
                "maxIOPS": 15000,
                "minIOPS": 50
            },
            "scsiEUIDeviceID": "306f746f00000018f47acc0100000000",
            "scsiNAADeviceID": "6f47acc100000000306f746f00000018",
            "sliceCount": 1,
            "status": "active",
            "totalSize": 10737418240,
            "virtualVolumeID": null,
            "volumeAccessGroups": [],
            "volumeID": 24,
            "volumePairs": [
                {
                    "clusterPairID": 2,
```

```
                        "remoteReplication": {
                            "mode": "Async",
                            "pauseLimit": 3145728000,
                            "remoteServiceID": 14,
                            "resumeDetails": "",
                            "snapshotReplication": {
                                "state": "Idle",
                                "stateDetails": ""
                            },
                            "state": "Active",
                            "stateDetails": ""
                        },
                        "remoteSliceID": 8,
                        "remoteVolumeID": 8,
                        "remoteVolumeName": "PairingDoc",
                        "volumePairUUID": "229fcbf3-2d35-4625-865a-
d04bb9455cef"
                    }
                ]
            }
        ]
    }
}
```

TestHardwareConfig

The `TestHardwareConfig` method returns a response similar to the following example.

```
{
"id": 1,
"result": {
    "nodes": [
        {
            "nodeID": 1,
            "result": {
                "details": {
                    "BIOS_REVISION": {
                        "Passed": true,
                        "actual": "2.0",
                        "comparator": ">=",
                        "expected": "1.0.0.0"
                    },
                    "BIOS_VENDOR": {
                        "Passed": true,
                        "actual": "SolidFire",
```

```
                        "comparator": "==",
                        "expected": "SolidFire"
                    },
                    "BIOS_VERSION": {
                        "Passed": true,
                        "actual": "2.0.19",
                        "comparator": ">=",
                        "expected": "2.0.19"
                    },
                    "CPU_CORES_00": {
                        "Passed": true,
                        "actual": "6",
                        "comparator": "==",
                        "expected": "6"
                    },
                    "CPU_CORES_01": {
                        "Passed": true,
                        "actual": "6",
                        "comparator": "==",
                        "expected": "6"
                    },
                    "CPU_CORES_ENABLED_00": {
                        "Passed": true,
                        "actual": "6",
                        "comparator": "==",
                        "expected": "6"
                    },
                    "CPU_CORES_ENABLED_01": {
                        "Passed": true,
                        "actual": "6",
                        "comparator": "==",
                        "expected": "6"
                    },
                    "CPU_MODEL_00": {
                        "Passed": true,
                        "actual": "Intel(R) Xeon(R) CPU E5-2620 v2 @
2.10GHz",
                        "comparator": "==",
                        "expected": "Intel(R) Xeon(R) CPU E5-2620 v2 @
2.10GHz"
                    },
                    "CPU_MODEL_01": {
                        "Passed": true,
                        "actual": "Intel(R) Xeon(R) CPU E5-2620 v2 @
2.10GHz",
                        "comparator": "==",
```

```
                        "expected": "Intel(R) Xeon(R) CPU E5-2620 v2 @
2.10GHz"
                    },
                    "CPU_THREADS_00": {
                        "Passed": true,
                        "actual": "12",
                        "comparator": "==",
                        "expected": "12"
                    },
                    "CPU_THREADS_01": {
                        "Passed": true,
                        "actual": "12",
                        "comparator": "==",
                        "expected": "12"
                    },
                    "CPU_THREADS_ENABLED": {
                        "Passed": true,
                        "actual": "24",
                        "comparator": "==",
                        "expected": "24"
                    },
                    "IDRAC_VERSION": {
                        "Passed": true,
                        "actual": "2.41.40.40",
                        "comparator": ">=",
                        "expected": "1.06.06"
                    },
                    "MEMORY_GB": {
                        "Passed": true,
                        "actual": "64",
                        "comparator": ">=",
                        "expected": "64"
                    },
                    "MEMORY_MHZ_00": {
                        "Passed": true,
                        "actual": "1600",
                        "comparator": ">=",
                        "expected": "1333"
                    },
                    "MEMORY_MHZ_01": {
                        "Passed": true,
                        "actual": "1600",
                        "comparator": ">=",
                        "expected": "1333"
                    },
                    "MEMORY_MHZ_02": {
```

```
                    "Passed": true,
                    "actual": "1600",
                    "comparator": ">=",
                    "expected": "1333"
                },
                "MEMORY_MHZ_03": {
                    "Passed": true,
                    "actual": "1600",
                    "comparator": ">=",
                    "expected": "1333"
                },
                "MEMORY_MHZ_04": {
                    "Passed": true,
                    "actual": "1600",
                    "comparator": ">=",
                    "expected": "1333"
                },
                "MEMORY_MHZ_05": {
                    "Passed": true,
                    "actual": "1600",
                    "comparator": ">=",
                    "expected": "1333"
                },
                "MEMORY_MHZ_06": {
                    "Passed": true,
                    "actual": "1600",
                    "comparator": ">=",
                    "expected": "1333"
                },
                "MEMORY_MHZ_07": {
                    "Passed": true,
                    "actual": "1600",
                    "comparator": ">=",
                    "expected": "1333"
                },
                "MPTSAS_BIOS_VERSION": {
                    "Passed": true,
                    "actual": "07.24.01.00",
                    "comparator": "ANY",
                    "expected": "7.25.0.0"
                },
                "MPTSAS_FIRMWARE_VERSION": {
                    "Passed": true,
                    "actual": "13.00.57.00",
                    "comparator": "==",
                    "expected": "13.0.57.0"
```

```
                    },
                    "NETWORK_DRIVER_ETH0": {
                        "Passed": true,
                        "actual": "bnx2x",
                        "comparator": "==",
                        "expected": "bnx2x"
                    },
                    "NETWORK_DRIVER_ETH1": {
                        "Passed": true,
                        "actual": "bnx2x",
                        "comparator": "==",
                        "expected": "bnx2x"
                    },
                    "NETWORK_DRIVER_ETH2": {
                        "Passed": true,
                        "actual": "bnx2x",
                        "comparator": "==",
                        "expected": "bnx2x"
                    },
                    "NETWORK_DRIVER_ETH3": {
                        "Passed": true,
                        "actual": "bnx2x",
                        "comparator": "==",
                        "expected": "bnx2x"
                    },
                    "NETWORK_FIRMWARE_VERSION_ETH0": {
                        "Passed": true,
                        "actual": "7.10.18-solidfire-5f3ccbc781d53",
                        "comparator": "==",
                        "expected": "7.10.18-solidfire-5f3ccbc781d53"
                    },
                    "NETWORK_FIRMWARE_VERSION_ETH1": {
                        "Passed": true,
                        "actual": "7.10.18-solidfire-5f3ccbc781d53",
                        "comparator": "==",
                        "expected": "7.10.18-solidfire-5f3ccbc781d53"
                    },
                    "NETWORK_FIRMWARE_VERSION_ETH2": {
                        "Passed": true,
                        "actual": "7.10.18-solidfire-5f3ccbc781d53",
                        "comparator": "==",
                        "expected": "7.10.18-solidfire-5f3ccbc781d53"
                    },
                    "NETWORK_FIRMWARE_VERSION_ETH3": {
                        "Passed": true,
                        "actual": "7.10.18-solidfire-5f3ccbc781d53",
```

```
                        "comparator": "==",
                        "expected": "7.10.18-solidfire-5f3ccbc781d53"
                    },
                    "NUM_CPU": {
                        "Passed": true,
                        "actual": "2",
                        "comparator": "==",
                        "expected": "2"
                    },
                    "Parse failure in /var/log/sf-bios.info": {
                        "Passed": true,
                        "actual": "false",
                        "comparator": "==",
                        "expected": "false"
                    }
                },
                "duration": "00:00:00.195067",
                "result": "Passed"
            }
        }
    ]
}
}
```

NetApp Element Plug-in for vCenter Server

NetApp Element Plug-in for vCenter Server provides a plug-in to the VMware vSphere interface so that you can manage and monitor storage clusters running NetApp Element software.

To learn about Element Plug-in for vCenter Server, see the NetApp Element Plug-in for vCenter Server documentation.

For more information

- SolidFire and Element Software Documentation

Monitor storage with SolidFire Active IQ

SolidFire Active IQ is a web-based tool that provides continually updated historical views of cluster-wide data. You can set up alerts for specific events, thresholds, or metrics. SolidFire Active IQ enables you to monitor system performance and capacity, as well as stay informed about cluster health.

You can find the following information about your system in SolidFire Active IQ:

- Number of nodes and status of the nodes: healthy, offline, or fault
- Graphical representation of CPU, memory usage, and node throttling
- Details about the node, such as serial number, slot location in the chassis, model, and version of NetApp Element software running on the storage node
- CPU and storage-related information about the virtual machines

To learn about SolidFire Active IQ, see the SolidFire Active IQ documentation.

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server
- NetApp Support Site > Tools for Active IQ

Work with the management node

Management node overview

You can use the management node (mNode) to use system services, manage cluster assets and settings, run system tests and utilities, configure Active IQ for system monitoring, and enable NetApp Support access for troubleshooting.

> As a best practice, only associate one management node with one VMware vCenter instance, and avoid defining the same storage and compute resources or vCenter instances in multiple management nodes.

For clusters running Element software version 11.3 or later, you can work with the management node by using one of two interfaces:

- With the management node UI (`https://[mNode IP]:442`), you can make changes to network and cluster settings, run system tests, or use system utilities.
- With the built-in REST API UI (`https://[mNode IP]/mnode`), you can run or understand APIs relating to the management node services, including proxy server configuration, service level updates, or asset management.

Install or recover a management node:

- Install a management node
- Configure a storage Network Interface Controller (NIC)
- Recover a management node

Access the management node:

- Access the management node (UI or REST API)

Change the default SSL certificate:

- Change the management node default SSL certificate

Perform tasks with the management node UI:

- Management node UI overview

Perform tasks with the management node REST APIs:

- Management node REST API UI overview

Disable or enable remote SSH functionality or start a remote support tunnel session with NetApp Support to help you troubleshoot:

- Accessing storage nodes using SSH for basic troubleshooting
  - Enable remote NetApp Support connections
  - Manage SSH functionality on the management node

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Install or recover a management node

Install a management node

You can manually install the management node for your cluster running NetApp Element software using the appropriate image for your configuration.

This manual process is intended for SolidFire all-flash storage administrators who are not using the NetApp Deployment Engine for management node installation.

What you'll need

- Your cluster version is running NetApp Element software 11.3 or later.
- Your installation uses IPv4. The management node 11.3 does not support IPv6.

  If you need to IPv6 support, you can use the management node 11.1.

- You have permission to download software from the NetApp Support Site.
- You have identified the management node image type that is correct for your platform:

Platform	Installation image type
Microsoft Hyper-V	.iso
KVM	.iso
VMware vSphere	.iso, .ova
Citrix XenServer	.iso
OpenStack	.iso

- (Management node 12.0 and later with proxy server) You have updated NetApp Hybrid Cloud Control to management services version 2.16 before configuring a proxy server.

About this task

The Element 12.2 management node is an optional upgrade. It is not required for existing deployments.

Prior to following this procedure, you should have an understanding of Persistent volumes and whether or not you want to use them. Persistent volumes are optional but recommended for management node configuration data recovery in the event of a virtual machine (VM) loss.

Steps

1. Download ISO or OVA and deploy the VM
2. Create the management node admin and configure the network
3. Configure time sync
4. Set up the management node

5. Configure controller assets

Download ISO or OVA and deploy the VM

1. Download the OVA or ISO for your installation from the Element Software page on the NetApp Support Site.
   a. Select **Download Latest Release** and accept the EULA.
   b. Select the management node image you want to download.
2. If you downloaded the OVA, follow these steps:
   a. Deploy the OVA.
   b. If your storage cluster is on a separate subnet from your management node (eth0) and you want to use persistent volumes, add a second network interface controller (NIC) to the VM on the storage subnet (for example, eth1) or ensure that the management network can route to the storage network.
3. If you downloaded the ISO, follow these steps:
   a. Create a new 64-bit VM from your hypervisor with the following configuration:
      - Six virtual CPUs
      - 24GB RAM
      - Storage adapter type set to LSI Logic Parallel

        The default for your management node might be LSI Logic SAS. In the **New Virtual Machine** window, verify the storage adapter configuration by selecting **Customize hardware** > **Virtual Hardware**. If required, change LSI Logic SAS to **LSI Logic Parallel**.

      - 400GB virtual disk, thin provisioned
      - One virtual network interface with internet access and access to the storage MVIP.
      - (Optional) One virtual network interface with management network access to the storage cluster. If your storage cluster is on a separate subnet from your management node (eth0) and you want to use persistent volumes, add a second network interface controller (NIC) to the VM on the storage subnet (eth1) or ensure that the management network can route to the storage network.

        Do not power on the VM prior to the step indicating to do so later in this procedure.

   b. Attach the ISO to the VM and boot to the .iso install image.

      Installing a management node using the image might result in 30-second delay before the splash screen appears.

4. Power on the VM for the management node after the installation completes.

Create the management node admin and configure the network

1. Using the terminal user interface (TUI), create a management node admin user.

   To move through the menu options, press the Up or Down arrow keys. To move through the buttons, press Tab. To move from the buttons to the fields, press Tab. To navigate between fields, press the Up or Down arrow keys.

2. If there is a Dynamic Host Configuration Protocol (DHCP) server on the network that assigns IPs with a maximum transmission unit (MTU) less than 1500 bytes, you must perform the following steps:

   a. Temporarily put the management node on a vSphere network without DHCP, such as iSCSI.

   b. Reboot the VM or restart the VM network.

   c. Using the TUI, configure the correct IP on the management network with an MTU greater than or equal to 1500 bytes.

   d. Re-assign the correct VM network to the VM.

   > A DHCP that assigns IPs with an MTU less than 1500 bytes can prevent you configuring the management node network or using the management node UI.

3. Configure the management node network (eth0).

   > If you need an additional NIC to isolate storage traffic, see instructions on configuring another NIC: Configure a storage Network Interface Controller (NIC).

Configure time sync

1. Ensure time is synced between the management node and the storage cluster using NTP:

   > Starting with Element 12.3.1, substeps (a) to (e) are performed automatically. For management node 12.3.1, proceed to substep (f) to complete the time sync configuration.

   a. Log in to the management node using SSH or the console provided by your hypervisor.

   b. Stop NTPD:

   ```
   sudo service ntpd stop
   ```

   c. Edit the NTP configuration file `/etc/ntp.conf`:

      i. Comment out the default servers (`server 0.gentoo.pool.ntp.org`) by adding a `#` in front of each.

      ii. Add a new line for each default time server you want to add. The default time servers must be the same NTP servers used on the storage cluster that you will use in a later step.

      ```
      vi /etc/ntp.conf

      #server 0.gentoo.pool.ntp.org
      #server 1.gentoo.pool.ntp.org
      #server 2.gentoo.pool.ntp.org
      #server 3.gentoo.pool.ntp.org
      server <insert the hostname or IP address of the default time server>
      ```

      iii. Save the configuration file when complete.

   d. Force an NTP sync with the newly added server.

```
sudo ntpd -gq
```

e. Restart NTPD.

```
sudo service ntpd start
```

f. Disable time synchronization with host via the hypervisor (the following is a VMware example):

> If you deploy the mNode in a hypervisor environment other than VMware, for example, from the .iso image in an Openstack environment, refer to the hypervisor documentation for the equivalent commands.

i. Disable periodic time synchronization:

```
vmware-toolbox-cmd timesync disable
```

ii. Display and confirm the current status of the service:

```
vmware-toolbox-cmd timesync status
```

iii. In vSphere, verify that the `Synchronize guest time with host` box is un-checked in the VM options.

> Do not enable this option if you make future changes to the VM.

> Do not edit the NTP after you complete the time sync configuration because it affects the NTP when you run the setup command on the management node.

Set up the management node

1. Configure and run the management node setup command:

> You will be prompted to enter passwords in a secure prompt. If your cluster is behind a proxy server, you must configure the proxy settings so you can reach a public network.

```
sudo /sf/packages/mnode/setup-mnode --mnode_admin_user [username]
--storage_mvip [mvip] --storage_username [username] --telemetry_active
[true]
```

a. Replace the value in [ ] brackets (including the brackets) for each of the following required parameters:

The abbreviated form of the command name is in parentheses ( ) and can be substituted for the full name.

- **--mnode_admin_user (-mu) [username]**: The username for the management node administrator account. This is likely to be the username for the user account you used to log into the management node.
- **--storage_mvip (-sm) [MVIP address]**: The management virtual IP address (MVIP) of the storage cluster running Element software. Configure the management node with the same storage cluster that you used during NTP servers configuration.
- **--storage_username (-su) [username]**: The storage cluster administrator username for the cluster specified by the `--storage_mvip` parameter.
- **--telemetry_active (-t) [true]**: Retain the value true that enables data collection for analytics by Active IQ.

b. (Optional): Add Active IQ endpoint parameters to the command:

- **--remote_host (-rh) [AIQ_endpoint]**: The endpoint where Active IQ telemetry data is sent to be processed. If the parameter is not included, the default endpoint is used.

c. (Recommended): Add the following persistent volume parameters. Do not modify or delete the account and volumes created for persistent volumes functionality or a loss in management capability will result.

- **--use_persistent_volumes (-pv) [true/false, default: false]**: Enable or disable persistent volumes. Enter the value true to enable persistent volumes functionality.
- **--persistent_volumes_account (-pva) [account_name]**: If `--use_persistent_volumes` is set to true, use this parameter and enter the storage account name that will be used for persistent volumes.

  Use a unique account name for persistent volumes that is different from any existing account name on the cluster. It is critically important to keep the account for persistent volumes separate from the rest of your environment.

- **--persistent_volumes_mvip (-pvm) [mvip]**: Enter the management virtual IP address (MVIP) of the storage cluster running Element software that will be used with persistent volumes. This is only required if multiple storage clusters are managed by the management node. If multiple clusters are not managed, the default cluster MVIP will be used.

d. Configure a proxy server:

- **--use_proxy (-up) [true/false, default: false]**: Enable or disable the use of the proxy. This parameter is required to configure a proxy server.
- **--proxy_hostname_or_ip (-pi) [host]**: The proxy hostname or IP. This is required if you want to use a proxy. If you specify this, you will be prompted to input `--proxy_port`.
- **--proxy_username (-pu) [username]**: The proxy username. This parameter is optional.
- **--proxy_password (-pp) [password]**: The proxy password. This parameter is optional.
- **--proxy_port (-pq) [port, default: 0]**: The proxy port. If you specify this, you will be prompted to input the proxy host name or IP (`--proxy_hostname_or_ip`).
- **--proxy_ssh_port (-ps) [port, default: 443]**: The SSH proxy port. This defaults to port 443.

e. (Optional) Use parameter help if you need additional information about each parameter:

- **--help (-h)**: Returns information about each parameter. Parameters are defined as required or optional based on initial deployment. Upgrade and redeployment parameter requirements might

vary.

f. Run the `setup-mnode` command.

Configure controller assets

1. Locate the installation ID:
   a. From a browser, log into the management node REST API UI:
   b. Go to the storage MVIP and log in. This action causes the certificate to be accepted for the next step.
   c. Open the inventory service REST API UI on the management node:

      ```
      https://<ManagementNodeIP>/inventory/1/
      ```

   d. Select **Authorize** and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client`.
      iii. Select **Authorize** to begin a session.
   e. From the REST API UI, select **GET /installations**.
   f. Select **Try it out**.
   g. Select **Execute**.
   h. From the code 200 response body, copy and save the `id` for the installation for use in a later step.

      Your installation has a base asset configuration that was created during installation or upgrade.

2. Add a vCenter controller asset for NetApp Hybrid Cloud Control to the management node known assets:
   a. Access the mnode service API UI on the management node by entering the management node IP address followed by `/mnode`:

      ```
      https://<ManagementNodeIP>/mnode
      ```

   b. Select **Authorize** or any lock icon and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client`.
      iii. Select **Authorize** to begin a session.
      iv. Close the window.
   c. Select **POST /assets/{asset_id}/controllers** to add a controller sub-asset.

      > ⓘ You should create a new NetApp HCC role in vCenter to add a controller sub-asset. This new NetApp HCC role will limit the management node services view to NetApp-only assets. See Create a NetApp HCC role in vCenter.

   d. Select **Try it out**.

e. Enter the parent base asset ID you copied to your clipboard in the **asset_id** field.

f. Enter the required payload values with type `vCenter` and vCenter credentials.

g. Select **Execute**.

Find more Information

- Persistent volumes
- Add a controller asset to the management node
- Configure a storage NIC
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Configure a storage Network Interface Controller (NIC)

If you are using an additional NIC for storage, you can SSH in to the management node or use the vCenter console and run a curl command to set up a tagged or untagged network interface.

Before you begin

- You know your eth0 IP address.
- Your cluster version is running NetApp Element software 11.3 or later.
- You have deployed a management node 11.3 or later.

Configuration options

Choose the option that is relevant for your environment:

- Configure a storage Network Interface Controller (NIC) for an untagged network interface
- Configure a storage Network Interface Controller (NIC) for a tagged network interface

Configure a storage Network Interface Controller (NIC) for an untagged network interface

Steps

1. Open an SSH or vCenter console.
2. Replace the values in the following command template and run the command:

> Values are represented by `$` for each of the required parameters for your new storage network interface. The `cluster` object in the following template is required and can be used for management node host name renaming. `--insecure` or `-k` options should not be used in production environments.

```
curl -u $mnode_user_name:$mnode_password --insecure -X POST \
https://$mnode_IP:442/json-rpc/10.0 \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d ' {
     "params": {
                "network": {
                           "$eth1": {
                                    "#default" : false,
                                    "address" : "$storage_IP",
                                    "auto" : true,
                                    "family" : "inet",
                                    "method" : "static",
                                    "mtu" : "9000",
                                    "netmask" : "$subnet_mask",
                                    "status" : "Up"
                                    }
                           },
                "cluster": {
                           "name": "$mnode_host_name"
                           }
              },
    "method": "SetConfig"
}
'
```

Configure a storage Network Interface Controller (NIC) for a tagged network interface

Steps

1. Open an SSH or vCenter console.
2. Replace the values in the following command template and run the command:

> Values are represented by `$` for each of the required parameters for your new storage network interface. The `cluster` object in the following template is required and can be used for management node host name renaming. `--insecure` or `-k` options should not be used in production environments.

```
curl -u $mnode_user_name:$mnode_password --insecure -X POST \
https://$mnode_IP:442/json-rpc/10.0 \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d ' {
     "params": {
                "network": {
                           "$eth1": {
                                    "#default" : false,
                                    "address" : "$storage_IP",
                                    "auto" : true,
                                    "family" : "inet",
                                    "method" : "static",
                                    "mtu" : "9000",
                                    "netmask" : "$subnet_mask",
                                    "status" : "Up",
                                    "virtualNetworkTag" : "$vlan_id"
                                    }
                           },
                "cluster": {
                          "name": "$mnode_host_name",
                          "cipi": "$eth1.$vlan_id",
                          "sipi": "$eth1.$vlan_id"
                          }
              },
    "method": "SetConfig"
}
'
```

Find more Information

- Add a controller asset to the management node
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Recover a management node

You can manually recover and redeploy the management node for your cluster running NetApp Element software if your previous management node used persistent volumes.

You can deploy a new OVA and run a redeploy script to pull configuration data from a previously installed management node running version 11.3 and later.

What you'll need

- Your previous management node was running NetApp Element software version 11.3 or later with

Persistent volumes functionality engaged.

- You know the MVIP and SVIP of the cluster containing the persistent volumes.
- Your cluster version is running NetApp Element software 11.3 or later.
- Your installation uses IPv4. The management node 11.3 does not support IPv6.
- You have permission to download software from the NetApp Support Site.
- You have identified the management node image type that is correct for your platform:

Platform	Installation image type
Microsoft Hyper-V	.iso
KVM	.iso
VMware vSphere	.iso, .ova
Citrix XenServer	.iso
OpenStack	.iso

Steps

1. Download ISO or OVA and deploy the VM
2. Configure the network
3. Configure time sync
4. Configure the management node

Download ISO or OVA and deploy the VM

1. Download the OVA or ISO for your installation from the Element software page on the NetApp Support Site.
   a. Select **Download Latest Release** and accept the EULA.
   b. Select the management node image you want to download.
2. If you downloaded the OVA, follow these steps:
   a. Deploy the OVA.
   b. If your storage cluster is on a separate subnet from your management node (eth0) and you want to use persistent volumes, add a second network interface controller (NIC) to the VM on the storage subnet (for example, eth1) or ensure that the management network can route to the storage network.
3. If you downloaded the ISO, follow these steps:
   a. Create a new 64-bit virtual machine from your hypervisor with the following configuration:
      - Six virtual CPUs
      - 24GB RAM
      - 400GB virtual disk, thin provisioned
      - One virtual network interface with internet access and access to the storage MVIP.
      - (Optional for SolidFire all-flash storage) One virtual network interface with management network access to the storage cluster. If your storage cluster is on a separate subnet from your management node (eth0) and you want to use persistent volumes, add a second network interface controller (NIC) to the VM on the storage subnet (eth1) or ensure that the management network

can route to the storage network.

> Do not power on the virtual machine prior to the step indicating to do so later in this procedure.

b. Attach the ISO to the virtual machine and boot to the .iso install image.

> Installing a management node using the image might result in 30-second delay before the splash screen appears.

4. Power on the virtual machine for the management node after the installation completes.

Configure the network

1. Using the terminal user interface (TUI), create a management node admin user.

> To move through the menu options, press the Up or Down arrow keys. To move through the buttons, press Tab. To move from the buttons to the fields, press Tab. To navigate between fields, press the Up or Down arrow keys.

2. Configure the management node network (eth0).

> If you need an additional NIC to isolate storage traffic, see instructions on configuring another NIC: Configure a storage Network Interface Controller (NIC).

Configure time sync

1. Ensure time is synced between the management node and the storage cluster using NTP:

> Beginning with Element 12.3.1, substeps (a) to (e) are performed automatically. For management node 12.3.1 or later, proceed to substep (f) to complete the time sync configuration.

a. Log in to the management node using SSH or the console provided by your hypervisor.

b. Stop NTPD:

```
sudo service ntpd stop
```

c. Edit the NTP configuration file `/etc/ntp.conf`:

i. Comment out the default servers (`server 0.gentoo.pool.ntp.org`) by adding a `#` in front of each.

ii. Add a new line for each default time server you want to add. The default time servers must be the same NTP servers used on the storage cluster that you will use in a later step.

```
vi /etc/ntp.conf

#server 0.gentoo.pool.ntp.org
#server 1.gentoo.pool.ntp.org
#server 2.gentoo.pool.ntp.org
#server 3.gentoo.pool.ntp.org
server <insert the hostname or IP address of the default time server>
```

iii. Save the configuration file when complete.

d. Force an NTP sync with the newly added server.

```
sudo ntpd -gq
```

e. Restart NTPD.

```
sudo service ntpd start
```

f. Disable time synchronization with host via the hypervisor (the following is a VMware example):

> If you deploy the mNode in a hypervisor environment other than VMware, for example, from the .iso image in an Openstack environment, refer to the hypervisor documentation for the equivalent commands.

i. Disable periodic time synchronization:

```
vmware-toolbox-cmd timesync disable
```

ii. Display and confirm the current status of the service:

```
vmware-toolbox-cmd timesync status
```

iii. In vSphere, verify that the `Synchronize guest time with host` box is un-checked in the VM options.

> Do not enable this option if you make future changes to the VM.

> Do not edit the NTP after you complete the time sync configuration because it affects the NTP when you run the redeploy command on the management node.

Configure the management node

1. Create a temporary destination directory for the management services bundle contents:

```
mkdir -p /sf/etc/mnode/mnode-archive
```

2. Download the management services bundle (version 2.15.28 or later) that was previously installed on the existing management node and save it in the `/sf/etc/mnode/` directory.
3. Extract the downloaded bundle using the following command, replacing the value in [ ] brackets (including the brackets) with the name of the bundle file:

```
tar -C /sf/etc/mnode -xvf /sf/etc/mnode/[management services bundle
file]
```

4. Extract the resulting file to the `/sf/etc/mnode-archive` directory:

```
tar -C /sf/etc/mnode/mnode-archive -xvf
/sf/etc/mnode/services_deploy_bundle.tar.gz
```

5. Create a configuration file for accounts and volumes:

```
echo '{"trident": true, "mvip": "[mvip IP address]", "account_name":
"[persistent volume account name]"}' | sudo tee /sf/etc/mnode/mnode-
archive/management-services-metadata.json
```

   a. Replace the value in [ ] brackets (including the brackets) for each of the following required parameters:
      - **[mvip IP address]**: The management virtual IP address of the storage cluster. Configure the management node with the same storage cluster that you used during NTP servers configuration.
      - **[persistent volume account name]**: The name of the account associated with all persistent volumes in this storage cluster.
6. Configure and run the management node redeploy command to connect to persistent volumes hosted on the cluster and start services with previous management node configuration data:

> You will be prompted to enter passwords in a secure prompt. If your cluster is behind a proxy server, you must configure the proxy settings so you can reach a public network.

```
sudo /sf/packages/mnode/redeploy-mnode --mnode_admin_user [username]
```

   a. Replace the value in [ ] brackets (including the brackets) with the user name for the management node administrator account. This is likely to be the username for the user account you used to log into the management node.

You can add the user name or allow the script to prompt you for the information.

b. Run the `redeploy-mnode` command. The script displays a success message when the redeployment is complete.

c. If you access Element web interfaces (such as the management node or NetApp Hybrid Cloud Control) using the Fully Qualified Domain Name (FQDN) of the system, reconfigure authentication for the management node.

SSH capability that provides NetApp Support remote support tunnel (RST) session access is disabled by default on management nodes running management services 2.18 and later. If you had previously enabled SSH functionality on the management node, you might need to disable SSH again on the recovered management node.

Find more Information

- Persistent volumes
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Access the management node

Beginning with NetApp Element software version 11.3, the management node contains two UIs: a UI for managing REST-based services and a per-node UI for managing network and cluster settings and operating system tests and utilities.

For clusters running Element software version 11.3 or later, you can make use one of two interfaces:

- By using the management node UI (`https:// [mNode IP}:442`), you can make changes to network and cluster settings, run system tests, or use system utilities.
- By using the built-in REST API UI (`https://[mNode IP}/mnode`), you can run or understand APIs relating to the management node services, including proxy server configuration, service level updates, or asset management.

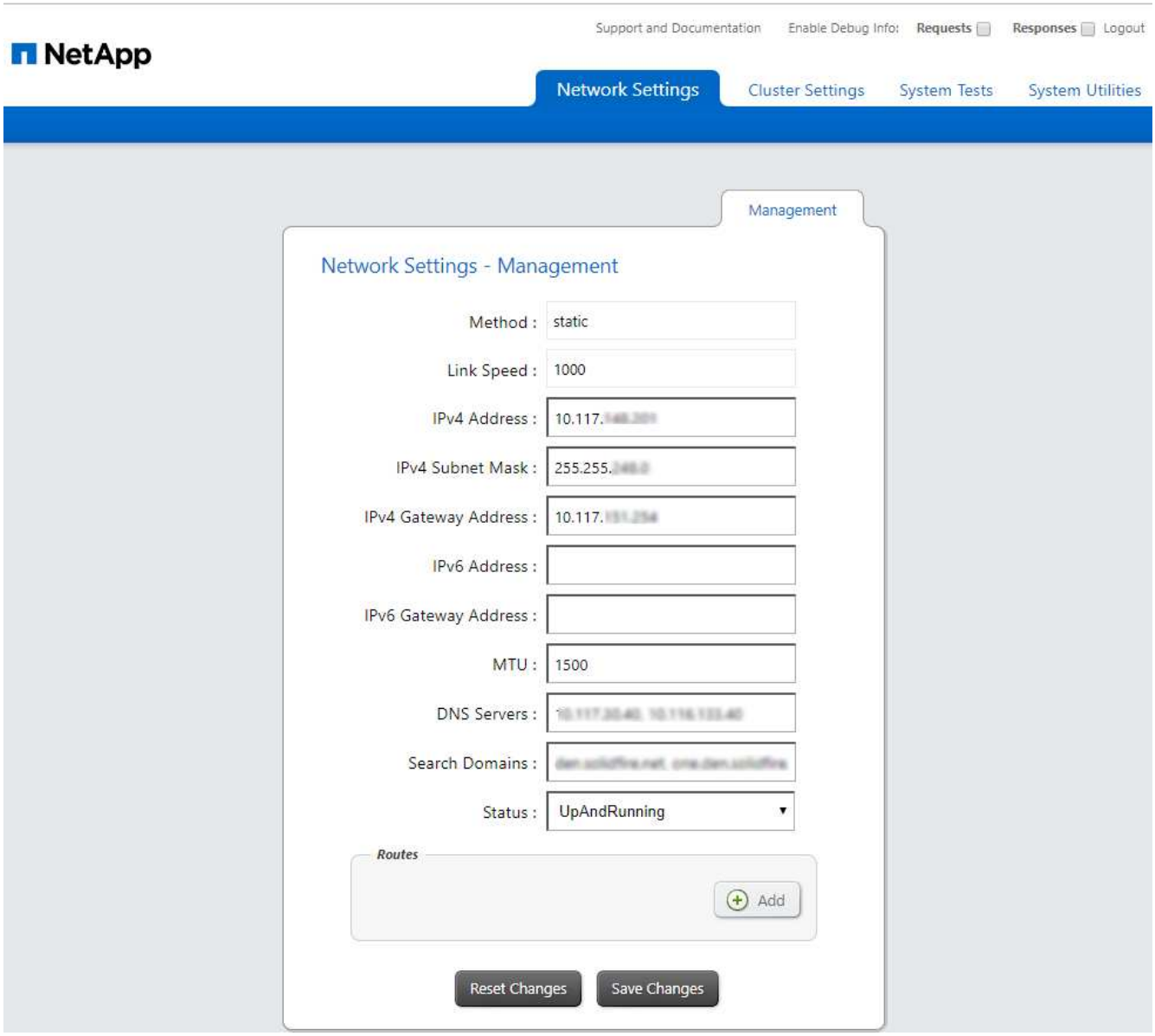
Access the management node per-node UI

From the per-node UI, you can access network and cluster settings and utilize system tests and utilities.

Steps

1. Access the per-node UI for the management node by entering the management node IP address followed by :442

```
https://[IP address]:442
```



2. Enter the management node user name and password when prompted.

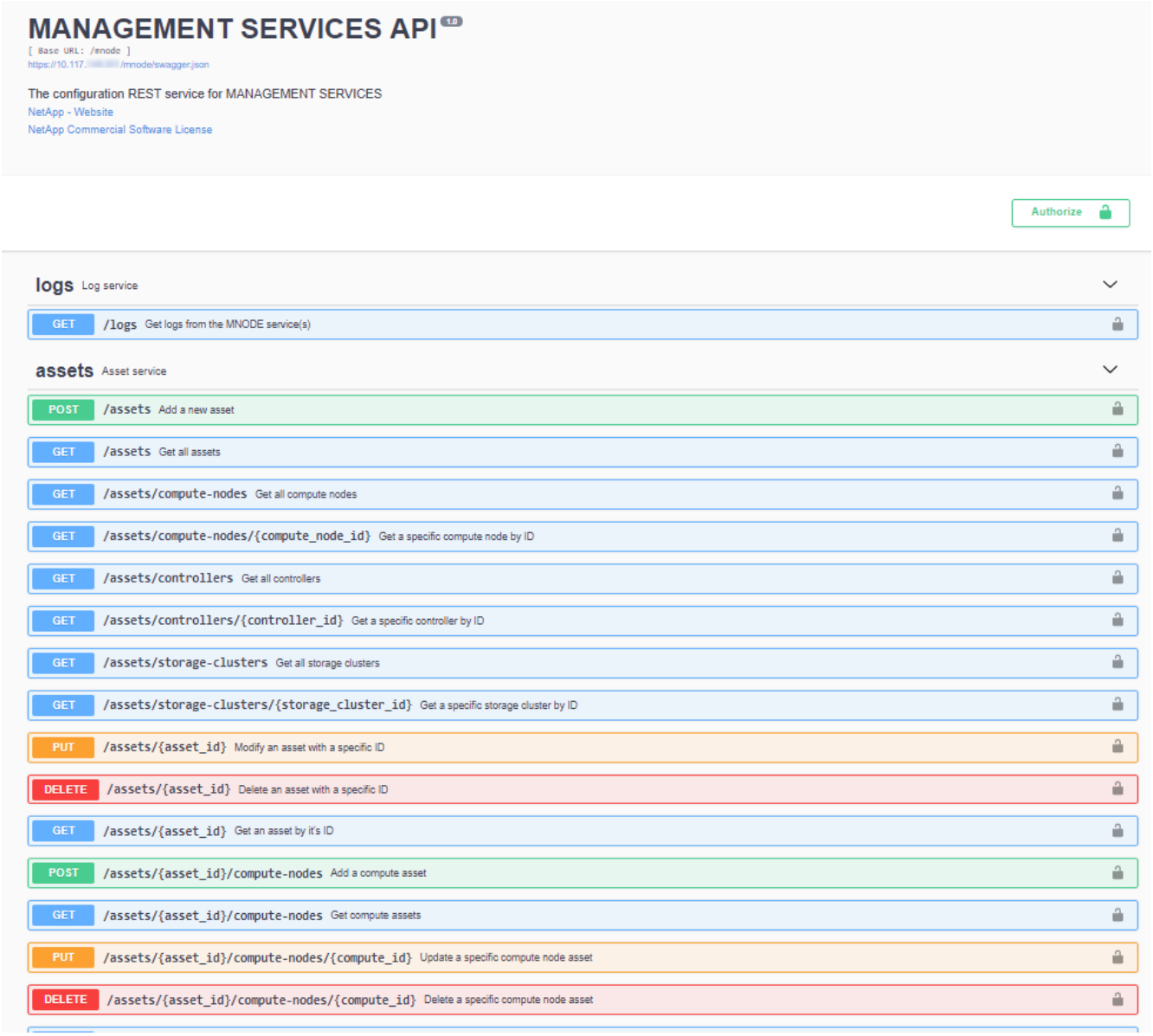
Access the management node REST API UI

From the REST API UI, you can access a menu of service-related APIs that control management services on the management node.

Steps

1. To access the REST API UI for management services, enter the management node IP address followed by `/mnode`:

```
https://[IP address]/mnode
```



2. Select **Authorize** or any lock icon and enter cluster admin credentials for permissions to use APIs.

Find more Information

- Enable Active IQ and NetApp monitoring
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Work with the management node UI

Management node UI overview

With the management node UI (`https://<ManagementNodeIP>:442`), you can make changes to network and cluster settings, run system tests, or use system utilities.

Tasks you can perform with the management node UI:

- Configure alert monitoring
- Modify and test the management node network, cluster, and system settings
- Run system utilities from the management node

Find more information

- Access the management node
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Configure alert monitoring

The alert monitoring tools are configured for NetApp HCI alert monitoring. These tools are not configured or used for SolidFire all-flash storage. Running the tools for these clusters results in the following 405 error, which is expected given the configuration: `webUIParseError : Invalid response from server. 405`

For more information on configuring alert monitoring for NetApp HCI, see Configure alert monitoring

Modify and test the management node network, cluster, and system settings

You can modify and test the management node network, cluster, and system settings.

- Update management node network settings
- Update management node cluster settings
- Test the management node settings

Update management node network settings

On the Network Settings tab of the per-node management node UI, you can modify the management node network interface fields.

1. Open the per-node management node UI.
2. Select the **Network Settings** tab.
3. View or enter the following information:
   a. **Method**: Choose one of the following methods to configure the interface:
      - `loopback`: Use to define the IPv4 loopback interface.
      - `manual`: Use to define interfaces for which no configuration is done by default.
      - `dhop`: Use to obtain an IP address via DHCP.
      - `static`: Use to define Ethernet interfaces with statically allocated IPv4 addresses.
   b. **Link Speed**: The speed negotiated by the virtual NIC.
   c. **IPv4 Address**: The IPv4 address for the eth0 network.
   d. **IPv4 Subnet Mask**: Address subdivisions of the IPv4 network.
   e. **IPv4 Gateway Address**: Router network address to send packets out of the local network.

f. **IPv6 Address**: The IPv6 address for the eth0 network.

g. **IPv6 Gateway Address**: Router network address to send packets out of the local network.

> The IPv6 options are not supported for 11.3 or later versions of the management node.

h. **MTU**: Largest packet size that a network protocol can transmit. Must be greater than or equal to 1500. If you add a second storage NIC, the value should be 9000.

i. **DNS Servers**: Network interface used for cluster communication.

j. **Search Domains**: Search for additional MAC addresses available to the system.

k. **Status**: Possible values:

- `UpAndRunning`
- `Down`
- `Up`

l. **Routes**: Static routes to specific hosts or networks via the associated interface the routes are configured to use.

Update management node cluster settings

On the Cluster Settings tab of the per-node UI for the management node, you can modify cluster interface fields when a node is in Available, Pending, PendingActive, and Active states.

1. Open the per-node management node UI.
2. Select the **Cluster Settings** tab.
3. View or enter the following information:
   - **Role**: Role the management node has in the cluster. Possible value: `Management`.
   - **Version**: Element software version running on the cluster.
   - **Default Interface**: Default network interface used for management node communication with the cluster running Element software.

Test the management node settings

After you change management and network settings for the management node and commit the changes, you can run tests to validate the changes you made.

1. Open the per-node management node UI.
2. In the management node UI, select **System Tests**.
3. Complete any of the following:
   a. To verify that the network settings you configured are valid for the system, select **Test Network Config**.
   b. To test network connectivity to all nodes in the cluster on both 1G and 10G interfaces using ICMP packets, select **Test Ping**.
4. View or enter the following:
   - **Hosts**: Specify a comma-separated list of addresses or host names of devices to ping.
   - **Attempts**: Specify the number of times the system should repeat the test ping. Default: 5.

- **Packet Size**: Specify the number of bytes to send in the ICMP packet that is sent to each IP. The number of bytes must be less than the maximum MTU specified in the network configuration.
- **Timeout mSec**: Specify the number of milliseconds to wait for each individual ping response. Default: 500 ms.
- **Total Timeout Sec**: Specify the time in seconds the ping should wait for a system response before issuing the next ping attempt or ending the process. Default: 5.
- **Prohibit Fragmentation**: Enable the DF (do not fragment) flag for the ICMP packets.

Find more Information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Run system utilities from the management node

You can use the per-node UI for the management node to create or delete cluster support bundles, reset node configuration settings, or restart networking.

Steps

1. Open the per-node management node UI using the management node admin credentials.
2. Select **System Utilities**.
3. Select the button for the utility that you want to run:
   a. **Control Power**: Reboots, power cycles, or shuts down the node. Specify any of the following options.

      This operation causes temporary loss of networking connectivity.

      - **Action**: Options include `Restart` and `Halt` (power off).
      - **Wakeup Delay**: Any additional time before the node comes back online.

   b. **Create Cluster Support Bundle**: Creates the cluster support bundle to assist NetApp Support diagnostic evaluations of one or more nodes in a cluster. Specify the following options:
      - **Bundle Name**: Unique name for each support bundle created. If no name is provided, then "supportbundle" and the node name are used as the file name.
      - **Mvip**: The MVIP of the cluster. Bundles are gathered from all nodes in the cluster. This parameter is required if the Nodes parameter is not specified.
      - **Nodes**: The IP addresses of the nodes from which to gather bundles. Use either Nodes or Mvip, but not both, to specify the nodes from which to gather bundles. This parameter is required if Mvip is not specified.
      - **Username**: The cluster admin user name.
      - **Password**: The cluster admin password.
      - **Allow Incomplete**: Allows the script to continue to run if bundles cannot be gathered from one or more of the nodes.
      - **Extra Args**: This parameter is fed to the `sf_make_support_bundle` script. This parameter should be used only at the request of NetApp Support.

   c. **Delete All Support Bundles**: Deletes any current support bundles on the management node.

d. **Reset Node**: Resets the management node to a new install image. This changes all settings except the network configuration to the default state. Specify the following options:

   - **Build**: The URL to a remote Element software image to which the node will be reset.
   - **Options**: Specifications for running the reset operations. Details are be provided by NetApp Support, if required.

     This operation causes temporary loss of networking connectivity.

e. **Restart Networking**: Restarts all networking services on the management node.

   This operation causes temporary loss of networking connectivity.

Find more Information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Work with the management node REST API

Management node REST API UI overview

By using the built-in REST API UI (`https://<ManagementNodeIP>/mnode`), you can run or understand APIs relating to the management node services, including proxy server configuration, service level updates, or asset management.

Tasks you can perform with REST APIs:

Authorization

- Get authorization to use REST APIs

Asset configuration

- Enable Active IQ and NetApp monitoring
- Configure a proxy server for the management node
- Configure NetApp Hybrid Cloud Control for multiple vCenters
- Add a controller asset to the management node
- Create and manage storage cluster assets

Asset management

- View or edit existing controller assets
- Create and manage storage cluster assets
- Use the REST API to collect Element system logs
- Verify management node OS and services versions
- Getting logs from management services

Find more information

- Access the management node
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Get authorization to use REST APIs

You must authorize before you can use APIs for management services in the REST API UI. You do this by obtaining an access token.

To obtain a token, you provide cluster admin credentials and a client ID. Each token lasts approximately ten minutes. After a token expires, you can authorize again for a new access token.

Authorization functionality is set up for you during management node installation and deployment. The token service is based on the storage cluster you defined during setup.

Before you begin

- Your cluster version should be running NetApp Element software 11.3 or later.
- You should have deployed a management node running version 11.3 or later.

API command

```
TOKEN=`curl -k -X POST https://MVIP/auth/connect/token -F client_id=mnode-
client -F grant_type=password -F username=CLUSTER_ADMIN -F
password=CLUSTER_PASSWORD|awk -F':' '{print $2}'|awk -F',' '{print
$1}'|sed s/\"//g`
```

REST API UI steps

1. Access the REST API UI for the service by entering the management node IP address followed by the service name, for example `/mnode/`:

   ```
   https://<ManagementNodeIP>/mnode/
   ```

2. Select **Authorize**.

   Alternately, you can select on a lock icon next to any service API.

3. Complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Do not enter a value for the client secret.
   d. Select **Authorize** to begin a session.
4. Close the **Available authorizations** dialog box.

If you try to run a command after the token expires, a `401 Error: UNAUTHORIZED` message appears. If you see this, authorize again.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Enable Active IQ and NetApp monitoring

You can enable Active IQ storage monitoring if you did not already do so during installation or upgrade. You might need to use this procedure if you did not set up SolidFire Active IQ during installation for a SolidFire all-flash storage system.

The Active IQ collector service forwards configuration data and Element software-based cluster performance metrics to SolidFire Active IQ for historical reporting and near real-time performance monitoring. The NetApp monitoring service enables forwarding of storage cluster faults to vCenter for alert notification.

Before you begin

- Some functions in Active IQ, for example, quality of service (QoS), require Element 11.3 or later to work correctly. To confirm that you have the capability to use all Active IQ functions, NetApp recommends the following:
  - Your storage cluster is running NetApp Element software 11.3 or later.
  - You have deployed a management node running version 11.3 or later.
- You have internet access. The Active IQ collector service cannot be used from dark sites that do not have external connectivity.

Steps

1. Get the base asset ID for the installation:
   a. Open the inventory service REST API UI on the management node:

      ```
      https://<ManagementNodeIP>/inventory/1/
      ```

   b. Select **Authorize** and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client`.
      iii. Select **Authorize** to begin a session.
      iv. Close the window.
   c. From the REST API UI, select **GET /installations**.
   d. Select **Try it out**.
   e. Select **Execute**.
   f. From the code 200 response body, copy the `id` for the installation.

```
{
  "installations": [
    {
      "_links": {
        "collection":
"https://10.111.211.111/inventory/1/installations",
        "self":
"https://10.111.217.111/inventory/1/installations/abcd01e2-ab00-1xxx-
91ee-12f111xxc7x0x"
      },
      "id": "abcd01e2-ab00-1xxx-91ee-12f111xxc7x0x",
```

Your installation has a base asset configuration that was created during installation or upgrade.

2. Activate telemetry:
   a. Access the mnode service API UI on the management node by entering the management node IP address followed by `/mnode`:

```
https://<ManagementNodeIP>/mnode
```

   b. Select **Authorize** or any lock icon and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client`.
      iii. Select **Authorize** to begin a session.
      iv. Close the window.
   c. Configure the base asset:
      i. Select **PUT /assets/{asset_id}**.
      ii. Select **Try it out**.
      iii. Enter the following in the JSON payload:

```
{
"telemetry_active": true
"config": {}
}
```

      iv. Enter the base ID from the previous step in **asset_ID**.
      v. Select **Execute**.

      The Active IQ service is automatically restarted whenever assets are changed. Modifying assets results in a short delay before settings are applied.

3. If you have not already done so, add a vCenter controller asset for NetApp Hybrid Cloud Control to the management node known assets:

   A controller asset is required for NetApp monitoring services.

   a. Select **POST /assets/{asset_id}/controllers** to add a controller sub-asset.
   b. Select **Try it out**.
   c. Enter the parent base asset ID you copied to your clipboard in the **asset_id** field.
   d. Enter the required payload values with `type` as `vCenter` and vCenter credentials.

```
{
"username": "string",
"password": "string",
"ip": "string",
"type": "vCenter",
"host_name": "string",
"config": {}
}
```

   `ip` is the vCenter IP address.

   e. Select **Execute**.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Configure NetApp Hybrid Cloud Control for multiple vCenters

You can configure NetApp Hybrid Cloud Control to manage assets from two or more vCenters that are not using Linked Mode.

You should use this process after your initial installation when you need to add assets for a recently scaled installation or when new assets were not added automatically to your configuration. Use these APIs to add assets that are recent additions to your installation.

What you'll need

- Your cluster version is running NetApp Element software 11.3 or later.
- You have deployed a management node running version 11.3 or later.

Steps

1. Add new vCenters as controller assets to the management node configuration.
2. Refresh the inventory service API on the management node:

```
https://<ManagementNodeIP>/inventory/1/
```

As an alternative, you can wait 2 minutes for the inventory to update in NetApp Hybrid Cloud Control UI.

a. Select **Authorize** and complete the following:
   i. Enter the cluster user name and password.
   ii. Enter the client ID as `mnode-client`.
   iii. Select **Authorize** to begin a session.
   iv. Close the window.
b. From the REST API UI, select **GET /installations**.
c. Select **Try it out**.
d. Select **Execute**.
e. From the response, copy the installation asset ID (`"id"`).
f. From the REST API UI, select **GET /installations/{id}**.
g. Select **Try it out**.
h. Set refresh to `True`.
i. Paste the installation asset ID into the **id** field.
j. Select **Execute**.

3. Refresh the NetApp Hybrid Cloud Control browser to see the changes.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Add a controller asset to the management node

You can add a controller asset to the management node configuration using the REST API UI.

You might need to add an asset if you recently scaled your installation and new assets were not added automatically to your configuration. Use these APIs to add assets that are recent additions to your installation.

What you'll need

- Your cluster version is running NetApp Element software 11.3 or later.
- You have deployed a management node running version 11.3 or later.
- You have created a new NetApp HCC role in vCenter to limit the management node services view to NetApp-only assets. See Create a NetApp HCC role in vCenter

Steps

1. Get the base asset ID for the installation:

a. Open the inventory service REST API UI on the management node:

```
https://<ManagementNodeIP>/inventory/1/
```

b. Select **Authorize** and complete the following:

i. Enter the cluster user name and password.

ii. Enter the client ID as `mnode-client`.

iii. Select **Authorize** to begin a session.

iv. Close the window.

c. From the REST API UI, select **GET /installations**.

d. Select **Try it out**.

e. Select **Execute**.

f. From the code 200 response body, copy the `id` for the installation.

```
{
  "installations": [
    {
      "_links": {
        "collection":
"https://10.111.211.111/inventory/1/installations",
        "self":
"https://10.111.217.111/inventory/1/installations/abcd01e2-ab00-1xxx-
91ee-12f111xxc7x0x"
      },
      "id": "abcd01e2-ab00-1xxx-91ee-12f111xxc7x0x",
```

Your installation has a base asset configuration that was created during installation or upgrade.

g. From the REST API UI, select **GET /installations/{id}**.

h. Select **Try it out**.

i. Paste the installation asset ID into the **id** field.

j. Select **Execute**.

k. From the response, copy and save the cluster controller ID (`"controllerId"`) for use in a later step.

2. To add a controller sub-asset to an existing base asset, select:

```
POST /assets/{asset_id}/controllers
```

a. Open the mNode service REST API UI on the management node:

```
https://<ManagementNodeIP>/mnode
```

b. Select **Authorize** and complete the following:

   i. Enter the cluster user name and password.

   ii. Enter the client ID as `mnode-client`.

   iii. Select **Authorize** to begin a session.

   iv. Close the window.

c. Select **POST /assets/{asset_id}/controllers**.

d. Select **Try it out**.

e. Enter the parent base asset ID in the **asset_id** field.

f. Add the required values to the payload.

g. Select **Execute**.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Create and manage storage cluster assets

You can add new storage cluster assets to the management node, edit the stored credentials for known storage cluster assets, and delete storage cluster assets from the management node using the REST API.

What you'll need

- Ensure that your storage cluster version is running NetApp Element software 11.3 or later.
- Ensure that you have deployed a management node running version 11.3 or later.

Storage cluster asset management options

Choose one of the following options:

- Retrieve the installation ID and cluster ID of a storage cluster asset
- Add a new storage cluster asset
- Edit the stored credentials for a storage cluster asset
- Delete a storage cluster asset

Retrieve the installation ID and cluster ID of a storage cluster asset

You can use the REST API get the installation ID and the ID of the storage cluster. You need the installation ID to add a new storage cluster asset, and the cluster ID to modify or delete a specific storage cluster asset.

Steps

1. Access the REST API UI for the inventory service by entering the management node IP address followed by `/inventory/1/`:

```
https://<ManagementNodeIP>/inventory/1/
```

2. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. Select **GET /installations**.
4. Select **Try it out**.
5. Select **Execute**.

   The API returns a list of all known installations.

6. From the code 200 response body, save the value in the `id` field, which you can find in the list of installations. This is the installation ID. For example:

```
"installations": [
    {
      "id": "1234a678-12ab-35dc-7b4a-1234a5b6a7ba",
      "name": "my-sf-installation",
      "_links": {
        "collection": "https://localhost/inventory/1/installations",
        "self": "https://localhost/inventory/1/installations/1234a678-
12ab-35dc-7b4a-1234a5b6a7ba"
      }
    }
  ]
```

7. Access the REST API UI for the storage service by entering the management node IP address followed by `/storage/1/`:

```
https://<ManagementNodeIP>/storage/1/
```

8. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
9. Select **GET /clusters**.
10. Select **Try it out**.

11. Enter the installation ID you saved earlier into the `installationId` parameter.
12. Select **Execute**.

    The API returns a list of all known storage clusters in this installation.

13. From the code 200 response body, find the correct storage cluster and save the value in the cluster's `storageId` field. This is the storage cluster ID.

Add a new storage cluster asset

You can use the REST API to add one or more new storage cluster assets to the management node inventory. When you add a new storage cluster asset, it is automatically registered with the management node.

What you'll need

- You have copied the storage cluster ID and installation ID for any storage clusters you want to add.
- If you are adding more than one storage node, you have read and understood the limitations of the Authoritative cluster and multiple storage cluster support.

> ⓘ All users defined on the authoritative cluster are defined as users on all other clusters tied to the NetApp Hybrid Cloud Control instance.

Steps

1. Access the REST API UI for the storage service by entering the management node IP address followed by `/storage/1/`:

   ```
   https://<ManagementNodeIP>/storage/1/
   ```

2. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. Select **POST /clusters**.
4. Select **Try it out**.
5. Enter the new storage cluster's information in the following parameters in the **Request body** field:

   ```
   {
     "installationId": "a1b2c34d-e56f-1a2b-c123-1ab2cd345d6e",
     "mvip": "10.0.0.1",
     "password": "admin",
     "userId": "admin"
   }
   ```

Parameter	Type	Description
`installationId`	string	The installation in which to add the new storage cluster. Enter the installation ID you saved earlier into this parameter.
`mvip`	string	The IPv4 management virtual IP address (MVIP) of the storage cluster.
`password`	string	The password used to communicate with the storage cluster.
`userId`	string	The user ID used to communicate with the storage cluster (the user must have administrator privileges).

6. Select **Execute**.

   The API returns an object containing information about the newly added storage cluster asset, such as the name, version, and IP address information.

Edit the stored credentials for a storage cluster asset

You can edit the stored credentials that the management node uses to log in to a storage cluster. The user you choose must have cluster admin access.

> Ensure you have followed the steps in Retrieve the installation ID and cluster ID of a storage cluster asset before continuing.

Steps

1. Access the REST API UI for the storage service by entering the management node IP address followed by `/storage/1/`:

   ```
   https://<ManagementNodeIP>/storage/1/
   ```

2. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. Select **PUT /clusters/{storageId}**.
4. Select **Try it out**.
5. Paste the storage cluster ID you copied earlier into the `storageId` parameter.
6. Change one or both of the following parameters in the **Request body** field:

```
{
  "password": "adminadmin",
  "userId": "admin"
}
```

Parameter	Type	Description
password	string	The password used to communicate with the storage cluster.
userId	string	The user ID used to communicate with the storage cluster (the user must have administrator privileges).

7. Select **Execute**.

Delete a storage cluster asset

You can delete a storage cluster asset if the storage cluster is no longer in service. When you remove a storage cluster asset, it is automatically unregistered from the management node.

Ensure you have followed the steps in Retrieve the installation ID and cluster ID of a storage cluster asset before continuing.

Steps

1. Access the REST API UI for the storage service by entering the management node IP address followed by `/storage/1/`:

```
https://<ManagementNodeIP>/storage/1/
```

2. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. Select **DELETE /clusters/{storageId}**.
4. Select **Try it out**.
5. Enter the storage cluster ID you copied earlier in the `storageId` parameter.
6. Select **Execute**.

   Upon success, the API returns an empty response.

Find more information

- Authoritative cluster
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

View or edit existing controller assets

You can view information about and edit existing VMware vCenter controllers in the management node configuration using the REST API. Controllers are VMware vCenter instances registered to the management node for your NetApp SolidFire installation.

Before you begin

- Ensure that your cluster version is running NetApp Element software 11.3 or later.
- Ensure that you have deployed a management node running version 11.3 or later.

Access the management services REST API

Steps

1. Access the REST API UI for management services by entering the management node IP address followed by `/vcenter/1/`:

   ```
   https://<ManagementNodeIP>/vcenter/1/
   ```

2. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.

View stored information about existing controllers

You can list existing vCenter controllers that are registered with the management node and view stored information about them using the REST API.

Steps

1. Select **GET /compute/controllers**.
2. Select **Try it out**.
3. Select **Execute**.

   The API returns a list of all known vCenter controllers, along with the IP address, controller ID, hostname, and user ID used to communicate with each controller.

4. If you want the connection status of a particular controller, copy the controller ID from the `id` field of that controller to your clipboard and see View the status of an existing controller.

View the status of an existing controller

You can view the status of any of the existing vCenter controllers registered with the management node. The API returns a status indicating whether NetApp Hybrid Cloud Control can connect with the vCenter controller as well as the reason for that status.

Steps

1. Select **GET /compute/controllers/{controller_id}/status**.
2. Select **Try it out**.
3. Enter the controller ID you copied earlier in the `controller_id` parameter.
4. Select **Execute**.

   The API returns a status of this particular vCenter controller, along with a reason for that status.

Edit the stored properties of a controller

You can edit the stored user name or password for any of the existing vCenter controllers registered with the management node. You cannot edit the stored IP address of an existing vCenter controller.

Steps

1. Select **PUT /compute/controllers/{controller_id}**.
2. Enter the controller ID of a vCenter controller in the `controller_id` parameter.
3. Select **Try it out**.
4. Change either of the following parameters in the **Request body** field:

Parameter	Type	Description
`userId`	string	Change the user ID used to communicate with the vCenter controller (the user must have administrator privileges).
`password`	string	Change the password used to communicate with the vCenter controller.

5. Select **Execute**.

   The API returns updated controller information.

Find more information

- Add a controller asset to the management node
- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Configure a proxy server

If your cluster is behind a proxy server, you must configure the proxy settings so that you

can reach a public network.

A proxy server is used for telemetry collectors and reverse tunnel connections. You can enable and configure a proxy server using the REST API UI if you did not already configure a proxy server during installation or upgrade. You can also modify existing proxy server settings or disable a proxy server.

The command to configure a proxy server updates and then returns the current proxy settings for the management node. The proxy settings are used by Active IQ, the NetApp monitoring service, and other Element software utilities that are installed on the management node, including the reverse support tunnel for NetApp Support.

Before you begin

- You should know host and credential information for the proxy server you are configuring.
- Ensure that your cluster version is running NetApp Element software 11.3 or later.
- Ensure that you have deployed a management node running version 11.3 or later.
- (Management node 12.0 and later) You have updated NetApp Hybrid Cloud Control to management services version 2.16 before configuring a proxy server.

Steps

1. Access the REST API UI on the management node by entering the management node IP address followed by `/mnode`:

   ```
   https://<ManagementNodeIP>/mnode
   ```

2. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. Select **PUT /settings**.
4. Select **Try it out**.
5. To enable a proxy server, you must set `use_proxy` to true. Enter the IP or host name and proxy port destinations.

   The proxy user name, proxy password, and SSH port are optional and should be omitted if not used.

   ```
   {
   "proxy_ip_or_hostname": "[IP or name]",
   "use_proxy": [true/false],
   "proxy_username": "[username]",
   "proxy_password": "[password]",
   "proxy_port": [port value],
   "proxy_ssh_port": [port value: default is 443]
   }
   ```

6. Select **Execute**.

You might need to reboot your management node depending on your environment.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Verify management node OS and services versions

You can verify the version numbers of the management node OS, management services bundle, and individual services running on the management node using the REST API in the management node.

What you'll need

- Your cluster is running NetApp Element software 11.3 or later.
- You have deployed a management node running version 11.3 or later.

Options

- API commands
- REST API UI steps

API commands

- Get version information about the management node OS, the management services bundle, and the management node API (mnode-api) service that are running on the management node:

```
curl -X GET "https://<ManagementNodeIP>/mnode/about" -H  "accept:
application/json"
```

- Get version information about individual services running on the management node:

```
curl -X GET "https://<ManagementNodeIP>/mnode/services?status=running"
-H  "accept: */*" -H  "Authorization: ${TOKEN}"
```

You can find the bearer `${TOKEN}` used by the API command when you authorize. The bearer `${TOKEN}` is in the curl response.

REST API UI steps

1. Access the REST API UI for the service by entering the management node IP address followed by `/mnode/`:

```
https://<ManagementNodeIP>/mnode/
```

2. Do one of the following:
   - Get version information about the management node OS, the management services bundle, and the management node API (mnode-api) service that are running on the management node:
     a. Select **GET /about**.
     b. Select **Try it out**.
     c. Select **Execute**.

     The management services bundle version (`"mnode_bundle_version"`), management node OS version (`"os_version"`), and management node API version (`"version"`) are indicated in the response body.
   - Get version information about individual services running on the management node:
     a. Select **GET /services**.
     b. Select **Try it out**.
     c. Select the status as **Running**.
     d. Select **Execute**.

     The services that are running on the management node are indicated in the response body.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Getting logs from management services

You can retrieve logs from the services running on the management node using the REST API. You can pull logs from all public services or specify specific services and use query parameters to better define the return results.

What you'll need

- Your cluster version is running NetApp Element software 11.3 or later.
- You have deployed a management node running version 11.3 or later.

Steps

1. Open the REST API UI on the management node.
   - Beginning with management services 2.21.61:

     ```
     https://<ManagementNodeIP>/mnode/4/
     ```

   - For management services 2.20.69 or earlier:

```
https://<ManagementNodeIP>/mnode
```

2. Select **Authorize** or any lock icon and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as mnode-client if the value is not already populated.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. Select **GET /logs**.
4. Select **Try it out**.
5. Specify the following parameters:
   - `Lines`: Enter the number of lines you want the log to return. This parameter is an integer that defaults to 1000.

     Avoid requesting the entire history of log content by setting Lines to 0.

   - `since`: Adds a ISO-8601 timestamp for the service logs starting point.

     Use a reasonable `since` parameter when gathering logs of wider timespans.

   - `service-name`: Enter a service name.

     Use the `GET /services` command to list services on the management node.

   - `stopped`: Set to `true` to retrieve logs from stopped services.
6. Select **Execute**.
7. From the response body, select **Download** to save the log output.

Find more Information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Manage support connections

Accessing storage nodes using SSH for basic troubleshooting

Beginning with Element 12.5, you can use the sfreadonly system account on the storage nodes for basic troubleshooting. You can also enable and open remote support tunnel access for NetApp Support for advanced troubleshooting.

The sfreadonly system account enables access to run basic Linux system and network troubleshooting commands, including `ping`.

Unless advised by NetApp Support, any alterations to this system are unsupported, voiding your support contract, and might result in instability or inaccessibility of data.

Before you begin

- **Write permissions**: Verify that you have write permissions to the current working directory.
- **(Optional) Generate your own key pair**: Run `ssh-keygen` from Windows 10, MacOS, or Linux distribution. This is a one-time action to create a user key pair and can be reused for future troubleshooting sessions. You might want to use certificates associated with employee accounts, which would also work in this model.
- **Enable SSH capability on the management node**: To enable remote access functionality on the management mode, see this topic. For management services 2.18 and later, the capability for remote access is disabled on the management node by default.
- **Enable SSH capability on the storage cluster**: To enable remote access functionality on the storage cluster nodes, see this topic.
- **Firewall configuration**: If your management node is behind a proxy server, the following TCP ports are required in the sshd.config file:

TCP port	Description	Connection direction
443	API calls/HTTPS for reverse port forwarding via open support tunnel to the web UI	Management node to storage nodes
22	SSH login access	Management node to storage nodes or from storage nodes to management node

Troubleshooting options

- Troubleshoot a cluster node
- Troubleshoot a cluster node with NetApp Support
- Troubleshoot a node that is not part of cluster

Troubleshoot a cluster node

You can perform basic troubleshooting using the sfreadonly system account:

Steps

1. SSH to the management node using your account login credentials you selected when installing the management node VM.
2. On the management node, go to `/sf/bin`.
3. Find the appropriate script for your system:
   - SignSshKeys.ps1
   - SignSshKeys.py
   - SignSshKeys.sh

SignSshKeys.ps1 is dependent on PowerShell 7 or later and SignSshKeys.py is dependent on Python 3.6.0 or later and the requests module.

The `SignSshKeys` script writes `user`, `user.pub`, and `user-cert.pub` files into the current working directory, which are later used by the `ssh` command. However, when a public key file is provided to the script, only a `<public_key>` file (with `<public_key>` replaced with the prefix of the public key file passed into the script) is written out to the directory.

4. Run the script on the management node to generate the SSH keychain. The script enables SSH access using the sfreadonly system account across all nodes in the cluster.

```
SignSshKeys --ip [ip address] --user [username] --duration [hours]
--publickey [public key path]
```

   a. Replace the value in [ ] brackets (including the brackets) for each of the following parameters:

      You can use either the abbreviated or full form parameter.

      - **--ip | -i [ip address]**: IP address of the target node for the API to run against.
      - **--user | -u [username]**: Cluster user used to run the API call.
      - **(Optional) --duration | -d [hours]**: The duration a signed key should remain valid as an integer in hours. The default is 24 hours.
      - **(Optional) --publickey | -k [public key path]**: The path to a public key, if the user chooses to provide one.

   b. Compare your input against the following sample command. In this example, `10.116.139.195` is the IP of the storage node, `admin` is the cluster username, and the duration of key validity is two hours:

```
sh /sf/bin/SignSshKeys.sh --ip 10.116.139.195 --user admin --duration
2
```

   c. Run the command.

5. SSH to the node IPs:

```
ssh -i user sfreadonly@[node_ip]
```

   You will be able to run basic Linux system and network troubleshooting commands, such as `ping`, and other read-only commands.

6. (Optional) Disable remote access functionality again after troubleshooting is complete.

   SSH remains enabled on the management node if you do not disable it. SSH enabled configuration persists on the management node through updates and upgrades until it is manually disabled.

Troubleshoot a cluster node with NetApp Support

NetApp Support can perform advanced troubleshooting with a system account that allows a technician to run deeper Element diagnostics.

Steps

1. SSH to the management node using your account login credentials you selected when installing the management node VM.
2. Run the rst command with the port number sent by NetApp Support to open the support tunnel:

   ```
   rst -r sfsupport.solidfire.com -u element -p <port_number>
   ```

   NetApp Support will log in to your management node using the support tunnel.

3. On the management node, go to `/sf/bin`.
4. Find the appropriate script for your system:
   - SignSshKeys.ps1
   - SignSshKeys.py
   - SignSshKeys.sh

   > SignSshKeys.ps1 is dependent on PowerShell 7 or later and SignSshKeys.py is dependent on Python 3.6.0 or later and the requests module.
   >
   > The `SignSshKeys` script writes `user`, `user.pub`, and `user-cert.pub` files into the current working directory, which are later used by the `ssh` command. However, when a public key file is provided to the script, only a `<public_key>` file (with `<public_key>` replaced with the prefix of the public key file passed into the script) is written out to the directory.

5. Run the script to generate the SSH keychain with the `--sfadmin` flag. The script enables SSH across all nodes.

   ```
   SignSshKeys --ip [ip address] --user [username] --duration [hours]
   --sfadmin
   ```

To SSH as `--sfadmin` to a clustered node, you must generate the SSH keychain using a `--user` with `supportAdmin` access on the cluster.

To configure `supportAdmin` access for cluster administrator accounts, you can use the Element UI or APIs:

- Configure "supportAdmin" access using the Element UI
- Configure `supportAdmin` access by using APIs and adding `"supportAdmin"` as the `"access"` type in the API request:
  - Configure "supportAdmin" access for a new account
  - Configure "supportAdmin" access for an existing account

    To get the `clusterAdminID`, you can use the ListClusterAdmins API.

To add `supportAdmin` access, you must have cluster administrator or administrator privileges.

a. Replace the value in [ ] brackets (including the brackets) for each of the following parameters:

   You can use either the abbreviated or full form parameter.

   - **--ip | -i [ip address]**: IP address of the target node for the API to run against.
   - **--user | -u [username]**: Cluster user used to run the API call.
   - **(Optional) --duration | -d [hours]**: The duration a signed key should remain valid as an integer in hours. The default is 24 hours.

b. Compare your input against the following sample command. In this example, `192.168.0.1` is the IP of the storage node, `admin` is the cluster username, duration of key validity is two hours, and `--sfadmin` allows NetApp Support node access for troubleshooting:

```
sh /sf/bin/SignSshKeys.sh --ip 192.168.0.1 --user admin --duration 2
--sfadmin
```

c. Run the command.

6. SSH to the node IPs:

```
ssh -i user sfadmin@[node_ip]
```

7. To close the remote support tunnel, enter the following:

```
rst --killall
```

8. (Optional) Disable remote access functionality again after troubleshooting is complete.

SSH remains enabled on the management node if you do not disable it. SSH enabled configuration persists on the management node through updates and upgrades until it is manually disabled.

Troubleshoot a node that is not part of cluster

You can perform basic troubleshooting of a node that has not yet been added to a cluster. You can use the sfreadonly system account for this purpose with or without the help of NetApp Support. If you have a management node set up, you can use it for SSH and run the script provided for this task.

1. From a Windows, Linux, or Mac machine that has an SSH client installed, run the appropriate script for your system provided by NetApp Support.
2. SSH to the node IP:

```
ssh -i user sfreadonly@[node_ip]
```

3. (Optional) Disable remote access functionality again after troubleshooting is complete.

SSH remains enabled on the management node if you do not disable it. SSH enabled configuration persists on the management node through updates and upgrades until it is manually disabled.

Find more information

- NetApp Element Plug-in for vCenter Server
- NetApp HCI Resources Page

Start a remote NetApp Support session

If you require technical support for your SolidFire all-flash storage system, NetApp Support can connect remotely with your system. To start a session and gain remote access, NetApp Support can open a reverse Secure Shell (SSH) connection to your environment.

You can open a TCP port for an SSH reverse tunnel connection with NetApp Support. This connection enables NetApp Support to log in to your management node.

Before you begin

- For management services 2.18 and later, the capability for remote access is disabled on the management node by default. To enable remote access functionality, see Manage SSH functionality on the management node.
- If your management node is behind a proxy server, the following TCP ports are required in the sshd.config file:

TCP port	Description	Connection direction
443	API calls/HTTPS for reverse port forwarding via open support tunnel to the web UI	Management node to storage nodes
22	SSH login access	Management node to storage nodes or from storage nodes to management node

Steps

- Log in to your management node and open a terminal session.
- At a prompt, enter the following:

```
rst -r sfsupport.solidfire.com -u element -p <port_number>
```

- To close the remote support tunnel, enter the following:

```
rst --killall
```

- (Optional) Disable remote access functionality again.

> SSH remains enabled on the management node if you do not disable it. SSH enabled configuration persists on the management node through updates and upgrades until it is manually disabled.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Manage SSH functionality on the management node

You can disable, re-enable, or determine the status of the SSH capability on the management node (mNode) using the REST API. SSH capability that provides NetApp Support remote support tunnel (RST) session access is disabled by default on management nodes running management services 2.18 or later.

Beginning with Management Services 2.20.69, you can enable and disable SSH capability on the management node using the NetApp Hybrid Cloud Control UI.

What you'll need

- **NetApp Hybrid Cloud Control permissions**: You have permissions as administrator.
- **Cluster administrator permissions**: You have permissions as administrator on the storage cluster.
- **Element software**: Your cluster is running NetApp Element software 11.3 or later.
- **Management node**: You have deployed a management node running version 11.3 or later.
- **Management services updates**:
  - To use the NetApp Hybrid Cloud Control UI, you have updated your management services bundle to

version 2.20.69 or later.

- To use the REST API UI, you have updated your management services bundle to version 2.17.

Options

- Disable or enable the SSH capability on the management node using NetApp Hybrid Cloud Control UI

You can do any of the following tasks after you authenticate:

- Disable or enable the SSH capability on the management node using APIs
- Determine status of the SSH capability on the management node using APIs

Disable or enable the SSH capability on the management node using NetApp Hybrid Cloud Control UI

You can disable or re-enable SSH capability on the management node. SSH capability that provides NetApp Support remote support tunnel (RST) session access is disabled by default on management nodes running management services 2.18 or later. Disabling SSH does not terminate or disconnect existing SSH client sessions to the management node. If you disable SSH and elect to re-enable it at a later time, you can do so using the NetApp Hybrid Cloud Control UI.

To enable or disable support access using SSH for a storage cluster, you must use the Element UI cluster settings page.

Steps

1. From the Dashboard, select the options menu on the top right and select **Configure**.
2. In the **Support Access for Management Node** screen, toggle the switch to enable management node SSH.
3. After you complete troubleshooting, in the **Support Access for Management Node** screen, toggle the switch to disable management node SSH.

Disable or enable the SSH capability on the management node using APIs

You can disable or re-enable SSH capability on the management node. SSH capability that provides NetApp Support remote support tunnel (RST) session access is disabled by default on management nodes running management services 2.18 or later. Disabling SSH does not terminate or disconnect existing SSH client sessions to the management node. If you disable SSH and elect to re-enable it at a later time, you can do so using the same API.

API command

For management services 2.18 or later:

```
curl -k -X PUT
"https://<<ManagementNodeIP>/mnode/2/settings/ssh?enabled=<false/true>" -H
"accept: application/json" -H  "Authorization: Bearer ${TOKEN}"
```

For management services 2.17 or earlier:

```
curl -X PUT
"https://<ManagementNodeIP>/mnode/settings/ssh?enabled=<false/true>" -H
"accept: application/json" -H  "Authorization: Bearer ${TOKEN}"
```

You can find the bearer `${TOKEN}` used by the API command when you authorize. The bearer `${TOKEN}` is in the curl response.

REST API UI steps

1. Access the REST API UI for the management node API service by entering the management node IP address followed by `/mnode/`:

   ```
   https://<ManagementNodeIP>/mnode/
   ```

2. Select **Authorize** and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. From the REST API UI, select **PUT /settings/ssh**.
   a. Select **Try it out**.
   b. Set the **enabled** parameter to `false` to disable SSH or `true` to re-enable SSH capability that was previously disabled.
   c. Select **Execute**.

Determine status of the SSH capability on the management node using APIs

You can determine whether or not SSH capability is enabled on the management node using a management node service API. SSH is disabled by default on management nodes running management services 2.18 or later.

API command

For management services 2.18 or later:

```
curl -k -X PUT
"https://<<ManagementNodeIP>/mnode/2/settings/ssh?enabled=<false/true>" -H
"accept: application/json" -H  "Authorization: Bearer ${TOKEN}"
```

For management services 2.17 or earlier:

```
curl -X PUT
"https://<ManagementNodeIP>/mnode/settings/ssh?enabled=<false/true>" -H
"accept: application/json" -H  "Authorization: Bearer ${TOKEN}"
```

You can find the bearer `${TOKEN}` used by the API command when you authorize. The bearer `${TOKEN}` is in the curl response..

REST API UI steps

1. Access the REST API UI for the management node API service by entering the management node IP address followed by `/mnode/`:

   ```
   https://<ManagementNodeIP>/mnode/
   ```

2. Select **Authorize** and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client`.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. From the REST API UI, select **GET /settings/ssh**.
   a. Select **Try it out**.
   b. Select **Execute**.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

Upgrade your NetApp SolidFire all-flash storage system

Upgrade sequence overview

You can keep your SolidFire Element storage system up-to-date after deployment by sequentially upgrading all NetApp storage components.

These components include management services, NetApp Hybrid Cloud Control, Element software, management node, and (depending on your installation) the Element Plug-in for vCenter Server.

- Beginning November 2023, you cannot start a component upgrade using NetApp Hybrid Cloud Control or REST API because the signing key certificates (private and public) expired on November 5, 2023. You can resolve this issue by following the workaround documented in the Knowledge Base article SolidFire and HCI upgrades unable to start due to upgrade package upload error.
- Beginning with Element 12.7, the SF2405 and SF9608 storage nodes and FC0025 and SF-FCN-01 FC nodes are not supported. If you attempt to upgrade one of these nodes to Element 12.7, you will see an error stating that this node is not supported by Element 12.7.
- Beginning with Element 12.5, NetApp HealthTools is no longer supported for Element software upgrades. If you are running Element 11.0 or 11.1, you must first upgrade to Element 12.3 using HealthTools and then upgrade to Element 12.5 or later using NetApp Hybrid Cloud Control.

The system upgrade sequence content describes the tasks that are needed to complete a SolidFire all-flash storage system upgrade. Ideally these procedures are performed as part of the larger upgrade sequence and not in isolation. If a component-based upgrade or update is needed, see the procedure prerequisites to ensure additional complexities are addressed.

The vSphere upgrade sequence including Element Plug-in for vCenter Server content describes additional pre- and post-upgrade steps required to re-install the Element Plug-in for vCenter Server.

What you'll need

- You are running management node 11.3 or later. Newer versions of the management node have a modular architecture that provides individual services.

  To check the version, log in to your management node and view the Element version number in the login banner. If you do not have 11.3, see Upgrade your management node.

- You have upgraded your management services to at least version 2.1.326.

  Upgrades using NetApp Hybrid Cloud Control are not available in earlier service bundle versions.

- You have ensured that the system time on all nodes is synced and that NTP is correctly configured for the storage cluster and nodes. Each node must be configured with a DNS nameserver in the per-node web UI (`https://[IP address]:442`) with no unresolved cluster faults related to time skew.
- You have scheduled sufficient time for your Element software and storage firmware upgrades. When you upgrade to Element software 12.5 or later, the upgrade process time varies depending on the Element software version and firmware updates.

System upgrade sequence

You can use the following sequence to upgrade your NetApp SolidFire all-flash storage system for Element 12.5 or later.

Steps

1. Update management services from Hybrid Cloud Control.

   If you are updating management services to version 2.16 or later and you are running a management node 11.3 to 11.8, you will need to increase your management node VM's RAM prior to updating management services.

   You must update to the latest management services bundle before upgrading your Element software.

2. Run Element storage health checks prior to upgrading storage.
3. Upgrade your Element software and storage firmware.
4. (Optional) Upgrade your Element storage firmware only.

   You might perform this task when a new storage firmware upgrade becomes available outside of a major release.

5. (Optional) Upgrade your management node.

   Upgrading the management node operating system is no longer required to upgrade Element software on the storage cluster. If the management node is version 11.3 or higher, you can simply upgrade the management services to the latest version to perform Element upgrades using NetApp Hybrid Cloud Control. Follow the management node upgrade procedure for your scenario if you would like to upgrade the management node operating system for other reasons, such as security remediation.

6. Upgrade your Element Plug-in for vCenter Server.

Find more information

- NetApp Element Plug-in for vCenter Server
- SolidFire and Element Software Documentation

System upgrade procedures

Update management services

You can update your management services to the latest bundle version after you have installed management node 11.3 or later.

Beginning with the Element 11.3 management node release, the management node design has been changed based on a new modular architecture that provides individual services. These modular services provide central and extended management functionality for a SolidFire all-flash storage system. Management services include system telemetry, logging, and update services, the QoSSIOC service for Element Plug-in for vCenter Server,

NetApp Hybrid Cloud Control, and more.

About this task

- You must upgrade to the latest management services bundle before upgrading your Element software.

- Management services 2.22.7 includes Element Plug-in for vCenter Server 5.0 which contains the remote plug-in. If you use the Element plug-in, you should upgrade to management services 2.22.7 or later to comply with the VMware directive that removes support for local plug-ins. Learn more.
- For the latest management services release notes describing major services, new features, bug fixes, and workarounds for each service bundle, see the management services release notes

What you'll need

Beginning with management services 2.20.69, you must accept and save the End User License Agreement (EULA) before using the NetApp Hybrid Cloud Control UI or API to upgrade management services:

1. Open the IP address of the management node in a web browser:

```
https://<ManagementNodeIP>
```

2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
3. Select **Upgrade** near the top right of the interface.
4. The EULA pops up. Scroll down, select **I accept for current and all future updates**, and select **Save**.

Update options

You can update management services using the NetApp Hybrid Cloud Control UI or the management node REST API:

- Update management services using Hybrid Cloud Control (Recommended method)
- Update management services using the management node API

Update management services using Hybrid Cloud Control

You can update your NetApp management services using NetApp Hybrid Cloud Control.

Management service bundles provide enhanced functionality and fixes to your installation outside of major releases.

Before you begin

- You are running management node 11.3 or later.
- If you are updating management services to version 2.16 or later and you are running a management node 11.3 to 11.8, you will need to increase your management node VM's RAM prior to updating management services:
  a. Power off the management node VM.
  b. Change the RAM of the management node VM from 12GB to 24GB RAM.
  c. Power on the management node VM.

- Your cluster version is running NetApp Element software 11.3 or later.
- You have upgraded your management services to at least version 2.1.326. NetApp Hybrid Cloud Control upgrades are not available in earlier service bundles.

> For a list of available services for each service bundle version, see the Management Services Release Notes.

Steps

1. Open the IP address of the management node in a web browser:

   ```
   https://<ManagementNodeIP>
   ```

2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
3. Select **Upgrade** near the top right of the interface.
4. On the Upgrades page, select the **Management Services** tab.
5. Follow the instructions on the page to download and save a management services upgrade package to your computer.
6. Select **Browse** to locate the package you saved and upload it.

   After you upload the package, the upgrade starts automatically.

   After the upgrade begins, you can see the upgrade status on this page. During the upgrade, you might lose connection with NetApp Hybrid Cloud Control and have to log back in to see the results of the upgrade.

Update management services using the management node API

Users should ideally perform management services updates from NetApp Hybrid Cloud Control. You can however manually upload, extract, and deploy a service bundle update for management services to the management node using the REST API. You can run each command from the REST API UI for the management node.

Before you begin

- You have deployed a NetApp Element software management node 11.3 or later.
- If you are updating management services to version 2.16 or later and you are running a management node 11.3 to 11.8, you will need to increase your management node VM's RAM prior to updating management services:
  a. Power off the management node VM.
  b. Change the RAM of the management node VM from 12GB to 24GB RAM.
  c. Power on the management node VM.
- Your cluster version is running NetApp Element software 11.3 or later.
- You have upgraded your management services to at least version 2.1.326. NetApp Hybrid Cloud Control upgrades are not available in earlier service bundles.

> For a list of available services for each service bundle version, see the Management Services Release Notes.

Steps

1. Open the REST API UI on the management node: `https://<ManagementNodeIP>/mnode`
2. Select **Authorize** and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client` if the value is not already populated.
   c. Select **Authorize** to begin a session.
   d. Close the window.
3. Upload and extract the service bundle on the management node using this command: `PUT /services/upload`
4. Deploy the management services on the management node: `PUT /services/deploy`
5. Monitor the status of the update: `GET /services/update/status`

   A successful update returns a result similar to the following example:

```
{
"current_version": "2.10.29",
"details": "Updated to version 2.17.52",
"status": "success"
}
```

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Run Element storage health checks prior to upgrading storage

You must run health checks prior to upgrading Element storage to ensure all storage nodes in your cluster are ready for the next Element storage upgrade.

What you'll need

- **Management services**: You have updated to the latest management services bundle (2.10.27 or later).

  You must upgrade to the latest management services bundle before upgrading your Element software.

- **Management node**:You are running management node 11.3 or later.
- **Element software**: Your cluster version is running NetApp Element software 11.3 or later.
- **End User License Agreement (EULA)**: Beginning with management services 2.20.69, you must accept and save the EULA before using the NetApp Hybrid Cloud Control UI or API to run Element storage health checks:
  1. Open the IP address of the management node in a web browser:

```
https://<ManagementNodeIP>
```

2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
3. Select **Upgrade** near the top right of the interface.
4. The EULA pops up. Scroll down, select **I accept for current and all future updates**, and select **Save**.

Health check options

You can run health checks using the NetApp Hybrid Cloud Control UI or the NetApp Hybrid Cloud Control API:

- Use NetApp Hybrid Cloud Control to run Element storage health checks prior to upgrading storage (Preferred method)

You can also find out more about storage health checks that are run by the service:

- Storage health checks made by the service

Use NetApp Hybrid Cloud Control to run Element storage health checks prior to upgrading storage

Using NetApp Hybrid Cloud Control, you can verify that a storage cluster is ready to be upgraded.

Steps

1. Open the IP address of the management node in a web browser:

```
https://<ManagementNodeIP>
```

2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
3. Select **Upgrade** near the top right of the interface.
4. On the **Upgrades** page, select the **Storage** tab.
5. Select the health check  for the cluster you want to check for upgrade readiness.
6. On the **Storage Health Check** page, select **Run Health Check**.
7. If there are issues, do the following:
   a. Go to the specific KB article listed for each issue or perform the specified remedy.
   b. If a KB is specified, complete the process described in the relevant KB article.
   c. After you have resolved cluster issues, select **Re-Run Health Check**.

After the health check completes without errors, the storage cluster is ready to upgrade. See storage node upgrade instructions to proceed.

Use API to run Element storage health checks prior to upgrading storage

You can use REST API to verify that a storage cluster is ready to be upgraded. The health check verifies that there are no obstacles to upgrading, such as pending nodes, disk space issues, and cluster faults.

Steps

1. Locate the storage cluster ID:
   a. Open the management node REST API UI on the management node:

      ```
      https://<ManagementNodeIP>/mnode
      ```

   b. Select **Authorize** and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client` if the value is not already populated.
      iii. Select **Authorize** to begin a session.
      iv. Close the authorization window.
   c. From the REST API UI, select `GET /assets`.
   d. Select **Try it out**.
   e. Select **Execute**.
   f. From the response, copy the `"id"` from the `"storage"` section of the cluster you intend to check for upgrade readiness.

      Do not use the `"parent"` value in this section because this is the management node's ID, not the storage cluster's ID.

      ```
      "config": {},
      "credentialid": "12bbb2b2-f1be-123b-1234-12c3d4bc123e",
      "host_name": "SF_DEMO",
      "id": "12cc3a45-e6e7-8d91-a2bb-0bdb3456b789",
      "ip": "10.123.12.12",
      "parent": "d123ec42-456e-8912-ad3e-4bd56f4a789a",
      "sshcredentialid": null,
      "ssl_certificate": null
      ```

2. Run health checks on the storage cluster:
   a. Open the storage REST API UI on the management node:

      ```
      https://<ManagementNodeIP>/storage/1/
      ```

   b. Select **Authorize** and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client` if the value is not already populated.
      iii. Select **Authorize** to begin a session.
      iv. Close the authorization window.
   c. Select **POST /health-checks**.

d. Select **Try it out**.

e. In the parameter field, enter the storage cluster ID obtained in Step 1.

```
{
  "config": {},
  "storageId": "123a45b6-1a2b-12a3-1234-1a2b34c567d8"
}
```

f. Select **Execute** to run a health check on the specified storage cluster.

The response should indicate state as `initializing`:

```
{
  "_links": {
    "collection": "https://10.117.149.231/storage/1/health-checks",
    "log": "https://10.117.149.231/storage/1/health-checks/358f073f-
896e-4751-ab7b-ccbb5f61f9fc/log",
    "self": "https://10.117.149.231/storage/1/health-checks/358f073f-
896e-4751-ab7b-ccbb5f61f9fc"
  },
  "config": {},
  "dateCompleted": null,
  "dateCreated": "2020-02-21T22:11:15.476937+00:00",
  "healthCheckId": "358f073f-896e-4751-ab7b-ccbb5f61f9fc",
  "state": "initializing",
  "status": null,
  "storageId": "c6d124b2-396a-4417-8a47-df10d647f4ab",
  "taskId": "73f4df64-bda5-42c1-9074-b4e7843dbb77"
}
```

g. Copy the `healthCheckID` that is part of response.

3. Verify the results of the health checks:

a. Select **GET /health-checks/{healthCheckId}**.

b. Select **Try it out**.

c. Enter the health check ID in the parameter field.

d. Select **Execute**.

e. Scroll to the bottom of the response body.

If all health checks are successful, the return is similar to the following example:

```
"message": "All checks completed successfully.",
"percent": 100,
"timestamp": "2020-03-06T00:03:16.321621Z"
```

4. If the `message` return indicates that there were problems regarding cluster health, do the following:
   a. Select **GET /health-checks/{healthCheckId}/log**
   b. Select **Try it out**.
   c. Enter the health check ID in the parameter field.
   d. Select **Execute**.
   e. Review any specific errors and obtain their associated KB article links.
   f. Go to the specific KB article listed for each issue or perform the specified remedy.
   g. If a KB is specified, complete the process described in the relevant KB article.
   h. After you have resolved cluster issues, run **GET /health-checks/{healthCheckId}/log** again.

Storage health checks made by the service

Storage health checks make the following checks per cluster.

Check Name	Node/Cluster	Description
check_async_results	Cluster	Verifies that the number of asynchronous results in the database is below a threshold number.
check_cluster_faults	Cluster	Verifies that there are no upgrade blocking cluster faults (as defined in Element source).
check_upload_speed	Node	Measures the upload speed between the storage node and the management node.
connection_speed_check	Node	Verifies that nodes have connectivity to the management node serving upgrade packages and estimates connection speed.
check_cores	Node	Checks for kernel crash dump and core files on the node. The check fails for any crashes in a recent time period (threshold 7 days).
check_root_disk_space	Node	Verifies the root file system has sufficient free space to perform an upgrade.

Check Name	Node/Cluster	Description
check_var_log_disk_space	Node	Verifies that `/var/log` free space meets some percentage free threshold. If it does not, the check will rotate and purge older logs in order to fall under threshold. The check fails if it is unsuccessful at creating sufficient free space.
check_pending_nodes	Cluster	Verifies that there are no pending nodes on the cluster.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Upgrade Element software

To upgrade NetApp Element software, you can use the NetApp Hybrid Cloud Control UI or REST API. Certain operations are suppressed during an Element software upgrade, such as adding and removing nodes, adding and removing drives, and commands associated with initiators, volume access groups, and virtual networks, among others.

Beginning with Element 12.5, NetApp HealthTools is no longer supported for Element software upgrades. If you are running Element 11.0 or 11.1, you must first upgrade to Element 12.3.x using HealthTools and then upgrade to Element 12.5 or later using NetApp Hybrid Cloud Control.

What you'll need

- **Admin privileges**: You have storage cluster administrator permissions to perform the upgrade.
- **Valid upgrade path**: You have checked upgrade path information for the Element version you are upgrading to and verified that the upgrade path is valid.
  NetApp KB: Upgrade matrix for storage clusters running NetApp Element Software
- **System time sync**: You have ensured that the system time on all nodes is synced and that NTP is correctly configured for the storage cluster and nodes. Each node must be configured with a DNS nameserver in the per-node web UI (`https://[IP address]:442`) with no unresolved cluster faults related to time skew.
- **System ports**: If you are using NetApp Hybrid Cloud Control for upgrades, you have ensured that the necessary ports are open. See Network ports for more information.
- **Management node**: For NetApp Hybrid Cloud Control UI and API, the management node in your environment is running version 11.3.
- **Management services**: You have updated your management services bundle to the latest version.

You must upgrade to the latest management services bundle before upgrading your Element software to version 12.5 or later. If you are updating your Element software to version 12.5 or later, you need management services 2.21.61 or later to proceed.

- **Cluster health**: You have verified that the cluster is ready to be upgraded. See Run Element storage health checks prior to upgrading storage.
- **Updated baseboard management controller (BMC) for H610S storage nodes**: You have upgraded the BMC version for your H610S nodes. See the release notes and upgrade instructions.
- **Upgrade process time**: You have scheduled sufficient time to perform your upgrade. When you upgrade to Element software 12.5 or later, the upgrade process time varies depending on your current Element software version and firmware updates.

Storage Node	Current Element software version	Approximate software and firmware install time per node[1]	Approximate data synchronization time per node[2]	Approximate total upgrade time per node
All SolidFire and NetApp H-series nodes with up-to-date firmware[3]	12.x	15 minutes	10 to 15 minutes	20 to 30 minutes
H610S and H410S	12.x and 11.8	60 minutes	30 to 60 minutes	90 to 120 minutes
H610S	11.7 and earlier	90 minutes	40 to 70 minutes	130 to 160 minutes You must also perform a complete node shutdown and power disconnect for each H610S node.

[1]For a complete matrix of firmware and driver firmware for your hardware, see supported storage firmware versions for SolidFire storage nodes.

[2]If you combine a cluster with a heavy write IOPS load with a longer firmware update time, the data synchronization time will increase.

[3]Beginning with Element 12.7, the SF2405 and SF9608 storage nodes and FC0025 and SF-FCN-01 FC nodes are not supported. If you attempt to upgrade one of these nodes to Element 12.7, you will see an error stating that this node is not supported by Element 12.7.

- **End User License Agreement (EULA)**: Beginning with management services 2.20.69, you must accept and save the EULA before using the NetApp Hybrid Cloud Control UI or API to upgrade Element software:
  1. Open the IP address of the management node in a web browser:

     ```
     https://<ManagementNodeIP>
     ```

  2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
  3. Select **Upgrade** near the top right of the interface.
  4. The EULA pops up. Scroll down, select **I accept for current and all future updates**, and select **Save**.

Upgrade options

Choose one of the following Element software upgrade options:

- Use NetApp Hybrid Cloud Control UI to upgrade Element storage
- Use NetApp Hybrid Cloud Control API to upgrade Element storage

If you are upgrading an H610S series node to Element 12.5 or later and the node is running a version of Element earlier than 11.8, you will need to perform the additional upgrade steps in this KB article for each storage node. If you are running Element 11.8 or later, the additional upgrade steps are not required.

Use NetApp Hybrid Cloud Control UI to upgrade Element storage

Using the NetApp Hybrid Cloud Control UI, you can upgrade a storage cluster.

For potential issues while upgrading storage clusters using NetApp Hybrid Cloud Control and their workarounds, see this KB article.

Steps

1. Open the IP address of the management node in a web browser:

   ```
   https://<ManagementNodeIP>
   ```

2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
3. Select **Upgrade** near the top right of the interface.
4. On the **Upgrades** page, select **Storage**.

   The **Storage** tab lists the storage clusters that are part of your installation. If a cluster is inaccessible by NetApp Hybrid Cloud Control, it will not be displayed on the **Upgrades** page.

5. Choose from the following options and perform the set of steps that are applicable to your cluster:

Option	Steps
All clusters running Element 11.8 and later	1. Select **Browse** to upload the upgrade package that you downloaded. 2. Wait for the upload to complete. A progress bar shows the status of the upload. The file upload will be lost if you navigate away from the browser window. An on-screen message is displayed after the file is successfully uploaded and validated. Validation might take several minutes. If you navigate away from the browser window at this stage, the file upload is preserved. 3. Select **Begin Upgrade**. The **Upgrade Status** changes during the upgrade to reflect the status of the process. It also changes in response to actions you take, such as pausing the upgrade, or if the upgrade returns an error. See Upgrade status changes. While the upgrade is in progress, you can leave the page and come back to it later to continue monitoring the progress. The page does not dynamically update status and current version if the cluster row is collapsed. The cluster row must be expanded to update the table or you can refresh the page. You can download logs after the upgrade is complete.

Option	Steps
You are upgrading an H610S cluster running Element version earlier than 11.8.	1. Select the drop-down arrow next to the cluster you are upgrading, and select from the upgrade versions available. 2. Select **Begin Upgrade**. After the upgrade is complete, the UI prompts you to perform additional upgrade steps. 3. Complete the additional steps required in the KB article, and acknowledge in the UI that you have completed phase 2. You can download logs after the upgrade is complete. For information about the various upgrade status changes, see Upgrade status changes.

Upgrade status changes

Here are the different states that the **Upgrade Status** column in the UI shows before, during, and after the upgrade process:

Upgrade state	Description
Up to Date	The cluster was upgraded to the latest Element version available.
Versions Available	Newer versions of Element and/or storage firmware are available for upgrade.
In Progress	The upgrade is in progress. A progress bar shows the upgrade status. On-screen messages also show node-level faults and display the node ID of each node in the cluster as the upgrade progresses. You can monitor the status of each node using the Element UI or the NetApp Element plug-in for vCenter Server UI.
Upgrade Pausing	You can choose to pause the upgrade. Depending on the state of the upgrade process, the pause operation can succeed or fail. You will see a UI prompt asking you to confirm the pause operation. To ensure that the cluster is in a safe spot before pausing an upgrade, it can take up to two hours for the upgrade operation to be completely paused. To resume the upgrade, select **Resume**.
Paused	You paused the upgrade. Select **Resume** to resume the process.

Upgrade state	Description
Error	An error has occurred during the upgrade. You can download the error log and send it to NetApp Support. After you resolve the error, you can return to the page, and select **Resume**. When you resume the upgrade, the progress bar goes backwards for a few minutes while the system runs the health check and checks the current state of the upgrade.
Complete with Follow-up	Only for H610S nodes upgrading from Element version earlier than 11.8. After phase 1 of the upgrade process is complete, this state prompts you to perform additional upgrade steps (see the KB article). After you complete phase 2 and acknowledge that you have completed it, the status changes to **Up to Date**.

Use NetApp Hybrid Cloud Control API to upgrade Element storage

You can use APIs to upgrade storage nodes in a cluster to the latest Element software version. You can use an automation tool of your choice to run the APIs. The API workflow documented here uses the REST API UI available on the management node as an example.

Steps

1. Download the storage upgrade package to a device that is accessible to the management node.

   Go to the Element software downloads page and download the latest storage node image.

2. Upload the storage upgrade package to the management node:
   a. Open the management node REST API UI on the management node:

      ```
      https://<ManagementNodeIP>/package-repository/1/
      ```

   b. Select **Authorize** and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client`.
      iii. Select **Authorize** to begin a session.
      iv. Close the authorization window.
   c. From the REST API UI, select **POST /packages**.
   d. Select **Try it out**.
   e. Select **Browse** and select the upgrade package.
   f. Select **Execute** to initiate the upload.
   g. From the response, copy and save the package ID (`"id"`) for use in a later step.
3. Verify the status of the upload.
   a. From the REST API UI, select **GET /packages/{id}/status**.
   b. Select **Try it out**.

c. Enter the package ID you copied in the previous step in **id**.

d. Select **Execute** to initiate the status request.

The response indicates `state` as `SUCCESS` when complete.

4. Locate the storage cluster ID:

a. Open the management node REST API UI on the management node:

```
https://<ManagementNodeIP>/inventory/1/
```

b. Select **Authorize** and complete the following:

i. Enter the cluster user name and password.

ii. Enter the client ID as `mnode-client`.

iii. Select **Authorize** to begin a session.

iv. Close the authorization window.

c. From the REST API UI, select **GET /installations**.

d. Select **Try it out**.

e. Select **Execute**.

f. From the response, copy the installation asset ID (`"id"`).

g. From the REST API UI, select **GET /installations/{id}**.

h. Select **Try it out**.

i. Paste the installation asset ID into the **id** field.

j. Select **Execute**.

k. From the response, copy and save the storage cluster ID (`"id"`) of the cluster you intend to upgrade for use in a later step.

5. Run the storage upgrade:

a. Open the storage REST API UI on the management node:

```
https://<ManagementNodeIP>/storage/1/
```

b. Select **Authorize** and complete the following:

i. Enter the cluster user name and password.

ii. Enter the client ID as `mnode-client`.

iii. Select **Authorize** to begin a session.

iv. Close the authorization window.

c. Select **POST /upgrades**.

d. Select **Try it out**.

e. Enter the upgrade package ID in the parameter field.

f. Enter the storage cluster ID in the parameter field.

The payload should look similar to the following example:

```
{
  "config": {},
  "packageId": "884f14a4-5a2a-11e9-9088-6c0b84e211c4",
  "storageId": "884f14a4-5a2a-11e9-9088-6c0b84e211c4"
}
```

g. Select **Execute** to initiate the upgrade.

The response should indicate the state as `initializing`:

```
{
  "_links": {
    "collection": "https://localhost:442/storage/upgrades",
    "self": "https://localhost:442/storage/upgrades/3fa85f64-1111-
4562-b3fc-2c963f66abc1",
    "log": https://localhost:442/storage/upgrades/3fa85f64-1111-4562-
b3fc-2c963f66abc1/log
  },
  "storageId": "114f14a4-1a1a-11e9-9088-6c0b84e200b4",
  "upgradeId": "334f14a4-1a1a-11e9-1055`-6c0b84e2001b4",
  "packageId": "774f14a4-1a1a-11e9-8888-6c0b84e200b4",
  "config": {},
  "state": "initializing",
  "status": {
    "availableActions": [
      "string"
    ],
    "message": "string",
    "nodeDetails": [
      {
        "message": "string",
        "step": "NodePreStart",
        "nodeID": 0,
        "numAttempt": 0
      }
    ],
    "percent": 0,
    "step": "ClusterPreStart",
    "timestamp": "2020-04-21T22:10:57.057Z",
    "failedHealthChecks": [
      {
```

```
        "checkID": 0,
        "name": "string",
        "displayName": "string",
        "passed": true,
        "kb": "string",
        "description": "string",
        "remedy": "string",
        "severity": "string",
        "data": {},
        "nodeID": 0
      }
    ]
  },
  "taskId": "123f14a4-1a1a-11e9-7777-6c0b84e123b2",
  "dateCompleted": "2020-04-21T22:10:57.057Z",
  "dateCreated": "2020-04-21T22:10:57.057Z"
}
```


h. Copy the upgrade ID (`"upgradeId"`) that is part of the response.

6. Verify the upgrade progress and results:
   a. Select **GET /upgrades/{upgradeId}**.
   b. Select **Try it out**.
   c. Enter the upgrade ID from the previous step in **upgradeId**.
   d. Select **Execute**.
   e. Do one of the following if there are problems or special requirements during the upgrade:

Option	Steps
You need to correct cluster health issues due to `failedHealthChecks` message in the response body.	1. Go to the specific KB article listed for each issue or perform the specified remedy. 2. If a KB is specified, complete the process described in the relevant KB article. 3. After you have resolved cluster issues, reauthenticate if needed and select **PUT /upgrades/{upgradeId}**. 4. Select **Try it out**. 5. Enter the upgrade ID from the previous step in **upgradeId**. 6. Enter `"action":"resume"` in the request body. `{ "action": "resume" }` 7. Select **Execute**.
You need to pause the upgrade because the maintenance window is closing or for another reason.	1. Reauthenticate if needed and select **PUT /upgrades/{upgradeId}**. 2. Select **Try it out**. 3. Enter the upgrade ID from the previous step in **upgradeId**. 4. Enter `"action":"pause"` in the request body. `{ "action": "pause" }` 5. Select **Execute**.

Option	Steps
If you are upgrading an H610S cluster running an Element version earlier than 11.8, you see the state `finishedNeedsAck` in the response body.You must perform additional upgrade steps for each H610S storage node.	1. Complete the additional upgrade steps in this KB article for each node. 2. Reauthenticate if needed and select **PUT /upgrades/{upgradeId}**. 3. Select **Try it out**. 4. Enter the upgrade ID from the previous step in **upgradeId**. 5. Enter `"action":"acknowledge"` in the request body. `{ "action": "acknowledge" }` 6. Select **Execute**.

f. Run the **GET /upgrades/{upgradeId}** API multiple times, as needed, until the process is complete.

During the upgrade, the `status` indicates `running` if no errors are encountered. As each node is upgraded, the `step` value changes to `NodeFinished`.

The upgrade has finished successfully when the `percent` value is `100` and the `state` indicates `finished`.

What happens if an upgrade fails using NetApp Hybrid Cloud Control

If a drive or node fails during an upgrade, the Element UI will show cluster faults. The upgrade process does not proceed to the next node, and waits for the cluster faults to resolve. The progress bar in the UI shows that the upgrade is waiting for the cluster faults to resolve. At this stage, selecting **Pause** in the UI will not work, because the upgrade waits for the cluster to be healthy. You will need to engage NetApp Support to assist with the failure investigation.

NetApp Hybrid Cloud Control has a pre-set three-hour waiting period, during which one of the following scenarios can happen:

- The cluster faults get resolved within the three-hour window, and upgrade resumes. You do not need to take any action in this scenario.
- The problem persists after three hours, and the upgrade status shows **Error** with a red banner. You can resume the upgrade by selecting **Resume** after the problem is resolved.
- NetApp Support has determined that the upgrade needs to be temporarily aborted to take corrective action before the three-hour window. Support will use the API to abort the upgrade.

Aborting the cluster upgrade while a node is being updated might result in the drives being ungracefully removed from the node. If the drives are ungracefully removed, adding the drives back during an upgrade will require manual intervention by NetApp Support. The node might be taking longer to do firmware updates or post update syncing activities. If the upgrade progress seems stalled, contact NetApp Support for assistance.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Upgrade storage firmware

Starting with Element 12.0 and management services version 2.14, you can perform firmware-only upgrades on your storage nodes using the NetApp Hybrid Cloud Control UI and REST API. This procedure does not upgrade Element software and enables you to upgrade storage firmware outside of a major Element release.

What you'll need

- **Admin privileges**: You have storage cluster administrator permissions to perform the upgrade.
- **System time sync**: You have ensured that the system time on all nodes is synced and that NTP is correctly configured for the storage cluster and nodes. Each node must be configured with a DNS nameserver in the per-node web UI (`https://[IP address]:442`) with no unresolved cluster faults related to time skew.
- **System ports**: If you are using NetApp Hybrid Cloud Control for upgrades, you have ensured that the necessary ports are open. See Network ports for more information.
- **Management node**: For NetApp Hybrid Cloud Control UI and API, the management node in your environment is running version 11.3.
- **Management services**: You have updated your management services bundle to the latest version.

For H610S storage nodes running Element software version 12.0, you should apply D-patch SUST-909 before you upgrade to storage firmware bundle 2.27. Contact NetApp Support to obtain the D-patch before you upgrade. See Storage Firmware Bundle 2.27 Release Notes.

You must upgrade to the latest management services bundle before upgrading the firmware on your storage nodes. If you are updating your Element software to version 12.2 or later, you need management services 2.14.60 or later to proceed.

- **Cluster health**: You have run health checks. See Run Element storage health checks prior to upgrading storage.
- **Updated baseboard management controller (BMC) for H610S nodes**: You have upgraded the BMC version for your H610S nodes. See release notes and upgrade instructions.

For a complete matrix of firmware and driver firmware for your hardware, see supported storage firmware versions for SolidFire storage nodes.

- **Upgrade process time**: You have scheduled sufficient time to perform your upgrade. When you upgrade to Element software 12.5 or later, the upgrade process time varies depending on your current Element

software version and firmware updates.

Storage Node	Current Element software version	Approximate software and firmware install time per node[1]	Approximate data synchronization time per node[2]	Approximate total upgrade time per node
All SolidFire and NetApp H-series nodes with up-to-date firmware[3]	12.x	15 minutes	10 to 15 minutes	20 to 30 minutes
H610S and H410S	12.x and 11.8	60 minutes	30 to 60 minutes	90 to 120 minutes
H610S	11.7 and earlier	90 minutes	40 to 70 minutes	130 to 160 minutes You must also perform a complete node shutdown and power disconnect for each H610S node.

[1]For a complete matrix of firmware and driver firmware for your hardware, see supported storage firmware versions for SolidFire storage nodes.

[2]If you combine a cluster with a heavy write IOPS load with a longer firmware update time, the data synchronization time will increase.

[3]Beginning with Element 12.7, the SF2405 and SF9608 storage nodes and FC0025 and SF-FCN-01 FC nodes are not supported. If you attempt to upgrade one of these nodes to Element 12.7, you will see an error stating that this node is not supported by Element 12.7.

- **End User License Agreement (EULA)**: Beginning with management services 2.20.69, you must accept and save the EULA before using the NetApp Hybrid Cloud Control UI or API to upgrade storage firmware:
  1. Open the IP address of the management node in a web browser:

     ```
     https://<ManagementNodeIP>
     ```

  2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
  3. Select **Upgrade** near the top right of the interface.
  4. The EULA pops up. Scroll down, select **I accept for current and all future updates**, and select **Save**.

Upgrade options

Choose one of the following storage firmware upgrade options:

- Use NetApp Hybrid Cloud Control UI to upgrade storage firmware
- Use NetApp Hybrid Cloud Control API to upgrade storage firmware

Use NetApp Hybrid Cloud Control UI to upgrade storage firmware

You can use the NetApp Hybrid Cloud Control UI to upgrade the firmware of the storage nodes in your cluster.

What you'll need

- If your management node is not connected to the internet, you have downloaded the storage firmware bundle.

For potential issues while upgrading storage clusters using NetApp Hybrid Cloud Control and their workarounds, see the KB article.

The upgrade process takes approximately 30 minutes per storage node. If you are upgrading an Element storage cluster to storage firmware newer than version 2.76, individual storage nodes will only reboot during the upgrade if new firmware was written to the node.

Steps

1. Open the IP address of the management node in a web browser:

```
https://<ManagementNodeIP>
```

2. Log in to NetApp Hybrid Cloud Control by providing the storage cluster administrator credentials.
3. Select **Upgrade** near the top right of the interface.
4. On the **Upgrades** page, select **Storage**.

The **Storage** tab lists the storage clusters that are part of your installation. If a cluster is inaccessible by NetApp Hybrid Cloud Control, it will not be displayed on the **Upgrades** page. If you have clusters running Element 12.0 or later, you will see the current firmware bundle version listed for these clusters. If the nodes in a single cluster have different firmware versions on them or as the upgrade progresses, you will see **Multiple** in the **Current Firmware Bundle Version** column. You can select **Multiple** to navigate to the **Nodes** page to compare firmware versions. If all your clusters are running Element versions earlier than 12.0, you will not see any information about firmware bundle version numbers.

If the cluster is up to date and/or no upgrade packages are available, the **Element** and **Firmware Only** tabs are not displayed. These tabs are also not displayed when an upgrade is in progress. If the **Element** tab is displayed, but not the **Firmware Only** tab, no firmware packages are available.

5. Select the drop-down arrow next to the cluster you are upgrading.
6. Select **Browse** to upload the upgrade package that you downloaded.
7. Wait for the upload to complete. A progress bar shows the status of the upload.

The file upload will be lost if you navigate away from the browser window.

An on-screen message is displayed after the file is successfully uploaded and validated. Validation might take several minutes. If you navigate away from the browser window at this stage, the file upload is preserved.

8. Select **Firmware Only**, and select from the upgrade versions available.
9. Select **Begin Upgrade**.

The **Upgrade Status** changes during the upgrade to reflect the status of the process. It also changes in response to actions you take, such as pausing the upgrade, or if the upgrade returns an error. See Upgrade status changes.

While the upgrade is in progress, you can leave the page and come back to it later to continue monitoring the progress. The page does not dynamically update status and current version if the cluster row is collapsed. The cluster row must be expanded to update the table or you can refresh the page.

You can download logs after the upgrade is complete.

Upgrade status changes

Here are the different states that the **Upgrade Status** column in the UI shows before, during, and after the upgrade process:

Upgrade state	Description
Up to Date	The cluster was upgraded to the latest Element version available or the firmware was upgraded to the latest version.
Unable to Detect	This status is displayed when the storage service API returns an upgrade status that is not in the enumerated list of possible upgrade statuses.
Versions Available	Newer versions of Element and/or storage firmware are available for upgrade.
In Progress	The upgrade is in progress. A progress bar shows the upgrade status. On-screen messages also show node-level faults and display the node ID of each node in the cluster as the upgrade progresses. You can monitor the status of each node using the Element UI or the NetApp Element plug-in for vCenter Server UI.
Upgrade Pausing	You can choose to pause the upgrade. Depending on the state of the upgrade process, the pause operation can succeed or fail. You will see a UI prompt asking you to confirm the pause operation. To ensure that the cluster is in a safe spot before pausing an upgrade, it can take up to two hours for the upgrade operation to be completely paused. To resume the upgrade, select **Resume**.
Paused	You paused the upgrade. Select **Resume** to resume the process.

Upgrade state	Description
Error	An error has occurred during the upgrade. You can download the error log and send it to NetApp Support. After you resolve the error, you can return to the page, and select **Resume**. When you resume the upgrade, the progress bar goes backwards for a few minutes while the system runs the health check and checks the current state of the upgrade.

What happens if an upgrade fails using NetApp Hybrid Cloud Control

If a drive or node fails during an upgrade, the Element UI will show cluster faults. The upgrade process does not proceed to the next node, and waits for the cluster faults to resolve. The progress bar in the UI shows that the upgrade is waiting for the cluster faults to resolve. At this stage, selecting **Pause** in the UI will not work, because the upgrade waits for the cluster to be healthy. You will need to engage NetApp Support to assist with the failure investigation.

NetApp Hybrid Cloud Control has a pre-set three-hour waiting period, during which one of the following scenarios can happen:

- The cluster faults get resolved within the three-hour window, and upgrade resumes. You do not need to take any action in this scenario.
- The problem persists after three hours, and the upgrade status shows **Error** with a red banner. You can resume the upgrade by selecting **Resume** after the problem is resolved.
- NetApp Support has determined that the upgrade needs to be temporarily aborted to take corrective action before the three-hour window. Support will use the API to abort the upgrade.

> Aborting the cluster upgrade while a node is being updated might result in the drives being ungracefully removed from the node. If the drives are ungracefully removed, adding the drives back during an upgrade will require manual intervention by NetApp Support. The node might be taking longer to do firmware updates or post update syncing activities. If the upgrade progress seems stalled, contact NetApp Support for assistance.

Use NetApp Hybrid Cloud Control API to upgrade storage firmware

You can use APIs to upgrade storage nodes in a cluster to the latest Element software version. You can use an automation tool of your choice to run the APIs. The API workflow documented here uses the REST API UI available on the management node as an example.

Steps

1. Download the storage firmware upgrade package to a device that is accessible to the management node; go to the Element software downloads page and download the latest storage firmware image.
2. Upload the storage firmware upgrade package to the management node:
   a. Open the management node REST API UI on the management node:

      ```
      https://<ManagementNodeIP>/package-repository/1/
      ```

   b. Select **Authorize** and complete the following:

i. Enter the cluster user name and password.
ii. Enter the client ID as `mnode-client`.
iii. Select **Authorize** to begin a session.
iv. Close the authorization window.

c. From the REST API UI, select **POST /packages**.
d. Select **Try it out**.
e. Select **Browse** and select the upgrade package.
f. Select **Execute** to initiate the upload.
g. From the response, copy and save the package ID (`"id"`) for use in a later step.

3. Verify the status of the upload.
   a. From the REST API UI, select **GET /packages/{id}/status**.
   b. Select **Try it out**.
   c. Enter the firmware package ID you copied in the previous step in **id**.
   d. Select **Execute** to initiate the status request.

      The response indicates `state` as `SUCCESS` when complete.

4. Locate the installation asset ID:
   a. Open the management node REST API UI on the management node:

      ```
      https://<ManagementNodeIP>/inventory/1/
      ```

   b. Select **Authorize** and complete the following:
      i. Enter the cluster user name and password.
      ii. Enter the client ID as `mnode-client`.
      iii. Select **Authorize** to begin a session.
      iv. Close the authorization window.
   c. From the REST API UI, select **GET /installations**.
   d. Select **Try it out**.
   e. Select **Execute**.
   f. From the response, copy the installation asset ID (`id`).

```
"id": "abcd01e2-xx00-4ccf-11ee-11f111xx9a0b",
"management": {
  "errors": [],
  "inventory": {
    "authoritativeClusterMvip": "10.111.111.111",
    "bundleVersion": "2.14.19",
    "managementIp": "10.111.111.111",
    "version": "1.4.12"
```

g. From the REST API UI, select **GET /installations/{id}**.

h. Select **Try it out**.

i. Paste the installation asset ID into the **id** field.

j. Select **Execute**.

k. From the response, copy and save the storage cluster ID (`"id"`) of the cluster you intend to upgrade for use in a later step.

```
"storage": {
  "errors": [],
  "inventory": {
    "clusters": [
      {
        "clusterUuid": "a1bd1111-4f1e-46zz-ab6f-0a1111b1111x",
        "id": "a1bd1111-4f1e-46zz-ab6f-a1a1a111b012",
```

5. Run the storage firmware upgrade:

a. Open the storage REST API UI on the management node:

```
https://<ManagementNodeIP>/storage/1/
```

b. Select **Authorize** and complete the following:

i. Enter the cluster user name and password.

ii. Enter the client ID as `mnode-client`.

iii. Select **Authorize** to begin a session.

iv. Close the window.

c. Select **POST /upgrades**.

d. Select **Try it out**.

e. Enter the upgrade package ID in the parameter field.

f. Enter the storage cluster ID in the parameter field.

g. Select **Execute** to initiate the upgrade.

The response should indicate state as `initializing`:


```
{
  "_links": {
    "collection": "https://localhost:442/storage/upgrades",
    "self": "https://localhost:442/storage/upgrades/3fa85f64-1111-
4562-b3fc-2c963f66abc1",
    "log": https://localhost:442/storage/upgrades/3fa85f64-1111-4562-
b3fc-2c963f66abc1/log
  },
  "storageId": "114f14a4-1a1a-11e9-9088-6c0b84e200b4",
  "upgradeId": "334f14a4-1a1a-11e9-1055-6c0b84e2001b4",
  "packageId": "774f14a4-1a1a-11e9-8888-6c0b84e200b4",
  "config": {},
  "state": "initializing",
  "status": {
    "availableActions": [
      "string"
    ],
    "message": "string",
    "nodeDetails": [
      {
        "message": "string",
        "step": "NodePreStart",
        "nodeID": 0,
        "numAttempt": 0
      }
    ],
    "percent": 0,
    "step": "ClusterPreStart",
    "timestamp": "2020-04-21T22:10:57.057Z",
    "failedHealthChecks": [
      {
        "checkID": 0,
        "name": "string",
        "displayName": "string",
        "passed": true,
        "kb": "string",
        "description": "string",
        "remedy": "string",
        "severity": "string",
        "data": {},
        "nodeID": 0
      }
    ]
  },
```

```
  "taskId": "123f14a4-1a1a-11e9-7777-6c0b84e123b2",
  "dateCompleted": "2020-04-21T22:10:57.057Z",
  "dateCreated": "2020-04-21T22:10:57.057Z"
}
```

h. Copy the upgrade ID (`"upgradeId"`) that is part of the response.

6. Verify the upgrade progress and results:
   a. Select **GET /upgrades/{upgradeId}**.
   b. Select **Try it out**.
   c. Enter the upgrade ID from the previous step in **upgradeId**.
   d. Select **Execute**.
   e. Do one of the following if there are problems or special requirements during the upgrade:

Option	Steps
You need to correct cluster health issues due to `failedHealthChecks` message in the response body.	1. Go to the specific KB article listed for each issue or perform the specified remedy. 2. If a KB is specified, complete the process described in the relevant KB article. 3. After you have resolved cluster issues, reauthenticate if needed and select **PUT /upgrades/{upgradeId}**. 4. Select **Try it out**. 5. Enter the upgrade ID from the previous step in **upgradeId**. 6. Enter `"action":"resume"` in the request body. `{ "action": "resume" }` 7. Select **Execute**.

Option	Steps
You need to pause the upgrade because the maintenance window is closing or for another reason.	1. Reauthenticate if needed and select **PUT /upgrades/{upgradeId}**. 2. Select **Try it out**. 3. Enter the upgrade ID from the previous step in **upgradeId**. 4. Enter `"action":"pause"` in the request body. `{ "action": "pause" }` 5. Select **Execute**.

f. Run the **GET /upgrades/{upgradeId}** API multiple times, as needed, until the process is complete.

During the upgrade, the `status` indicates `running` if no errors are encountered. As each node is upgraded, the `step` value changes to `NodeFinished`.

The upgrade has finished successfully when the `percent` value is `100` and the `state` indicates `finished`.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Upgrade a management node

You can upgrade your management node to management node 12.5 or later from version 12.3.x or later.

Upgrading the management node operating system is no longer required to upgrade Element software on the storage cluster. You can simply upgrade the management services to the latest version to perform Element upgrades using NetApp Hybrid Cloud Control. Follow the management node upgrade procedure for your scenario if you would like to upgrade the management node operating system for other reasons, such as security remediation.

If you require information on upgrading management nodes 12.2 or earlier, see the Element 12.3.x management node upgrade documentation.

Upgrade options

Choose one of the following options:

- Upgrade a management node to version 12.5 or later from version 12.3.x or later

- Reconfigure authentication using the management node REST API

  Choose this option if you have **sequentially** updated (1) your management services version and (2) your Element storage version and you want to **keep** your existing management node:

> If you do not sequentially update your management services followed by Element storage, you cannot reconfigure reauthentication using this procedure. Follow the appropriate upgrade procedure instead.

Upgrade a management node to version 12.5 or later from version 12.3.x or later

You can perform an in-place upgrade of the management node from version 12.3.x or later to version 12.5 or later without needing to provision a new management node virtual machine.

> The Element 12.5 or later management node is an optional upgrade. It is not required for existing deployments.

What you'll need

- The RAM of the management node VM is 24GB.
- The management node you are intending to upgrade is version 12.0 and uses IPv4 networking. The management node version 12.5 or later does not support IPv6.

  > To check the version of your management node, log in to your management node and view the Element version number in the login banner.

- You have updated your management services bundle to the latest version using NetApp Hybrid Cloud Control. You can access NetApp Hybrid Cloud Control from the following IP: `https://<ManagementNodeIP>`
- If you are updating your management node to version 12.5 or later, you need management services 2.21.61 or later to proceed.
- You have configured an additional network adapter (if required) using the instructions for configuring an additional storage NIC.

  > Persistent volumes might require an additional network adapter if eth0 is not able to be routed to the SVIP. Configure a new network adapter on the iSCSI storage network to allow the configuration of persistent volumes.

- Storage nodes are running Element 12.3.x or later.

Steps

1. Log in to the management node virtual machine using SSH or console access.
2. Download the management node ISO for Element software from the NetApp Support Site to the management node virtual machine.

   > The name of the ISO is similar to `solidfire-fdva-<Element release>-patchX-XX.X.X.XXXX.iso`

3. Check the integrity of the download by running md5sum on the downloaded file and compare the output to what is available on the NetApp Support Site for Element software, as in the following example:

```
sudo md5sum -b <path to iso>/solidfire-fdva-<Element release>-patchX-
XX.X.X.XXXX.iso
```

4. Mount the management node ISO image and copy the contents to the file system using the following commands:

```
sudo mkdir -p /upgrade
```

```
sudo mount <solidfire-fdva-<Element release>-patchX-XX.X.X.XXXX.iso>
/mnt
```

```
sudo cp -r /mnt/* /upgrade
```

5. Change to the home directory, and unmount the ISO file from `/mnt`:

```
sudo umount /mnt
```

6. Delete the ISO to conserve space on the management node:

```
sudo rm <path to iso>/solidfire-fdva-<Element release>-patchX-
XX.X.X.XXXX.iso
```

7. On the management node that you are upgrading, run the following command to upgrade your management node OS version. The script retains all necessary configuration files after the upgrade, such as Active IQ collector and proxy settings.

```
sudo /sf/rtfi/bin/sfrtfi_inplace
file:///upgrade/casper/filesystem.squashfs sf_upgrade=1
```

The management node reboots with a new OS after the upgrade process completes.

> After you run the sudo command described in this step, the SSH session is killed. Console access is required for continued monitoring. If no console access is available to you when performing the upgrade, retry the SSH login and verify connectivity after 15 to 30 minutes. Once you log in, you can confirm the new OS version in the SSH banner that indicates that the upgrade was successful.

8. On the management node, run the `redeploy-mnode` script to retain previous management services configuration settings:

> The script retains previous management services configuration, including configuration from the Active IQ collector service, controllers (vCenters), or proxy, depending on your settings.

```
sudo /sf/packages/mnode/redeploy-mnode -mu <mnode user>
```

If you had previously disabled SSH functionality on the management node, you need to disable SSH again on the recovered management node. SSH capability that provides NetApp Support remote support tunnel (RST) session access is enabled on the management node by default.

Reconfigure authentication using the management node REST API

You can keep your existing management node if you have sequentially upgraded (1) management services and (2) Element storage. If you have followed a different upgrade order, see the procedures for in-place management node upgrades.

Before you begin

- You have updated your management services to version 2.20.69 or later.
- Your storage cluster is running Element 12.3 or later.
- You have sequentially updated your management services followed by upgrading your Element storage. You cannot reconfigure authentication using this procedure unless you have completed upgrades in the sequence described.

Steps

1. Open the management node REST API UI on the management node:

   ```
   https://<ManagementNodeIP>/mnode
   ```

2. Select **Authorize** and complete the following:
   a. Enter the cluster user name and password.
   b. Enter the client ID as `mnode-client` if the value is not already populated.
   c. Select **Authorize** to begin a session.
3. From the REST API UI, select **POST /services/reconfigure-auth**.
4. Select **Try it out**.
5. For the **load_images** parameter, select `true`.
6. Select **Execute**.

   The response body indicates that reconfiguration was successful.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Upgrade the Element Plug-in for vCenter Server

For existing vSphere environments with a registered NetApp Element Plug-in for VMware

vCenter Server, you can update your plug-in registration after you first update the management services package that contains the plug-in service.

You can update the plug-in registration on vCenter Server Virtual Appliance (vCSA) or Windows using the registration utility. You must change your registration for the vCenter Plug-in on every vCenter Server where you need to use the plug-in.

Management services 2.22.7 includes Element Plug-in for vCenter Server 5.0 which contains the remote plug-in. If you use the Element plug-in, you should upgrade to management services 2.22.7 or later to comply with the VMware directive that removes support for local plug-ins. Learn more.

Element vCenter Plug-in 5.0 or later

This upgrade procedure covers the following upgrade scenarios:

- You are upgrading to Element Plug-in for vCenter Server 5.3, 5.2, 5.1, or 5.0.
- You are upgrading to an 8.0 or 7.0 HTML5 vSphere Web Client.

Element Plug-in for vCenter 5.0 or later is not compatible with vCenter Server 6.7 and 6.5.

When you upgrade from Element Plug-in for vCenter Server 4.x to 5.x, the clusters already configured with the plug-in are lost because the data cannot be copied from a vCenter instance to a remote plug-in. You must re-add the clusters to the remote plug-in. This is a one-time activity when upgrading from a local plug-in to a remote plug-in.

Element vCenter Plug-in 4.10 or earlier

This upgrade procedure covers the following upgrade scenarios:

- You are upgrading to Element Plug-in for vCenter Server 4.10, 4.9, 4.8, 4.7, 4.6, 4.5, or 4.4.
- You are upgrading to a 7.0, 6.7, or 6.5 HTML5 vSphere Web Client.

  - The plug-in is not compatible with VMware vCenter Server 8.0 for Element Plug-in for VMware vCenter Server 4.x.
  - The plug-in is not compatible with VMware vCenter Server 6.5 for Element Plug-in for VMware vCenter Server 4.6, 4.7, and 4.8.

- You are upgrading to a 6.7 Flash vSphere Web Client.

The plug-in is compatible with vSphere Web Client version 6.7 U2 for Flash, 6.7 U3 (Flash and HTML5), and 7.0 U1. The plug-in is not compatible with version 6.7 U2 build 13007421 of the HTML5 vSphere Web Client and other 6.7 U2 builds released prior to update 2a (build 13643870). For more information about supported vSphere versions, see the release notes for your version of the plug-in.

What you'll need

- **Admin privileges**: You have vCenter Administrator role privileges to install a plug-in.

- **vSphere upgrades**: You have performed any required vCenter upgrades before upgrading the NetApp Element Plug-in for vCenter Server. This procedure assumes that vCenter upgrades have already been completed.
- **vCenter Server**: Your vCenter Plug-in version 4.x or 5.x is registered with a vCenter Server. From the registration utility (`https://<ManagementNodeIP>:9443`), select **Registration Status**, complete the necessary fields, and select **Check Status** to verify that the vCenter Plug-in is already registered and the version number of the current installation.
- **Management services updates**: You have updated your management services bundle to the latest version. Updates to the vCenter plug-in are distributed using management services updates that are released outside of major product releases for NetApp SolidFire all-flash storage.
- **Management node upgrades**:

  Element vCenter Plug-in 5.0 or later

  You are running a management node that has been upgraded to version 12.3.x or later.

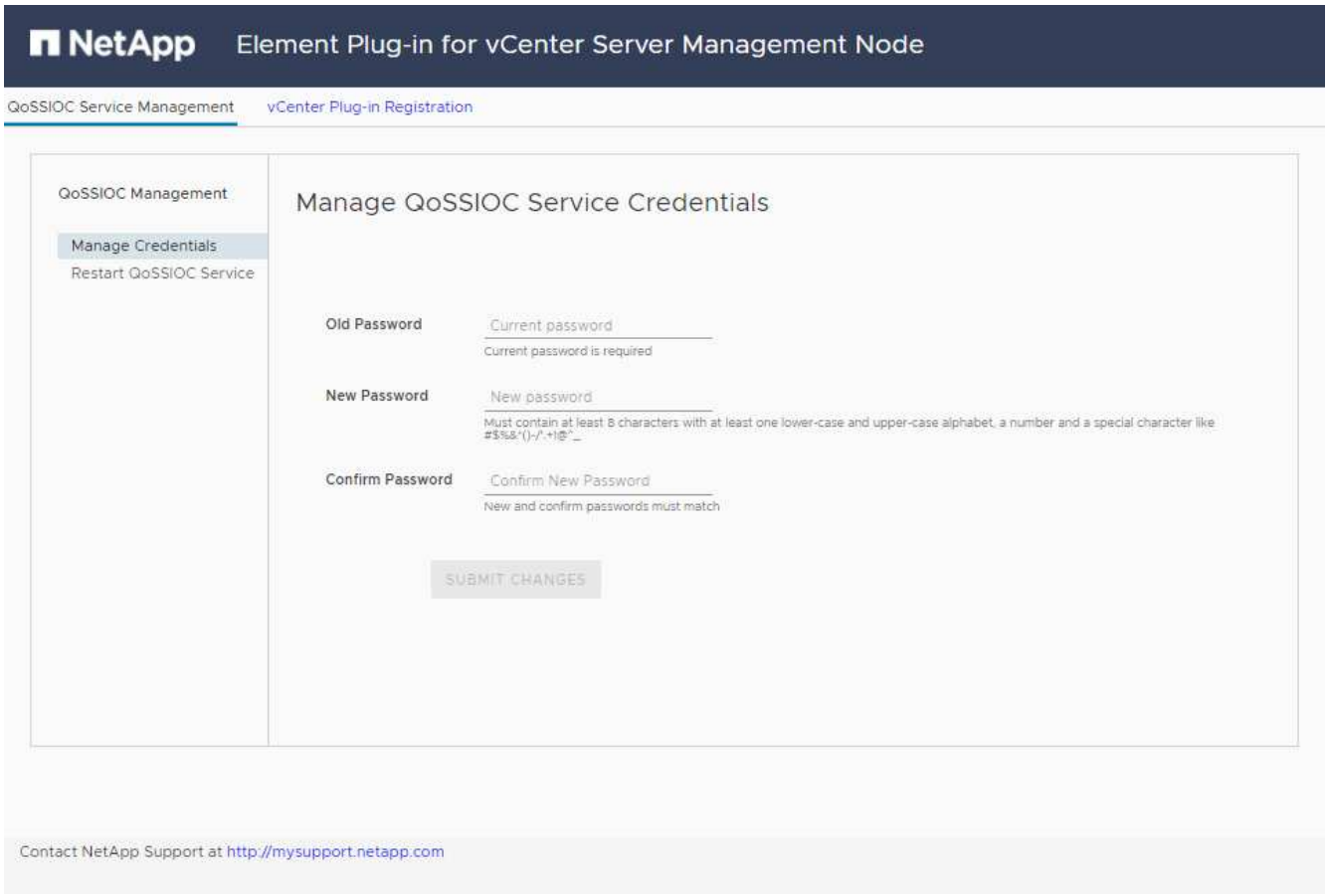
  Element vCenter Plug-in 4.10 or earlier

  For Element vCenter plug-in 4.4 to 4.10, you are running a management node that has been upgraded to version 11.3 or later. vCenter Plug-in 4.4 or later requires an 11.3 or later management node with a modular architecture that provides individual services. Your management node must be powered on with its IP address or DHCP address configured.

- **Element storage upgrades**:
  - Beginning with Element vCenter plug-in 5.0, you have a cluster running NetApp Element software 12.3.x or later.
  - For Element vCenter plug-in 4.10 or earlier, you have a cluster running NetApp Element software 11.3 or later.
- **vSphere Web Client**: You have logged out of the vSphere Web Client before beginning any plug-in upgrade. The web client will not recognize updates made during this process to your plug-in if you do not log out.

Steps

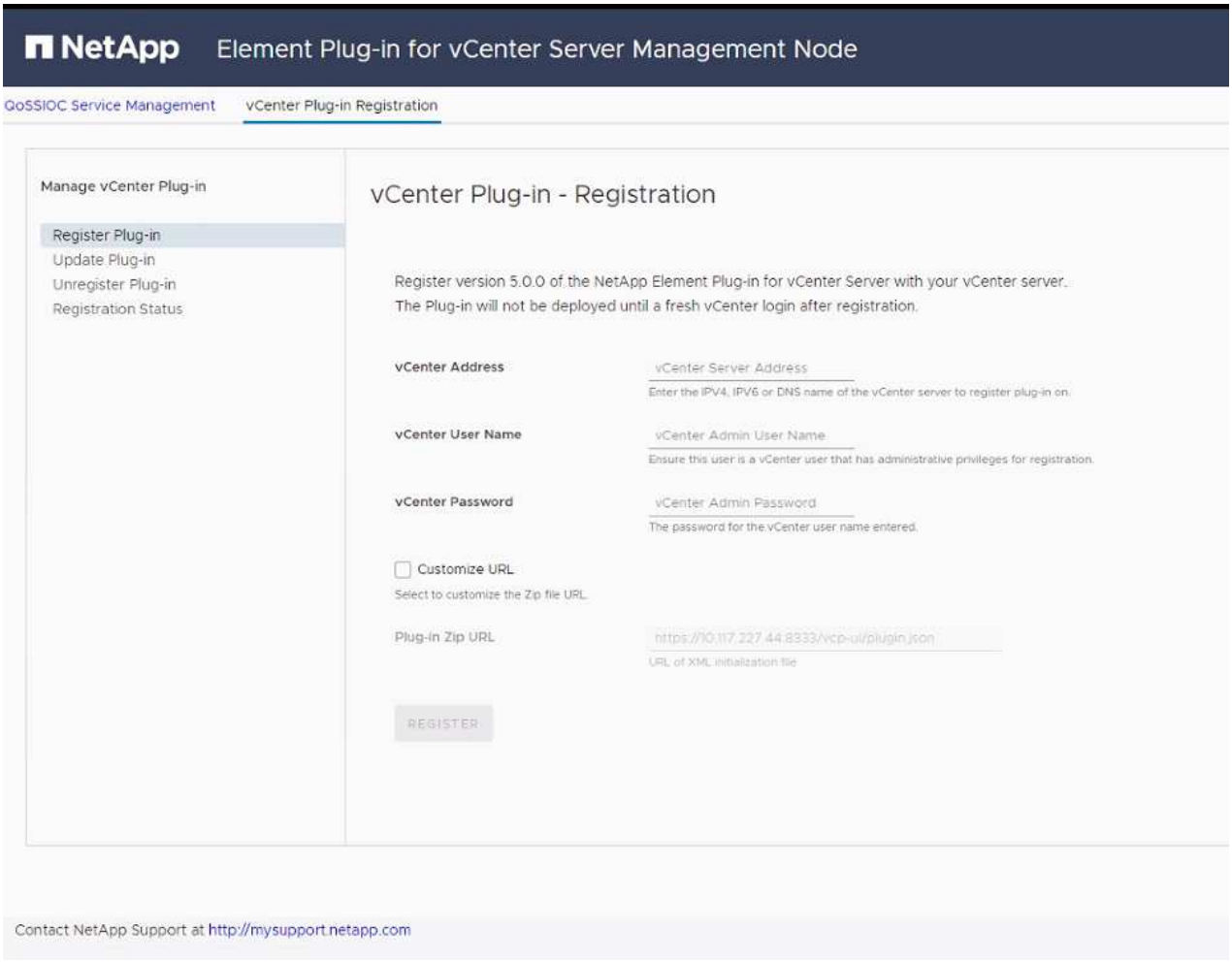
1. Enter the IP address for your management node in a browser, including the TCP port for registration:
   `https://<ManagementNodeIP>:9443`
   The registration utility UI opens to the **Manage QoSSIOC Service Credentials** page for the plug-in.



2. Select **vCenter Plug-in Registration**.

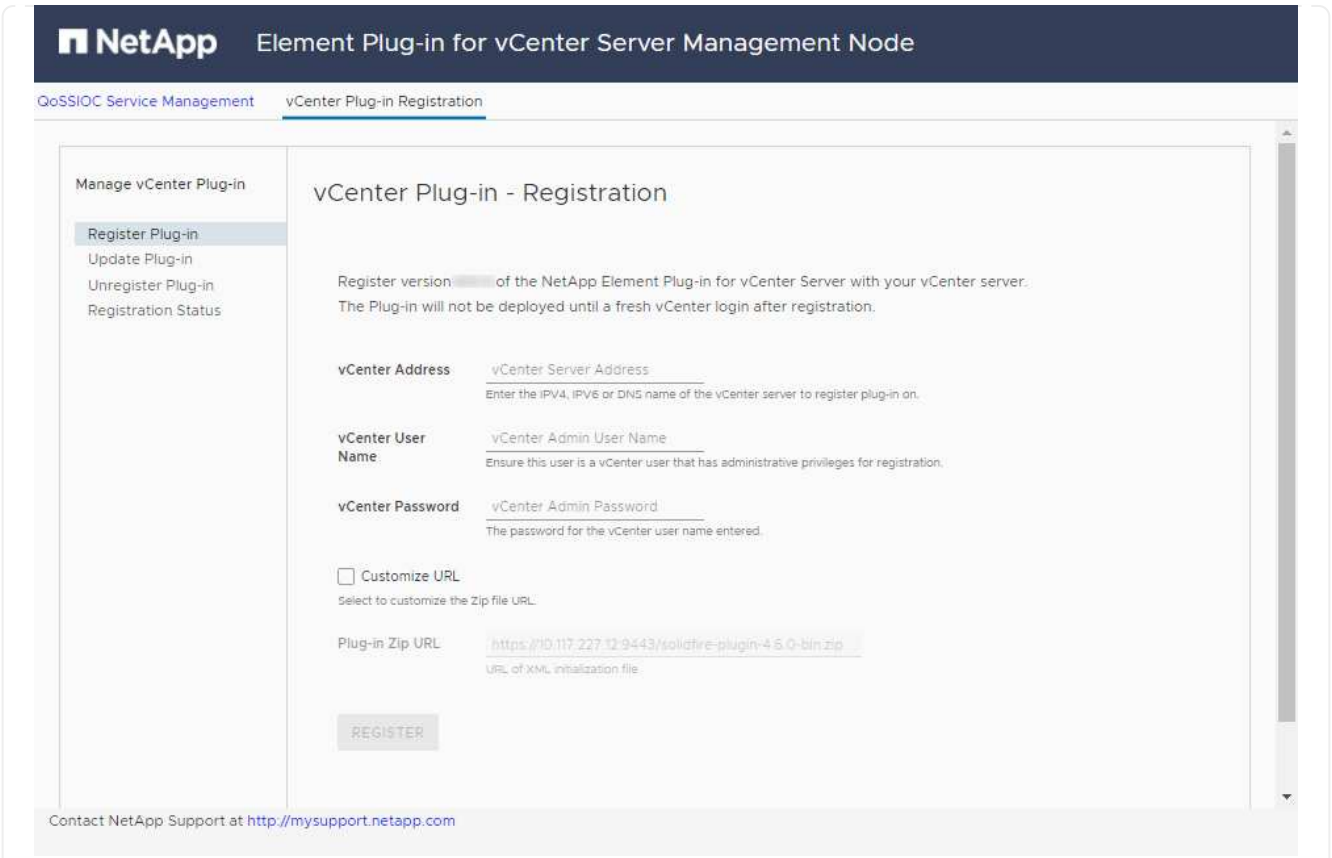
Element vCenter Plug-in 5.0 or later

The vCenter Plug-in Registration page appears:



Element vCenter Plug-in 4.10 or earlier

The vCenter Plug-in Registration page appears:



3. Within **Manage vCenter Plug-in**, select **Update Plug-in**.
4. Confirm or update the following information:
   a. The IPv4 address or the FQDN of the vCenter service on which you will register your plug-in.
   b. The vCenter Administrator user name.

      The user name and password credentials you enter must be for a user with vCenter Administrator role privileges.

   c. The vCenter Administrator password.
   d. (For in-house servers/dark sites) Depending on your Element Plug-in for vCenter version, a custom URL for the plug-in JSON file or plug-in ZIP:

Element vCenter Plug-in 5.0 or later

A custom URL for the plug-in JSON file.

> You can select **Custom URL** to customize the URL if you are using an HTTP or HTTPS server (dark site) or have modified the JSON file name or network settings. For additional configuration steps if you intend to customize a URL, see Element Plug-in for vCenter Server documentation about modifying vCenter properties for an in-house (dark site) HTTP server.

Element vCenter Plug-in 4.10 or earlier

A custom URL for the plug-in ZIP.

> You can select **Custom URL** to customize the URL if you are using an HTTP or HTTPS server (dark site) or have modified the ZIP file name or network settings. For additional configuration steps if you intend to customize a URL, see Element Plug-in for vCenter Server documentation about modifying vCenter properties for an in-house (dark site) HTTP server.

5. Select **Update**.

   A banner appears in the registration utility UI when the registration is successful.

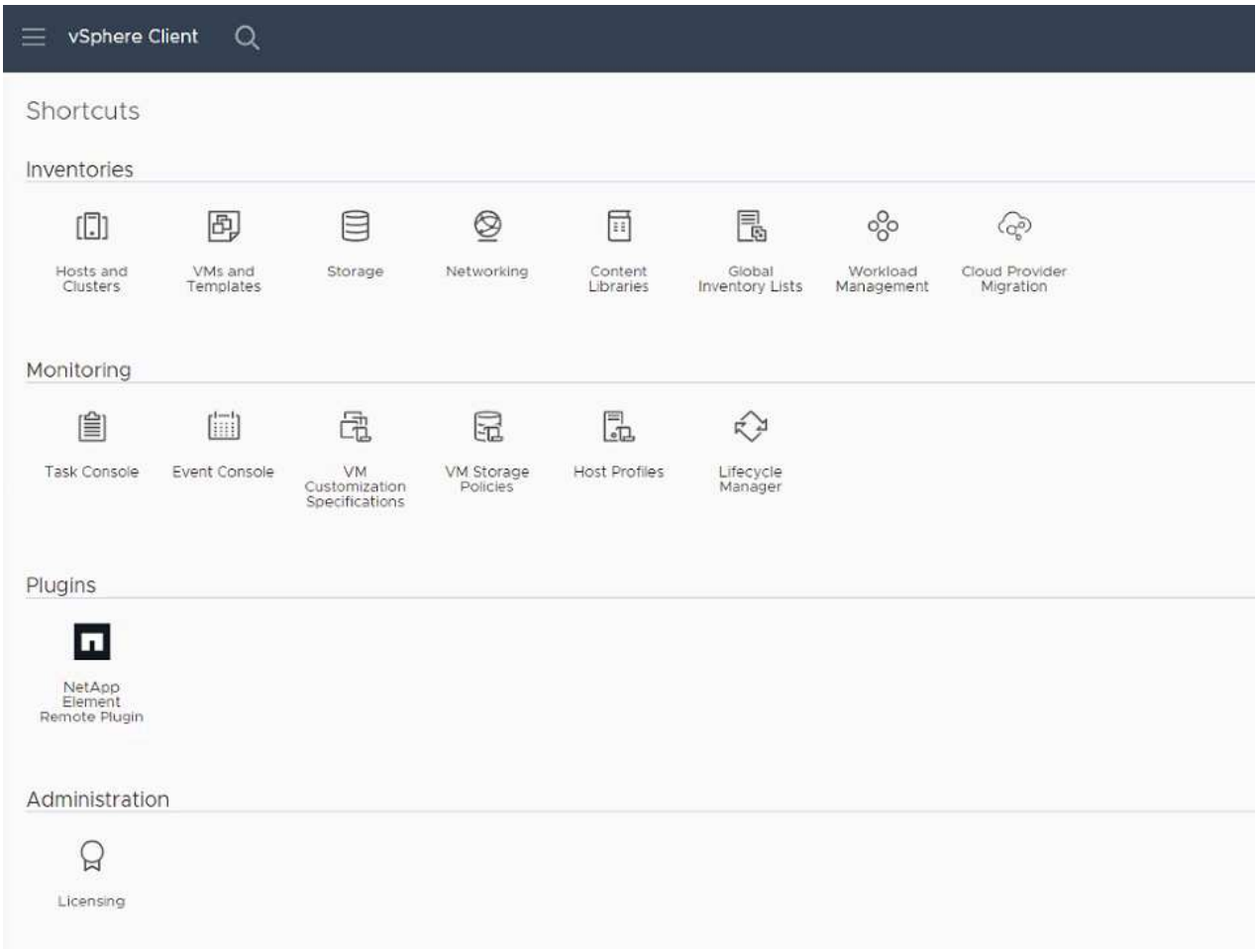
6. Log in to the vSphere Web Client as a vCenter Administrator. If you are already logged in to the vSphere Web Client, you must first log out, wait two to three minutes, and then log in again.

   > This action creates a new database and completes the installation in the vSphere Web Client.

7. In the vSphere Web Client, look for the following completed tasks in the task monitor to ensure installation has completed: `Download plug-in` and `Deploy plug-in`.
8. Verify that the plug-in extension points appear in the **Shortcuts** tab of the vSphere Web Client and in the side panel.

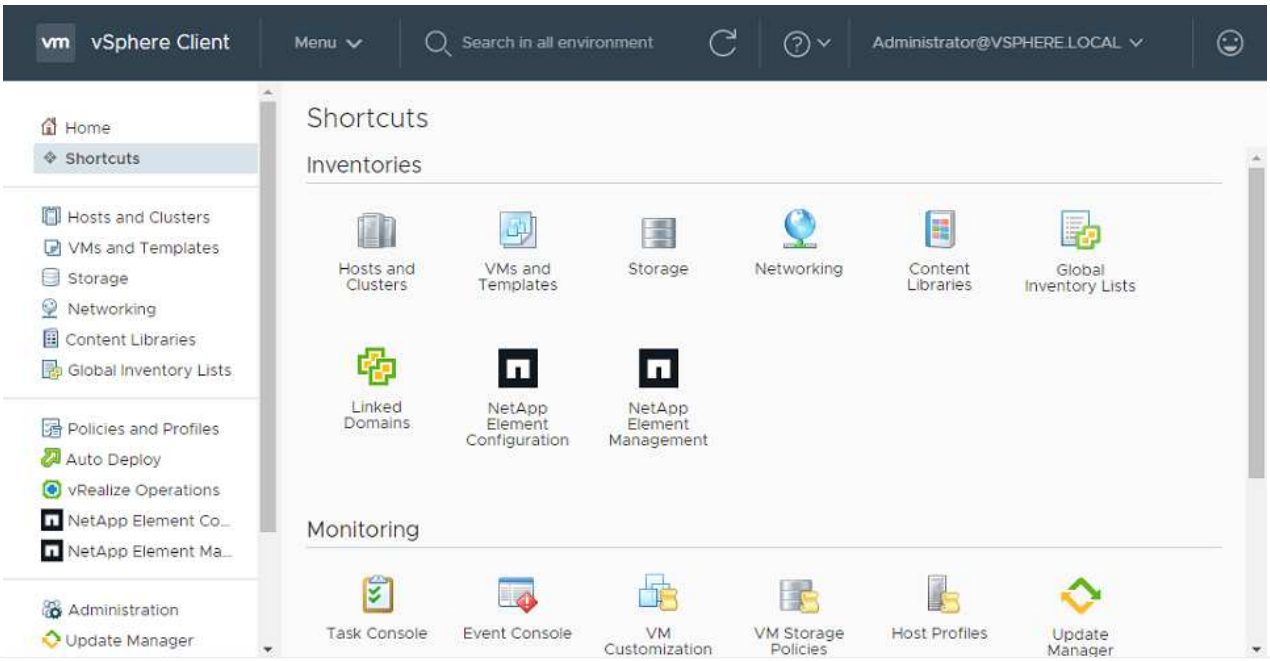
Element vCenter Plug-in 5.0 or later

The NetApp Element Remote Plugin extension point appears:



Element vCenter Plug-in 4.10 or earlier

The NetApp Element Configuration and Management extension points appear:



If the vCenter Plug-in icons are not visible, see Element Plug-in for vCenter Server documentation about troubleshooting the plug-in.

After upgrading to NetApp Element Plug-in for vCenter Server 4.8 or later with VMware vCenter Server 6.7U1, if the storage clusters are not listed or a server error appears in the **Clusters** and **QoSSIOC Settings** sections of the NetApp Element Configuration, see Element Plug-in for vCenter Server documentation about troubleshooting these errors.

9. Verify the version change in the **About** tab in the **NetApp Element Configuration** extension point of the plug-in.

   You should see the following version details or details of a more recent version:

```
NetApp Element Plug-in Version: 5.3
NetApp Element Plug-in Build Number: 9
```

The vCenter Plug-in contains online Help content. To ensure that your Help contains the latest content, clear your browser cache after upgrading your plug-in.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Upgrade your vSphere components for a NetApp SolidFire storage system with the Element Plug-in for vCenter Server

When you upgrade the VMware vSphere components of your SolidFire Element storage installation, there are some additional steps you will need to take for systems with Element Plug-in for vCenter Server.

Steps

1. For vCSA upgrades, clear QoSSIOC settings in the plug-in (**NetApp Element Configuration > QoSSIOC Settings**). The **QoSSIOC Status** field displays `Not Configured` after the process is complete.
2. For vCSA and Windows upgrades, unregister the plug-in from the vCenter Server with which it is associated using the registration utility.
3. Upgrade vSphere, including vCenter Server, ESXi, VMs, and other VMware components.

   You should upgrade to NetApp Element Plug-in for vCenter Server 5.0 or later to give you the capability to deploy the plug-in with VMware vCenter 7.0 Update 3 without having to apply a workaround.

   With Element Plug-in for vCenter Server 4.x, when you upgrade to VMware vCenter Server 7.0 Update 3, the plug-in fails to deploy. To resolve this issue using Spring Framework 4, see this KB article.

4. Register the Element Plug-in for vCenter Server again with vCenter.

5. Add clusters using the plug-in.
6. Configure QoSSIOC settings using the plug-in.
7. Enable QoSSIOC for all datastores controlled by the plug-in.

Find more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Earlier versions of SolidFire and NetApp Element software documentation

Documentation for previous releases is available on the NetApp Support site.

- Element 12.3.x documentation
- Element 12.2.1 documentation
- Element 12.2 documentation
- Element 12.0.1 documentation
- Element 12.0 documentation
- Element 11.8.2 documentation
- Element 11.8.1 documentation
- Element 11.8 documentation
- Element 11.7 documentation
- Element 11.5.1 documentation
- Element 11.5 documentation
- Element 11.3P1 documentation
- Element 11.3.2 documentation
- Element 11.1 documentation and earlier versions

For more information

- SolidFire and Element Software Documentation
- NetApp Element Plug-in for vCenter Server

Legal notices

Legal notices provide access to copyright statements, trademarks, patents, and more.

Copyright

https://www.netapp.com/company/legal/copyright/

Trademarks

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

https://www.netapp.com/company/legal/trademarks/

Patents

A current list of NetApp owned patents can be found at:

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

Privacy policy

https://www.netapp.com/company/legal/privacy-policy/

Open source

Notice files provide information about third-party copyright and licenses used in NetApp software.

- Notice for Element software 12.7
- Notice for Ember OS 12.7
- Notice for Management Node 12.7
- Notice for Element software 12.5
- Notice for Management Node 12.5
- Notice for Management Services 2.25.42 (NetApp Element Plug-in for VMware vCenter Server 5.3.9)
- Notice for Management Services 2.24.40 (NetApp Element Plug-in for VMware vCenter Server 5.2.12)
- Notice for Management Services 2.23.64 (NetApp Element Plug-in for VMware vCenter Server 5.1.12)
- Notice for Management Services 2.22.7 (NetApp Element Plug-in for VMware vCenter Server 5.0.37)
- Notice for Management Services 2.21.61 (NetApp Element Plug-in for VMware vCenter Server 4.10.12)
- Notice for Management Services 2.20.69 (NetApp Element Plug-in for vCenter Server 4.9.14)
- Notice for Management Services 2.19.48 (NetApp Element Plug-in for vCenter Server 4.8.34)
- Notice for Management Services 2.18.91 (NetApp Element Plug-in for vCenter Server 4.7.10)
- Notice for Management Services 2.17.56 (NetApp Element Plug-in for vCenter Server 4.6.32)
- Notice for Management Services 2.17.52 (NetApp Element Plug-in for vCenter Server 4.6.29)

- Notice for Management Services 2.16 (NetApp Element Plug-in for vCenter Server 4.6.29)
- Notice for Management Services 2.14 (NetApp Element Plug-in for vCenter Server 4.5.42)
- Notice for Management Services 2.13 (NetApp Element Plug-in for vCenter Server 4.5.42)
- Notice for Storage Firmware Bundle 2.175.0
- Notice for Storage Firmware Bundle 2.164.0
- Notice for Storage Firmware Bundle 2.150
- Notice for Storage Firmware Bundle 2.146
- Notice for Storage Firmware Bundle 2.99.2
- Notice for Storage Firmware Bundle 2.76
- Notice for Storage Firmware Bundle 2.27