NetApp® Element® 12.2

API Reference Guide

November 2020 | 215-15139_2020-11_en-us
doccomments@netapp.com

NetApp®

Contents

About the Element software API....12
Request object members....12
Response object members....12
Request endpoints....13
API authentication....13
Asynchronous methods....13
Attributes....14

Common objects....16
account....16
authSessionInfo....17
bulkVolumeJob....18
binding (virtual volumes)....18
certificateDetails....19
cluster....19
clusterAdmin....21
clusterCapacity....21
clusterConfig....23
clusterInfo....23
clusterPair....25
clusterStats....25
clusterStructure....27
drive....28
driveStats....29
error....30
event....31
fault....32
fibreChannelPort....34
fipsErrorNodeReport....35
fipsNodeReport....35
fipsReport....35
groupSnapshot....36
hardwareInfo....37
host (virtual volumes)....38
idpConfigInfo....38
initiator....39
keyProviderKmip....40
keyServerKmip....40
ldapConfiguration....41
loggingServer....42
network (bonded interfaces)....42
network (all interfaces)....45
network (Ethernet interfaces)....46
network (local interfaces)....47
network (SNMP)....48
networkInterface....49

node....49
nodeProtectionDomains....51
nodeStats....51
ontapVersionInfo....52
pendingActiveNode....52
pendingNode....54
protectionDomain....55
protectionDomainLevel....55
protectionDomainResiliency....56
protectionDomainTolerance....56
protectionSchemeResiliency....57
protectionSchemeTolerance....57
protocolEndpoint....58
QoS....58
QoSPolicy....59
remoteClusterSnapshotStatus....59
schedule....60
session (Fibre Channel)....61
session (iSCSI)....62
snapMirrorAggregate....64
snapMirrorClusterIdentity....64
snapMirrorEndpoint....64
snapMirrorJobScheduleCronInfo....65
snapMirrorLunInfo....65
snapMirrorNetworkInterface....66
snapMirrorNode....67
snapMirrorPolicy....67
snapMirrorPolicyRule....68
snapMirrorRelationship....68
snapMirrorVolume....70
snapMirrorVolumeInfo....71
snapMirrorVserver....71
snapMirrorVserverAggregateInfo....72
snapshot....72
snmpTrapRecipient....74
storageContainer....75
syncJob....75
task (virtual volumes)....77
usmUser....78
virtualNetwork....78
virtualVolume....79
volume....80
volumeAccessGroup....83
volumePair....84
volumeStats....85

Common methods....89
GetAPI....89
GetAsyncResult....95
GetCompleteStats....98
GetLimits....98
GetOrigin....100
GetRawStats....100

ListAsyncResults... 101

Account API methods... 103
AddAccount... 103
GetAccountByID... 104
GetAccountByName... 105
GetAccountEfficiency... 106
ListAccounts... 108
ModifyAccount... 109
RemoveAccount... 111

Administrator API methods... 113
AddClusterAdmin... 113
GetCurrentClusterAdmin... 114
GetLoginBanner... 115
ListClusterAdmins... 116
ModifyClusterAdmin... 117
RemoveClusterAdmin... 118
SetLoginBanner... 119

Cluster API methods... 121
AddNodes... 121
ClearClusterFaults... 123
CreateClusterInterfacePreference... 124
DeleteClusterInterfacePreference... 125
EnableFeature... 126
GetClusterCapacity... 127
GetClusterFullThreshold... 128
GetClusterHardwareInfo... 132
GetClusterInfo... 133
GetClusterInterfacePreference... 134
GetClusterMasterNodeID... 135
GetClusterStats... 135
GetClusterVersionInfo... 137
GetFeatureStatus... 139
GetLicenseKey... 140
GetLoginSessionInfo... 141
GetNodeHardwareInfo... 142
GetNodeStats... 143
ListActiveNodes... 144
ListAllNodes... 144
ListClusterFaults... 146
ListClusterInterfacePreferences... 147
ListEvents... 148
ListNodeStats... 151
ListISCSISessions... 152
ListServices... 153
ListPendingNodes... 155
ListPendingActiveNodes... 156
ModifyClusterFullThreshold... 157
ModifyClusterInterfacePreference... 162

RemoveNodes....163
SetLicenseKey....164
SetLoginSessionInfo....165
Shutdown....166

Cluster creation API Methods....168
CheckProposedCluster....168
CreateCluster....169
GetBootstrapConfig....171

Drive API methods....174
AddDrives....174
GetDriveHardwareInfo....175
GetDriveStats....177
ListDrives....178
ListDriveStats....179
RemoveDrives....181
SecureEraseDrives....182

Fibre Channel API methods....184
GetVolumeAccessGroupLunAssignments....184
ListFibreChannelPortInfo....185
ListFibreChannelSessions....188
ListNodeFibreChannelPortInfo....189
ModifyVolumeAccessGroupLunAssignments....190

Initiator API methods....193
CreateInitiators....193
DeleteInitiators....196
ListInitiators....197
ModifyInitiators....198

LDAP API methods....202
AddLdapClusterAdmin....202
EnableLdapAuthentication....203
DisableLdapAuthentication....205
GetLdapConfiguration....206
TestLdapAuthentication....207

Multi-factor authentication API methods....209
AddIdpClusterAdmin....209
CreateIdpConfiguration....210
DeleteAuthSession....211
DeleteAuthSessionsByClusterAdmin....212
DeleteAuthSessionsByUsername....213
DeleteIdpConfiguration....215
DisableIdpAuthentication....216
EnableIdpAuthentication....216

GetIdpAuthenticationState.... 217
ListActiveAuthSessions.... 218
ListIdpConfigurations.... 218
UpdateIdpConfiguration.... 220

Session authentication API methods.... 222
ListAuthSessionsByClusterAdmin.... 222
ListAuthSessionsByUsername.... 223

Node API methods.... 225
CheckPingOnVlan.... 225
CheckProposedNodeAdditions.... 227
CreateClusterSupportBundle.... 229
CreateSupportBundle.... 230
DeleteAllSupportBundles.... 232
DisableMaintenanceMode.... 233
DisableSsh.... 234
EnableMaintenanceMode.... 235
EnableSsh.... 237
GetClusterConfig.... 237
GetClusterState.... 238
GetConfig.... 239
GetDriveConfig.... 240
GetHardwareConfig.... 242
GetHardwareInfo.... 243
GetIpmiConfig.... 245
GetIpmiInfo.... 248
GetNetworkConfig.... 251
GetNetworkInterface.... 252
GetNodeActiveTlsCiphers.... 254
GetNodeFipsDrivesReport.... 255
GetNodeSSLCertificate.... 256
GetNodeSupportedTlsCiphers.... 258
GetPendingOperation.... 259
GetSshInfo.... 260
ListDriveHardware.... 260
ListNetworkInterfaces.... 262
ListTests.... 264
ListUtilities.... 265
RemoveNodeSSLCertificate.... 266
ResetDrives.... 266
ResetNode.... 267
ResetNodeSupplementalTlsCiphers.... 269
RestartNetworking.... 270
RestartServices.... 271
SetClusterConfig.... 272
SetConfig.... 273
SetNetworkConfig.... 274
SetNodeSSLCertificate.... 276
SetNodeSupplementalTlsCiphers.... 278
Shutdown.... 279
TestConnectEnsemble.... 280

TestConnectMvip....281
TestConnectSvip....283
TestDrives....285
TestHardwareConfig....286
TestLocateCluster....287
TestLocalConnectivity....288
TestNetworkConfig....290
TestPing....292
TestRemoteConnectivity....294

Replication API methods....297
Cluster pairing order of operations....297
Volume pairing order of operations....297
Supported modes of replication for paired clusters....298
CompleteClusterPairing....298
CompleteVolumePairing....299
ListClusterPairs....300
ListActivePairedVolumes....301
ModifyVolumePair....303
RemoveClusterPair....304
RemoveVolumePair....305
StartClusterPairing....305
StartVolumePairing....307

Security API methods....309
AddKeyServerToProviderKmip....309
CreateKeyProviderKmip....310
CreateKeyServerKmip....311
CreatePublicPrivateKeyPair....313
DeleteKeyProviderKmip....314
DeleteKeyServerKmip....315
DisableEncryptionAtRest....315
EnableEncryptionAtRest....316
GetClientCertificateSignRequest....318
GetKeyProviderKmip....319
GetKeyServerKmip....320
ListKeyProvidersKmip....321
ListKeyServersKmip....323
ModifyKeyServerKmip....325
RemoveKeyServerFromProviderKmip....327
TestKeyProviderKmip....328
TestKeyServerKmip....329

SnapMirror API methods....330
AbortSnapMirrorRelationship....330
BreakSnapMirrorRelationship....330
BreakSnapMirrorVolume....331
CreateSnapMirrorEndpoint....332
CreateSnapMirrorEndpointUnmanaged....333
CreateSnapMirrorRelationship....333
CreateSnapMirrorVolume....334

DeleteSnapMirrorEndpoints....335
DeleteSnapMirrorRelationships....336
GetOntapVersionInfo....336
GetSnapMirrorClusterIdentity....337
InitializeSnapMirrorRelationship....338
ListSnapMirrorAggregates....338
ListSnapMirrorEndpoints....339
ListSnapMirrorLuns....340
ListSnapMirrorNetworkInterfaces....340
ListSnapMirrorNodes....341
ListSnapMirrorPolicies....341
ListSnapMirrorSchedules....342
ListSnapMirrorRelationships....343
ListSnapMirrorVolumes....343
ListSnapMirrorVservers....344
ModifySnapMirrorEndpoint....345
ModifySnapMirrorEndpoint (unmanaged)....346
ModifySnapMirrorRelationship....347
UpdateSnapMirrorRelationship....347
QuiesceSnapMirrorRelationship....348
ResumeSnapMirrorRelationship....349
ResyncSnapMirrorRelationship....349

System configuration API methods....351
DisableBmcColdReset....351
DisableClusterSsh....351
DisableSnmp....353
EnableBmcColdReset....353
EnableClusterSsh....354
EnableSnmp....355
GetBinAssignmentProperties....356
GetClusterSshInfo....358
GetClusterStructure....359
GetFipsReport....360
GetLldpConfig....361
GetLldpInfo....361
GetNodeFipsDrivesReport....362
GetNtpInfo....363
GetNvramInfo....364
GetProtectionDomainLayout....364
GetRemoteLoggingHosts....366
GetSnmpACL....367
GetSnmpInfo....368
GetSnmpState....370
GetSnmpTrapInfo....371
GetSSLCertificate....372
ListProtectionDomainLevels....373
RemoveSSLCertificate....375
ResetNetworkConfig....375
ResetSupplementalTlsCiphers....376
SetClusterStructure....376
SetLldpConfig....377
SetNtpInfo....378

SetProtectionDomainLayout....379
SetRemoteLoggingHosts....382
SetSnmpACL....383
SetSnmpInfo....384
SetSnmpTrapInfo....386
SetSSLCertificate....387
SnmpSendTestTraps....389
TestAddressAvailability....390

Multitenant networking API methods....392
Virtual network naming conventions....392
AddVirtualNetwork....392
ModifyVirtualNetwork....394
ListVirtualNetworks....397
RemoveVirtualNetwork....399

Volume API methods....400
CancelClone....400
CancelGroupClone....400
CloneMultipleVolumes....401
CloneVolume....403
CopyVolume....406
CreateQoSPolicy....408
CreateVolume....409
CreateBackupTarget....413
DeleteQoSPolicy....414
DeleteVolume....415
DeleteVolumes....417
GetBackupTarget....419
GetVolumeStats....420
GetDefaultQoS....421
GetQoSPolicy....422
GetVolumeCount....423
GetVolumeEfficiency....424
ListActiveVolumes....425
ListBackupTargets....426
ListBulkVolumeJobs....427
ListDeletedVolumes....428
ListQoSPolicies....430
ListSyncJobs....431
ListVolumeQoSHistograms....433
ListVolumes....435
ListVolumeStats....437
ListVolumesForAccount....439
ListVolumeStatsByAccount....440
ListVolumeStatsByVirtualVolume....442
ListVolumeStatsByVolume....443
ListVolumeStatsByVolumeAccessGroup....445
ModifyBackupTarget....446
ModifyQoSPolicy....447
ModifyVolume....449
ModifyVolumes....453

PurgeDeletedVolume....457
PurgeDeletedVolumes....458
RemoveBackupTarget....459
RestoreDeletedVolume....460
SetDefaultQoS....461
StartBulkVolumeRead....462
StartBulkVolumeWrite....464
UpdateBulkVolumeStatus....466

Volume access group API methods....468
AddInitiatorsToVolumeAccessGroup....468
AddVolumesToVolumeAccessGroup....469
CreateVolumeAccessGroup....471
DeleteVolumeAccessGroup....473
ListVolumeAccessGroups....474
RemoveVolumesFromVolumeAccessGroup....475
RemoveInitiatorsFromVolumeAccessGroup....477
ModifyVolumeAccessGroup....478
GetVolumeAccessGroupEfficiency....481

Volume snapshot API methods....483
Snapshots overview....483
CreateGroupSnapshot....483
CreateSchedule....486
CreateSnapshot....493
DeleteGroupSnapshot....496
DeleteSnapshot....497
GetSchedule....498
ListGroupSnapshots....499
ListSchedules....501
ListSnapshots....502
ModifyGroupSnapshot....504
ModifySchedule....506
ModifySnapshot....509
RollbackToGroupSnapshot....511
RollbackToSnapshot....513

Virtual volume API methods....516
CreateStorageContainer....516
DeleteStorageContainers....517
GetStorageContainerEfficiency....518
GetVirtualVolumeCount....519
ListProtocolEndpoints....520
ListStorageContainers....521
ListVirtualVolumeBindings....522
ListVirtualVolumeHosts....524
ListVirtualVolumes....525
ListVirtualVolumeTasks....527
ModifyStorageContainer....528

Cluster fault codes.....530

Access control.....544

Response examples.....548
GetConfig..... 548
GetClusterHardwareInfo.....550
GetLldpInfo..... 560
GetNetworkConfig.....583
GetNodeHardwareInfo (output for iSCSI)..... 586
GetNodeHardwareInfo (output for Fibre Channel nodes).....587
GetNvramInfo.....593
ListActiveNodes..... 600
ListActiveVolumes.....602
TestHardwareConfig.....609

Contact NetApp Support.....614

Copyright and trademark.....615
Copyright..... 615
Trademark.....615

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.

Note: 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.

Note: 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

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
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

Name	Description	Type
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.

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.

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.

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

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

Related reference

AddAccount on page 103

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.

GetAccountByID on page 104

You can use `GetAccountByID` to get details about a specific account, given its accountID.

GetAccountByName on page 105

You can use `GetAccountByName` to get details about a specific account, given its username.

ListAccounts on page 108

You can use `ListAccounts` to get the entire list of storage tenant accounts, with optional paging support. Element accounts enable access to volumes.

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
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: • compressed • native	string
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 • active • done • 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

Name	Description	Type
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

Related reference

ListVirtualVolumeBindings on page 522

You can use the `ListVirtualVolumeBindings` method to get a list of all virtual volumes in the cluster that are bound to protocol endpoints.

protocolEndpoint on page 58

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.

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
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

Name	Description	Type
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
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

Related reference

GetClusterConfig on page 237

You can use the `GetClusterConfig` API method to return information about the cluster configuration the node uses to communicate with its cluster.

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

Related reference

GetCurrentClusterAdmin on page 114

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.

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

Name	Description	Type
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
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
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

Name	Description	Type
zeroBlocks	The total number of empty 4KiB blocks without data after the last round of garbage collection operation has completed.	integer

Related reference

GetClusterCapacity on page 127

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.

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
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.	string
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.	string
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
supportedProtectionSchemes	A list of all protection schemes that are supported on this storage cluster.	string array
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.	string
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.	string

Related reference

GetClusterInfo on page 133

You can use the `GetClusterInfo` method to return configuration information about the cluster.

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
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

Related reference

ListClusterPairs on page 300

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.

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

Name	Description	Calculation	Type
averageIOPSize	Average size in bytes of recent I/O to the cluster in the last 500 milliseconds.	Point in time	integer
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
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

Name	Description	Calculation	Type
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
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

Related reference

GetClusterStats on page 135

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.

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*

Related reference

GetClusterStructure on page 359

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.

SetClusterStructure on page 376

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.

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
driveSecurityFaultReason	If enabling or disabling drive security failed, the reason why it failed. If the value is "none", there was no failure.	string

Name	Description	Type
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
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

Related reference

ListDrives on page 178

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.

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
lifeRemainingPercent	Drive media wear out indicator.	integer

Name	Description	Type
lifetimeReadBytes	Total bytes read from this drive for the lifetime of the drive.	integer
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
readBytes	Total bytes read from the drive due to client operations.	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
writeBytes	Total bytes written to the drive due to client activity.	integer
writeOps	Total write operations to the drive due to client activity.	integer

Related reference

GetDriveStats on page 177

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.

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.

Related reference

ListEvents on page 148

You can use the `ListEvents` method to list events detected on the cluster, sorted from oldest to newest.

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
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

Name	Description	Type
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
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
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

Related reference

ListClusterFaults on page 146

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.

Cluster fault codes on page 530

If a storage cluster experiences an error or a state that might be of interest to an administrator, it generates a cluster fault. You can use the `ListClusterFaults` method to retrieve the current list of resolved and unresolved faults on a storage cluster.

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
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

Related reference

ListNodeFibreChannelPortInfo on page 189

You can use the `ListNodeFibreChannelPortInfo` method to list information about the Fibre Channel ports on a node.

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
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
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

Related reference

ListGroupSnapshots on page 499

You can use `ListGroupSnapshots` method to return information about all group snapshots that have been created.

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
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
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

Related reference

GetHardwareInfo on page 243

You can use the `GetHardwareInfo` method to get hardware information and status for a single node. Hardware information generally includes manufacturers, vendors, versions, drives, and other associated identification information.

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

Related reference

ListVirtualVolumeHosts on page 524

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.

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

Name	Description	Type
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

Name	Description	Type
volumeAccessGroups	A list of volume access group IDs that this initiator belongs to.	integer array

Related reference

ListInitiators on page 197

You can use the `ListInitiators` method to get the list of initiator IQNs or World Wide Port Names (WWPNs).

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
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

Name	Description	Type
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
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
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

Name	Description	Type
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

Related reference

GetLdapConfiguration on page 206

You can use the `GetLdapConfiguration` method to get the currently active LDAP configuration on the cluster.

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

Name	Description	Type
bond-downdelay	Time to wait 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	How often the MII link state is inspected for link failures.	integer
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	integer
bond-updelay	The time to wait before enabling a slave after a link is detected.	integer
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

Name	Description	Type
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

Member name	Available state	Pending state	Active state
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

Related reference

GetConfig on page 239
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.

GetNetworkConfig on page 251
You can use the `GetNetworkConfig` method to display the network configuration information for a node.

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)*

Name	Description	Type
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)*

Related reference

GetConfig on page 239
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.

GetNetworkConfig on page 251
You can use the `GetNetworkConfig` method to display the network configuration information for a node.

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

Related reference

GetConfig on page 239
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.

GetNetworkConfig on page 251
You can use the `GetNetworkConfig` method to display the network configuration information for a node.

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

Name	Description	Type
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

Related reference

GetConfig on page 239
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.

GetNetworkConfig on page 251
You can use the `GetNetworkConfig` method to display the network configuration information for a node.

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

Name	Description	Type
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

Related reference

GetSnmpInfo on page 368

You can use the `GetSnmpInfo` method to get the current simple network management protocol (SNMP) configuration information.

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

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

Name	Description	Type
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
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
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

Name	Description	Type
virtualNetworks	Object containing virtual network IP addresses and IDs.	*virtualNetwork* array

Related reference

ListActiveNodes on page 144
You can use the `ListActiveNodes` method to return the list of currently active nodes that are in the cluster.

ListAllNodes on page 144
You can use the `ListAllNodes` method to list active and pending nodes in the cluster.

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
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

Name	Description	Type
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

Related reference

GetNodeStats on page 143

You can use the `GetNodeStats` method to retrieve the high-level activity measurements for a single node.

ListNodeStats on page 151

You can use the `ListNodeStats` method to view the high-level activity measurements for all storage nodes in a storage cluster.

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
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

Related reference

ListPendingActiveNodes on page 156

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.

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
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

Related reference

AddNodes on page 121
You can use the `AddNodes` method to add one or more new nodes to a cluster.

ListPendingNodes on page 155
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.

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

Name	Description	Type
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
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

Related reference

ListProtocolEndpoints on page 520

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.

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

Name	Description	Type
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

Related reference

GetDefaultQoS on page 421

You can use the `GetDefaultQoS` method to get the default quality of service (QoS) values for a newly created volume.

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	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

Related reference

GetQoSPolicy on page 422

You can use the `GetQoSPolicy` method to get details about a specific QoS policy from the system.

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
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.	JSON object
scheduleName	The unique name assigned to the schedule.	string
scheduleType	Only schedule types of `snapshot` are supported at this time.	string
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

Related reference

ListSchedules on page 501

You can use `ListSchedules` to get information about all scheduled snapshots that have been created.

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
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

Related reference

ListFibreChannelSessions on page 188

You can use the `ListFibreChannelSessions` method to list information about the Fibre Channel sessions on a cluster.

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
accountName	The name of the account used for CHAP authentication, if any.	string
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

Name	Description	Type
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
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

Related reference

ListISCSISessions on page 152

You can use the `ListISCSISessions` method to list iSCSI connection information for volumes in the cluster.

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
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

Name	Description	Type
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

Name	Description	Type
state	The current access state of the LUN. Possible values: • online • offline • foreign_lun_error • nvfail • space_error	string
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
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
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

Name	Description	Type
policyRules	A list of objects describing the policy rules.	*snapMirrorPolicyRule* array
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
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

Name	Description	Type
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
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

Name	Description	Type
policyType	The type of the ONTAP SnapMirror policy for the relationship. See ListSnapMirrorPolicies. Examples are: “async_mirror” or “mirror_vault”.	string
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

Name	Description	Type
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

Name	Description	Type
vserverType	The type of Vserver. Possible values: • data • admin • system • node	string
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.	string
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

Name	Description	Type
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
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

Related reference

ListSnapshots on page 502

You can use `ListSnapshots` to return the attributes of each snapshot taken on the volume.

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
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

Related reference

ListStorageContainers on page 521

You can use the `ListStorageContainers` method to retrieve information about all virtual volume storage containers known to the system.

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.	float
branchType	Returned for remote replication sync jobs only. Possible values: • `snapshot` • `volume`	string

Name	Description	Type
bytesPerSecond	The number of bytes the clone is processing per second. Present only if the `type` member is set to clone or slice.	float
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
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.	integer
remainingTime	The estimated time, in seconds, to complete the operation.	integer
sliceID	The ID of the slice drive being synced.	integer
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

Related reference

ListSyncJobs on page 431

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.

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
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

Name	Description	Type
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

Related reference

ListVirtualVolumeTasks on page 527

You can use the `ListVirtualVolumeTasks` method to get a list of virtual volume tasks in the system.

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

Related reference

SetSnmpInfo on page 384

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.

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

Related reference

ListVirtualNetworks on page 397

You can use the `ListVirtualNetworks` method to list all configured virtual networks for the cluster.

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

Name	Description	Type
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
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*

Related reference

ListVirtualVolumes on page 525
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.

snapshot on page 72
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.

storageContainer on page 75
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.

volume on page 80
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.

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
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
name	The name of the volume as provided at creation time.	string

Name	Description	Type
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
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

Name	Description	Type
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

Related reference

ListActiveVolumes on page 425
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).

ListDeletedVolumes on page 428
You can use the `ListDeletedVolumes` method to retrieve the list of volumes that have been marked for deletion and purged from the system.

ListVolumes on page 435
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.

ListVolumesForAccount on page 439
You can use the `ListVolumesForAccount` method to list active and (pending) deleted volumes for an account.

QoS on page 58
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.

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

Name	Description	Type
volumes	A list of VolumeIDs belonging to the volume access group.	integer array

Related reference

ListVolumeAccessGroups on page 474

You can use the `ListVolumeAccessGroups` method to get information about the volume access groups that are currently in the system.

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
remoteVolumeName	The name of the remote volume.	string
volumePairUUID	A universally unique, cluster-defined identifier for this pairing in a canonical format.	string

Related reference

ListActivePairedVolumes on page 301

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.

ListActiveVolumes on page 425

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).

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
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
clusterUtilization	The amount of cluster capacity being utilized.	N/A	float
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

Name	Description	Calculation	Type
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
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
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
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

Name	Description	Calculation	Type
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
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
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

Name	Description	Calculation	Type
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

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

Name	Description	Type	Default value	Required
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
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.

Note: 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

Note: 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

Related reference

GetAsyncResult on page 95

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.

Account API methods

Account methods enable you to add, remove, view, and modify account and security information.

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
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*

Name	Description	Type
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
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

Name	Description	Type	Default value	Required
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: `active`: Account is active and connections are allowed. `locked`: Account is locked and connections are refused.	string	None	No
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

Related reference

DeleteVolume on page 415

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.

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

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

Related reference

Access control on page 544

The Element API methods available vary based on the type of access you set.

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

Name	Description	Type	Default value	Required
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

Related reference

Access control on page 544

The Element API methods available vary based on the type of access you set.

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

Name	Description	Type	Default value	Required
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

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.

Note: 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

Name	Description	Type	Default value	Required
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

Related reference

AddDrives on page 174
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.

GetAsyncResult on page 95
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.

ListAllNodes on page 144
You can use the `ListAllNodes` method to list active and pending nodes in the cluster.

ModifyVirtualNetwork on page 394
You can use the `ModifyVirtualNetwork` method to change the attributes of an existing virtual network.

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.

Note: 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.

Note: 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
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

Name	Description	Type
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
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

Related reference

ModifyClusterFullThreshold on page 157

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.

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

Related reference

GetClusterHardwareInfo on page 550

The `GetClusterHardwareInfo` method returns a response similar to the following example.

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": {},
      "encryptionAtRestState": "disabled",
      "ensemble": [
        "10.10.21.23",
        "10.10.21.24",
        "10.10.21.25"
      ],
      "mvip": "10.10.1.57",
      "mvipInterface": "Bond1G",
      "mvipNodeID": 2,
      "mvipVlanTag": "120",
      "name": "Hulk",
      "repCount": 2,
      "svip": "10.10.21.57",
      "svipInterface": "Bond10G",
      "svipNodeID": 2,
```

```
      "svipVlanTag": "0",
      "uniqueID": "17nx",
      "uuid": "8d385a91-9acf-4371-8939-b29e085191fd"
    }
  }
}
```

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
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

GetLicenseKey

On SolidFire Enterprise SDS storage clusters, you can use the `GetLicenseKey` method to retrieve the current license key for the storage cluster. The license key for a storage cluster consists of the cluster's order number combined with the serial number, and activates the capacity model you have purchased for Element software.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
orderNumber	The sales order number of the storage cluster.	string
serialNumber	The serial number of the storage cluster.	string

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetLicenseKey",
   "params": {
     },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "serialNumber": "30G56E3WV",
      "orderNumber": "33601"
     }
   }
}
```

New since version

12.2

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

Related reference

GetNodeHardwareInfo (output for Fibre Channel nodes) on page 587
The `GetNodeHardwareInfo` method for Fibre Channel nodes returns a response similar to the following example.

GetNodeHardwareInfo (output for iSCSI) on page 586
The `GetNodeHardwareInfo` method for iSCSI returns a response similar to the following example.

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

Related reference

ListActiveNodes on page 600

The `ListActiveNodes` method returns a response similar to the following example.

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": [
      {
        "clusterFaultID": 1,
        "code": "notUsingLACPBondMode",
        "data": null,
        "date": "2016-04-26T14:57:04.275286Z",
        "details": "Bond1G interfaces found not using LACP bond mode. Nodes not set to
LACP bond mode: {1,2,4,5}",
        "driveID": 0,
        "driveIDs": [],
        "nodeHardwareFaultID": 0,
        "nodeID": 0,
        "resolved": false,
        "resolvedDate": "",
        "serviceID": 0,
        "severity": "bestPractice",
        "type": "cluster"
      },
      {
        "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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

Name	Description	Type	Default value	Required
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
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":
[5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,24,25,26,27,28,29,30,31,40,41,42,43,44,45,46,47,5
2,53,54,55,56,57,58,59,60],
                  "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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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",
        "createTime": "2017-04-11T03:39:32.030291Z",
        "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

Related reference

AddNodes on page 121

You can use the `AddNodes` method to add one or more new nodes to a cluster.

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:

Note: 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

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 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

Name	Description	Type
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

SetLicenseKey

On SolidFire Enterprise SDS storage clusters, you can use the `SetLicenseKey` method to set the license key for the storage cluster. The license key for a storage cluster consists of the cluster's order number combined with the serial number, and activates the capacity model you have purchased for Element software.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
orderNumber	The new sales order number for this storage cluster.	string	None	Yes
serialNumber	The new serial number for this storage cluster.	string	None	Yes

Return values

This method has the following return values:

Name	Description	Type
orderNumber	The new sales order number of the storage cluster.	string
serialNumber	The new serial number of the storage cluster.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetLicenseKey",
  "params": {
    "orderNumber": "33601",
    "serialNumber": "30G56E3WV"   },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
   "id" : 1,
   "result" : {
      "serialNumber": "30G56E3WV",
      "orderNumber": "33601"
     }
   }
}
```

New since version

12.2

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.	string	30 minutes	Yes

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
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

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.

Note: 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 `true` on SolidFire Enterprise SDS clusters. Defaults to `false` on all other clusters.	boolean	`true`	No
mvip	Floating (virtual) IP address for the cluster on the management network.	string	None	Yes
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

Name	Description	Type	Default value	Required
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

Related reference

GetBootstrapConfig on page 171

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.

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
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

Related reference

CreateCluster on page 169

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.

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

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.

Note: 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

Related reference

GetAsyncResult on page 95

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.

ListDrives on page 178

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.

ListSyncJobs on page 431

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.

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

Related reference

ListDrives on page 178

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.

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

Related reference

ListDrives on page 178

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.

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

Related reference

GetDriveStats on page 177

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.

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. Any data on the drives is removed and migrated to other drives in the cluster before the drive is removed from the cluster.

`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

Related reference

GetAsyncResult on page 95

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.

ListDrives on page 178

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.

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

Related reference

GetAsyncResult on page 95

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.

ListDrives on page 178

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.

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

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.

Note: 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

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 this initiator. (boolean) • `targetSecret`: (Optional) The CHAP secret used to authenticate the target (when using mutual CHAP authentication). Defaults to a randomly generated secret if not specified during creation and `requireChap` is true. (string) • `virtualNetworkIDs`: (Optional) The list of virtual network identifiers associated with this initiator. If you define one or more virtual networks, this initiator is only able to login to those virtual networks. If you don't define any virtual networks, this initiator can login to all networks. • `volumeAccessGroupID`: (Optional) The ID of the volume access group to which this newly created initiator will be added. (integer)	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

Related reference

ListInitiators on page 197

You can use the `ListInitiators` method to get the list of initiator IQNs or World Wide Port Names (WWPNs).

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 target (when using mutual CHAP authentication). (string) • `virtualNetworkIDs`: (Optional) The list of virtual network identifiers associated with this initiator. If you define one or more virtual networks, this initiator is only able to login to those virtual networks. If you don't define any virtual networks, this initiator can login to all networks. • `volumeAccessGroupID`: (Optional) The ID of the volume access group to which the initiator should be added. If the initiator was previously in a 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)	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:

```
{
  "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

Related reference

CreateInitiators on page 193
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.

DeleteInitiators on page 196
You can use `DeleteInitiators` to delete one or more initiators from the system (and from any associated volumes or volume access groups).

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

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

Related reference

Access control on page 544

The Element API methods available vary based on the type of access you set.

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
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

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
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 auth session. If the calling user is not in the ClusterAdmins / Administrator AccessGroup, only auth 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

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

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
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

Name	Description	Type	Default value	Required
sourceAddressV6	The source IPv6 address to use in the ICMP ping packets.	string	None	Yes
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
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": "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": {
    "details": {
      "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
      }
    },
    "duration": "00:00:00.244379",
    "result": "Passed"
  }
}
```


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

Name	Description	Type
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. When a bundle is created, it is stored on the node as a `.tar.gz` file. You can only run this method on a management node; it does not work when run on a storage node.

Parameters

Note: 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
extraArgs	This parameter is fed to the sf_make_support_bundle script. This parameter should be used only at the request of NetApp Support.	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 the `CreateSupportBundle` to create a support bundle file under the node's directory. After creation, the bundle is stored on the node as a tar.gz file.

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

Name	Description	Type	Default value	Required
extraArgs	This parameter is passed to the sf_make_support_bundle script. This parameter should be used only at the request of NetApp Support.	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 `CreateSupportBundle` API 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": {}
 },
"id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
"result": {
  "details": {
    "bundleName": "supportbundle",
    "extraArgs": "",
    "files": [
         "supportbundle.SF-D220.tar.gz"
     ],
     "output": "timeout -s KILL 1500s /sf/scripts/sf_make_support_bundle --quiet /tmp/
solidfire- dtemp.bYRag7/supportbundle timeout -s KILL 1500s pigz --verbose --fast \"/tmp/
```

```
solidfire-dtemp.bYRag7/supportbundle.*.tar\" /tmp/solidfire-dtemp.bYRag7/supportbundle.SF-
D220.tar to /tmp/solidfire-dtemp.bYRag7/supportbundle.SF-D220.tar.gz Moved '/tmp/
solidfire-dtemp.bYRag7/supportbundle.SF-D220.tar.gz' to /tmp/supportbundles",
      "timeoutSec": 1500,
      "url": [
          "https://192.168.130.155:442/config/supportbundles/supportbundle.SF-D220.tar.gz"
      ]
    },
    "duration": "00:00:21.166692",
    "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

Related 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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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
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

Related information

NetApp HCI storage maintenance mode concepts

EnableSsh

You can use the `EnableSsh` method to enable the 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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

Related reference

GetClusterConfig on page 237
You can use the `GetClusterConfig` API method to return information about the cluster configuration the node uses to communicate with its cluster.

GetNetworkConfig on page 251
You can use the `GetNetworkConfig` method to display the network configuration information for a node.

GetConfig on page 548
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.

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.

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 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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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": {
      "nodes": [
        {
          "nodeID": 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

Related reference

GetNetworkConfig on page 583

The `GetNetworkConfig` method returns a response similar to the following example.

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

Note: 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
\nVQQGEwJVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G
\nA1UEChMYV2hhdCBIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN
\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD
\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdh
\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZl
\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28fnLKQNWEWMR
\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FHP4\nt6P/
Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC
\nmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3Jmoyx+zj/
Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS
\n7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq
+itawIDAQABo4HrMIHoMB0GA1UdDgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB
\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrnDJyTsdnEbaGBiaSBhjCBgzELMAkG
\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxWZWdhcywgQmFieSExITAf
\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqGSIb3DQEJARYe
\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYDVR0T\nBAUwAwEB/
zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe
\nMqo5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N
\niebEgAaE5eWvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl
+DwkxnS1cx1vERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR
\nHgMoPDym099kvc84B1k7HkDGHpr4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG
+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDgl0QoQDWNDoTeRBsz80nwiouA==\n-----END CERTIFICATE-----\n",
        "details": {
            "issuer": "/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org",
            "modulus":
"F14FB6F1F9CB290356116311E9A91E0CAB9E852A52EFDA1D2C68A0235F2A94257F01463964B8EAB138C1BD325
546FE38CA809380DAF1DFA53B1473F8B7A3FF4A2D1A62BE28BF1979C03A44337432CB924F07B25E94E07A003ED
F9A24F078FDB41D162966F63E533ECB6041429AB829199405DE239221C047B4B284E75F3A2554FA8F9760EB28D
41903B7E76CA573D1D71DC9FA95BFE3CA5D0399535467471A430026212DC99A8CB1FB38FF61AE162AAFB64AA4C
05FB6D7D05DF01C77D79D99479CCF1F113E4DFFD03E2BA952EDD83D7325EEE1A7D77202B2D78262341BEA6C18E
1809B44EFAC80CBAAD31EED313378E376471BF58F2688DCF117E002ABE8AD6B",
            "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"
        ]
    }
    }
```

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

Note: 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

Related reference

EnableEncryptionAtRest on page 316

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.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

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

Note: 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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

Note: 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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

Related reference

SetClusterConfig on page 272
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.

SetNetworkConfig on page 274
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.

GetConfig on page 548
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.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

Related reference

GetNetworkConfig on page 251
You can use the `GetNetworkConfig` method to display the network configuration information for a node.

GetNetworkConfig on page 583

The `GetNetworkConfig` method returns a response similar to the following example.

SetNodeSSLCertificate

You can use the `SetNodeSSLCertificate` method to set a user SSL certificate and private key for the management node.

Parameters

Note: 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
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\nZLjqsTjBvTJVRv44yoCTgNrx36U7FHP4t6P/
```

```
Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP20HRYpZvY
+Uz7LYEFCmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg\n6yjUGQO352ylc9HXHcn6lb/
jyl0DmVNUZ0caQwAmIS3Jmoyx+zj/Ya4WKq+2SqTA\nX7bX0F3wHHfXnZlHnM8fET5N/9A
+K6lS7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE\n76yAy6rTHu0xM3jjdkcb9Y8miNzxF+ACq
+itawIDAQABAoIBAH1jlIZr6/sltqVW\nO0qVC/49dyNu
+KWVSq92ti9rFe7hBPueh9gklh78hP9QlitLkir3YK4GFsTFUMux\n7z1NRCxA/
4LrmLSkAjW2kRXDfVl2bwZq0ua9NefGw92O8D2OZvbuOxk7Put2p6se
\nfgNzSjf2SI5DIX3UMe5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7wnq\nt+Am5/
Vu1LI6rGiG6oHEW0oGSuHl1esIyXXa2hqkU+1+iF2iGRMTiXac4C8d11NU\nWGIRCXFJAmsAQ+hQm7pmtsKdEqumj/
PIoGXf0BoFVEWaIJIMEgnfuLZp8IelJQXn\nSFJbk2ECgYEA
+d5ooU4thZXylWHUZqomaxyzOruA1T53UeH69HiFTrLjvfwuaiqj\nlHzPlhms6hxexwz1dzApgog/NOM
+2bAc0rn0dqvtV4doejtlDZKRqrNCf/cuN2QX
\njaCJClCWau3sEHCckLOhWeY4HaPSoWq0GKLmKkKDChB4nWUYg3gSWQkCgYEA9zuN\nHW8GPS+yjixeKXmkKO0x/
vvxzR+J5HH5znaIHss48THyhzXpLr+v30Hy2h0yAlBS\nny5Ja6wsomb0mVe4NxVtVawg2E9vVvTa1UC
+TNmFBBuLRPfjcnjDerrSuQ5lYY+M\nC9MJtXGfhp//
G0bzwsRzZxOBsUJb15tppaZIs9MCgYAJricpkKjM0xlZ1jdvXsos
\nPilnbho4qLngrzuUuxKXEPEnzBxUOqCpwQgdzZLYYw788TCVVIVXLEYem2s07dDA\nDTo
+WrzQNkvC6IgqtXH1RgqegIoG1VbgQsbsYmDhdaQ+os4+AOeQXw3vgAhJ/qNJ
\njQ4Ttw3ylt7FYkRH26ACWQKBgQC74Zmf4JuRLAo5WSZFxpcmMvtnlvdutqUH4kXA\nzPssy6t
+QELa1fFbAXkZ5Pg1ITK752aiaX6KQNG6qRsA3VS1J6drD9/2AofOQU17\n
+jOkGzmmoXf49Zj3iSakwg0ZbQNGXNxEsCAUr0BYAobPp9/fB4PbtUs99fvtocFr\njS562QKBgCb
+JMDP5q7jpUuspj0obd/ZS+MsomE+gFAMBJ71KFQ7KuoNezNFO+ZE
\n3rnR8AqAm4VMzqRahs2PWNe2Hl4J4hKu96qNpNHbsW1NjXdAL9P7oqQIrhGLVdhX
\nInDXvTgXMdMoet4BKnftelrXFKHgGqXJoczq4JWzGSIHNgvkrH60\n-----END RSA PRIVATE KEY-----\n",
        "certificate": "-----BEGIN CERTIFICATE-----\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/
zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD
\nVQQGEwJVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G
\nA1UEChMYV2hhdCBIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN
\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD
\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdh
\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZl
\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28fnLKQNWEWMR
\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FHP4\nt6P/
Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC
\nmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3Jmoyx+zj/
Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS
\n7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq
+itawIDAQABo4HrMIHoMB0GA1UdDgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB
\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrnDJyTsdnEbaGBiaSBhjCBgzELMAkG
\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxWZWdhcywgQmFieSExITAf
\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqGSIb3DQEJARYe
\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYDVR0T\nBAUwAwEB/
zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe
\nMqo5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N
\niebEgAaE5eWvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl
+DwkxnS1cx1vERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR
\nHgMoPDym099kvc84B1k7HkDGHpr4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG
+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDgl0QoQDWNDoTeRBsz80nwiouA==\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

Note: 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 a node that has not yet been added to a cluster. To use this method, login in to the MIP for the pending node and enter the `shutdown` method with either the `restart` or `halt` options in the following table.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

Parameter

CAUTION: This method restarts all networking services on a node, causing temporary loss of networking connectivity.

This method has the following input parameter:

Name	Description	Type	Default value	Required
option	Action to take for the node shutdown: • `restart`: Restarts the node. • `halt`: Performs full power-off of the node.	string	None	Yes

Note: You need to use port 443 to access the API service on an individual node. See the following HTTPS API request example, replacing <MIP> with the management IP address of the individual node, and <API version> with the version of Element software running on the node:

```
https://<MIP>:443/json-rpc/<API version>/?method=Shutdown&option=halt
```

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
   "method": "Shutdown",
   "params": {
      "option": "restart"
   },
```

```
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{ "id" : 1,
  "result" : {}
}
```

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) ◦ `result`: Result of the test as a whole (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
}
```

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) ◦ `result`: Result of the test as a whole (string)	string

Request example

Requests for this method are similar to the following example:

```
{
   "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.

Note: 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

Name	Description	Type	Default value	Required
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

Note: 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

Name	Description	Type	Default value	Required
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

Related reference

TestHardwareConfig on page 609

The `TestHardwareConfig` method returns a response similar to the following example.

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

Related reference

SetNetworkConfig on page 274

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.

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
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

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	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
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

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.

  Note: 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.

Related reference

StartClusterPairing on page 305
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.

CompleteClusterPairing on page 298
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.

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.

Related reference

StartVolumePairing on page 307

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.

CompleteVolumePairing on page 299

You can use `CompleteVolumePairing` to complete the pairing of two volumes.

ModifyVolumePair on page 303

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).

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" :
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223a223163656131333
6322d346338662d343631612d626537322d373435363661393533643266222c22636c7573746572556e6971756
54944223a2278736d36222c226d766970223a223139322e3136382e3133392e313232222c226e616d65223a224
175746f54657374322d63307552222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222
c22727063436f6e6e656374696f6e4944223a393133313432363439 2c22757365726e616d65223a225f5f53465
f706169725f50597a796647704c7246564432444a42227d"
   },
   "id" : 1
}
```


Response example

This method returns a response similar to the following example:


```
{
   "id" : 1,
   "result" : {
       "clusterPairID" : 1
   }
}
```


New since version

9.6

Related reference

StartClusterPairing on page 305

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.

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" :
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223a223163656131333
6322d346338662d343631612d626537322d373435363661393533643266222c22636c7573746572556e6971756
54944223a2278736d36222c226d766970223a223139322e3136382e3133392e313232222c226e616d65223a224
175746f54657374322d63307552222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222
c22727063436f6e6e656374696f6e4944223a393133313432363439 2c22757365726e616d65223a225f5f53465
f706169725f50597a796647704c7246564432444a42227d"
        },
   "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
"result": {}
}
```

New since version

9.6

Related reference

StartVolumePairing on page 307

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.

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

Note: 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":
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223a223163656131333
6322d346338662d343631612d626537322d373435363661393533643266222c22636c7573746572556e6971756
54944223a2278736d36222c226d766970223a223139322e3136382e3133392e313232222c226e616d65223a224
175746f54657374322d63307552222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222
c22727063436f6e6e656374696f6e4944223a393133313432363439 2c22757365726e616d65223a225f5f53465
f706169725f50597a796647704c7246564432444a42227d"
   }
}
```

New since version

9.6

Related reference

CompleteClusterPairing on page 298

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.

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" :
"7b226d766970223a223139322e3136382e3133392e313232222c22766f6c756d654944223a312c22766f6c756
d654e616d65223a2254657374222c22766f6c756d6550616972555549442 23a22363936323466636 22d3230326
52d343332352d613536392d656339633635356337623561227d"
      }
}
```

New since version

9.6

Related reference

CompleteVolumePairing on page 299

You can use `CompleteVolumePairing` to complete the pairing of two volumes.

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

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

Name	Description	Type	Default value	Required
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 (C).	string	None	No
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.

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

Related reference

GetClusterInfo on page 133

You can use the `GetClusterInfo` method to return configuration information about the cluster.

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.

When you enable Encryption at Rest, the cluster automatically manages encryption keys internally for the drives on each node in the cluster. Nodes do not store the keys to unlock drives and the keys are never passed over the network. Two nodes participating in a cluster are required to access the key to disable encryption on a drive. The encryption management does not affect performance or efficiency on the cluster. If an encryption-enabled drive or node is removed from the cluster with the API, Encryption at Rest is disabled and the data is not secure erased. Data can be secure erased using the `SecureEraseDrives` API method.

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.

Note: 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".

Note: 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.

Note: 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

Related reference

SecureEraseDrives on page 182

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.

GetClusterInfo on page 133

You can use the `GetClusterInfo` method to return configuration information about the cluster.

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

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

Name	Description	Type	Default value	Required
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
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

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	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

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

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

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

Related reference

BreakSnapMirrorVolume on page 331

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.

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

Name	Description	Type	Default value	Required
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

Related reference

BreakSnapMirrorRelationship on page 330

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.

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

Name	Description	Type	Default value	Required
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
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	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
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

Name	Description	Type	Default value	Required
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

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
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

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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

Related reference

GetLldpInfo on page 560

The `GetLldpInfo` method returns a response similar to the following example.

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

Related reference

GetNvramInfo on page 593

The `GetNvramInfo` method returns a response similar to the following example.

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

Related reference

SetRemoteLoggingHosts on page 382

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.

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
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

Note: `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

Name	Description	Type
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

Related reference

GetSnmpState on page 370
You can use the `GetSnmpState` method to get the current state of the SNMP feature.

SetSnmpACL on page 383
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`.

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

Related reference

SetSnmpACL on page 383

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`.

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
\nVQQGEwJVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G
\nA1UEChMYV2hhdCBIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN
\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD
\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdh
\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZl
\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28fnLKQNWEWMR
\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FHP4\nt6P/
Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC
\nmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3Jmoyx+zj/
Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS
```

```
\n7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq
+itawIDAQABo4HrMIHoMB0GA1UdDgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB
\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrnDJyTsdnEbaGBiaSBhjCBgzELMAkG
\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxWZWdhcywgQmFieSExITAf
\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqGSIb3DQEJARYe
\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYDVR0T\nBAUwAwEB/
zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe
\nMqo5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N
\niebEgAaE5eWvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl
+DwkxnS1cx1vERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR
\nHgMoPDym099kvc84B1k7HkDGHpr4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG
+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDgl0QoQDWNDoTeRBsz80nwiouA==\n-----END CERTIFICATE-----\n",
        "details": {
            "issuer": "/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org",
            "modulus":
"F14FB6F1F9CB290356116311E9A91E0CAB9E852A52EFDA1D2C68A0235F2A94257F01463964B8EAB138C1BD325
546FE38CA809380DAF1DFA53B1473F8B7A3FF4A2D1A62BE28BF1979C03A44337432CB924F07B25E94E07A003ED
F9A24F078FDB41D162966F63E533ECB6041429AB829199405DE239221C047B4B284E75F3A2554FA8F9760EB28D
41903B7E76CA573D1D71DC9FA95BFE3CA5D0399535467471A430026212DC99A8CB1FB38FF61AE162AAFB64AA4C
05FB6D7D05DF01C77D79D99479CCF1F113E4DFFD03E2BA952EDD83D7325EEE1A7D77202B2D78262341BEA6C18E
1809B44EFAC80CBAAD31EED313378E376471BF58F2688DCF117E002ABE8AD6B",
            "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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

Parameters

Note: 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
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.

Note: 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

Related reference

GetRemoteLoggingHosts on page 366
You can use the `GetRemoteLoggingHosts` method to get the current list of log servers.

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`.

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

Related reference

SetSnmpInfo on page 384

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.

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

Note: `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

Name	Description	Type	Default value	Required
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
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.

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

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" : "SetSSLCertificate",
    "params" : {
        "privateKey": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEA8U
+28fnLKQNWEWMR6akeDKuehSpS79odLGigI18qlCV/AUY5\nZLjqsTjBvTJVRv44yoCTgNrx36U7FHP4t6P/
Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP20HRYpZvY
+Uz7LYEFCmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg\n6yjUGQO352ylc9HXHcn6lb/
jyl0DmVNUZ0caQwAmIS3Jmoyx+zj/Ya4WKq+2SqTA\nX7bX0F3wHHfXnZlHnM8fET5N/9A
+K6lS7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE\n76yAy6rTHu0xM3jjdkcb9Y8miNzxF+ACq
+itawIDAQABAoIBAH1jlIZr6/sltqVW\nO0qVC/49dyNu
+KWVSq92ti9rFe7hBPueh9gklh78hP9QlitLkir3YK4GFsTFUMux\n7z1NRCxA/
4LrmLSkAjW2kRXDfVl2bwZq0ua9NefGw92O8D2OZvbuOxk7Put2p6se
\nfgNzSjf2SI5DIX3UMe5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7wnq\nt+Am5/
Vu1LI6rGiG6oHEW0oGSuHl1esIyXXa2hqkU+1+iF2iGRMTiXac4C8d11NU\nWGIRCXFJAmsAQ+hQm7pmtsKdEqumj/
PIoGXf0BoFVEWaIJIMEgnfuLZp8IelJQXn\nSFJbk2ECgYEA
+d5ooU4thZXylWHUZqomaxyzOruA1T53UeH69HiFTrLjvfwuaiqj\nlHzPlhms6hxexwz1dzApgog/NOM
+2bAc0rn0dqvtV4doejtlDZKRqrNCf/cuN2QX
\njaCJClCWau3sEHCckLOhWeY4HaPSoWq0GKLmKkKDChB4nWUYg3gSWQkCgYEA9zuN\nHW8GPS+yjixeKXmkKO0x/
vvxzR+J5HH5znaIHss48THyhzXpLr+v30Hy2h0yAlBS\nny5Ja6wsomb0mVe4NxVtVawg2E9vVvTa1UC
+TNmFBBuLRPfjcnjDerrSuQ5lYY+M\nC9MJtXGfhp//
G0bzwsRzZxOBsUJb15tppaZIs9MCgYAJricpkKjM0xlZ1jdvXsos
\nPilnbho4qLngrzuUuxKXEPEnzBxUOqCpwQgdzZLYYw788TCVVIVXLEYem2s07dDA\nDTo
+WrzQNkvC6IgqtXH1RgqegIoG1VbgQsbsYmDhdaQ+os4+AOeQXw3vgAhJ/qNJ
\njQ4Ttw3ylt7FYkRH26ACWQKBgQC74Zmf4JuRLAo5WSZFxpcmMvtnlvdutqUH4kXA\nzPssy6t
+QELa1fFbAXkZ5Pg1ITK752aiaX6KQNG6qRsA3VS1J6drD9/2AofOQU17\n
+jOkGzmmoXf49Zj3iSakwg0ZbQNGXNxEsCAUr0BYAobPp9/fB4PbtUs99fvtocFr\njS562QKBgCb
+JMDP5q7jpUuspj0obd/ZS+MsomE+gFAMBJ71KFQ7KuoNezNFO+ZE
\n3rnR8AqAm4VMzqRahs2PWNe2Hl4J4hKu96qNpNHbsW1NjXdAL9P7oqQIrhGLVdhX
\nInDXvTgXMdMoet4BKnftelrXFKHgGqXJoczq4JWzGSIHNgvkrH60\n-----END RSA PRIVATE KEY-----\n",
        "certificate": "-----BEGIN CERTIFICATE-----\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/
zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD
\nVQQGEwJVUzELMAkGA1UECBMCTlYxFTATBgNVBAcUDFZlZ2FzLCBCYWJ5ITEhMB8G
\nA1UEChMYV2hhdCBIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN
\nMjcwMzA2MjI1MDI2WjCBgzELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYD
\nVQQHFAxWZWdhcywgQmFieSExITAfBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdh
\ncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZl
\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28fnLKQNWEWMR
\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FHP4\nt6P/
Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC
\nmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3Jmoyx+zj/
Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS
\n7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nq
+itawIDAQABo4HrMIHoMB0GA1UdDgQWBBRvvBRPno5S34zGRhrnDJyTsdnEbTCB
\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrnDJyTsdnEbaGBiaSBhjCBgzELMAkG
\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQQHFAxWZWdhcywgQmFieSExITAf
```

```
\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqGSIb3DQEJARYe
\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYDVR0T\nBAUwAwEB/
zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe
\nMqo5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7JyfS37l4rRolFpX6N
\niebEgAaE5eWvB6zgiAcMRIKqu3DmJ7y3CFGk9dHOlQ+WYnoO/eIMy0coT26JBl5H\nDEwvdl
+DwkxnS1cx1vERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR
\nHgMoPDym099kvc84B1k7HkDGHpr4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG
+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDgl0QoQDWNDoTeRBsz80nwiouA==\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.

Note: SolidFire Enterprise SDS does not support this method. If you use this method on a SolidFire eSDS system, the method will return an error.

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.

Related concepts

Virtual network naming conventions on page 392

NetApp Element storage systems use monotonically increasing numbers as unique identifiers for all objects in the system.

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.

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.

Note: 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
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.

Note: 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

Name	Description	Type	Default value	Required
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.

Note: This method requires either the virtualNetworkID or the virtualNetworkTag as a parameter, but not both.

Note: 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

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

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
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
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*.

Note: 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
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

Name	Description	Type	Default value	Required
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
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

Related reference

GetAsyncResult on page 95

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.

ListSyncJobs on page 431

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.

ModifyVolume on page 449

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.

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

Related reference

GetAsyncResult on page 95

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.

ListSyncJobs on page 431

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.

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 creation is complete, 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
accountID	The ID of the account that owns this volume.	integer	None	Yes

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	`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
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
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

Name	Description	Type	Default value	Required
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
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

Related reference

GetDefaultQoS on page 421

You can use the `GetDefaultQoS` method to get the default quality of service (QoS) values for a newly created volume.

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.

Note: 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*

Name	Description	Type
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

Name	Description	Type
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

Name	Description	Type	Default value	Required
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
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":200278016remote0,
           "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
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.

Note: 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
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.	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
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	`false`	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 this volume. If not specified, the QoS settings are not changed. Possible values: • `minIOPS` • `maxIOPS` • `burstIOPS`	*QoS*	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
setCreateTime	Set to `true` to change the recorded date of volume creation.	boolean	None	No
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

Related reference

GetDefaultQoS on page 421

You can use the `GetDefaultQoS` method to get the default quality of service (QoS) values for a newly created volume.

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.

Note: 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

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
enableSnapMirrorReplication	Determines whether the volume can be used for replication with SnapMirror endpoints. Possible values: • `true` • `false`	boolean	`false`	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

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
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

Related reference

GetDefaultQoS on page 421

You can use the `GetDefaultQoS` method to get the default quality of service (QoS) values for a newly created volume.

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

Related reference

DeleteVolume on page 415

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.

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.

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

Name	Description	Type	Default value	Required
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

Related reference

DeleteVolumes on page 417

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.

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
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.

Note: 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
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	JSON attributes for the bulk volume job.	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

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	JSON attributes for the bulk volume job.	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

Related reference

StartBulkVolumeRead on page 462
You can use the `StartBulkVolumeRead` method to start a bulk volume read session on a specified volume.

StartBulkVolumeWrite on page 464
You can use the `StartBulkVolumeWrite` method to start a bulk volume write session on a specified volume.

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

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.

Note: 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 other 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*

Name	Description	Type
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

Related reference

GetAsyncResult on page 95

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.

ListSyncJobs on page 431

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.

ModifyVolume on page 449
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.

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
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
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

Name	Description	Type
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
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.

You can also associate a virtual network with a volume access group. This association applies to all volumes in the 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 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

Related reference

AddInitiatorsToVolumeAccessGroup on page 468
You can use the `AddInitiatorsToVolumeAccessGroup` method to add initiators to a specified volume access group.

AddVolumesToVolumeAccessGroup on page 469
You can use the `AddVolumesToVolumeAccessGroup` method to add volumes to a specified volume access group.

RemoveInitiatorsFromVolumeAccessGroup on page 477

You can use the `RemoveInitiatorsFromVolumeAccessGroup` method to remove initiators from a specified volume access group.

RemoveVolumesFromVolumeAccessGroup on page 475

You can use the `RemoveVolumesFromVolumeAccessGroup` method to remove volumes from a specified volume access group.

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

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.

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.

Note: 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
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
retention	The amount of time the snapshot will be retained. If left empty, the snapshot is retained forever. Enter in the HH:mm:ss format.	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.

Note: 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
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

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
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	An object of schedule information about how the snapshot should be created at each scheduled interval. • `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. (string)	JSON object	None	Yes
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
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.

Note: 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	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
retention	The amount of time the snapshot will be retained. If left empty, the snapshot is retained forever. Enter in the HH:mm:ss format.	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
  }
}
```

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
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

Related reference

RollbackToSnapshot on page 513

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.

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	Use to set the time when the snapshot should be removed. If no time is entered, the current time will be used. When set to "null", the snapshot never expires.	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
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	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. Valid 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: • `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. (string)	JSON object	None	No
snapMirrorLabel	The label used by SnapMirror software to specify the snapshot retention policy on a SnapMirror endpoint.	string	None	No

Name	Description	Type	Default value	Required
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	Use to set the time when the snapshot should be removed. When set to "null", the snapshot never expires.	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

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
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.

Note: 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.

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 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`.

Note: 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
volumeID	VolumeID for the volume.	integer	None	Yes

Name	Description	Type	Default value	Required
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
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

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
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

Name	Description	Type	Default value	Required
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

Cluster fault codes

If a storage cluster experiences an error or a state that might be of interest to an administrator, it generates a cluster fault. You can use the `ListClusterFaults` method to retrieve the current list of resolved and unresolved faults on a storage cluster.

The following list gives more information about and possible solutions for NetApp Element storage cluster faults:

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, purge or delete volumes or add another storage node to the storage cluster.

clockSkewExceedsFaultThreshold

The time skew between the cluster master and the node which is presenting a token exceeds the recommended threshold.

The 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 already using internal NTP servers, contact NetApp Support for assistance.

clusterCannotSync

Cluster block data is in a degraded state and the auto-heal process to restore full block data redundancy cannot proceed; either too many nodes or block services are offline, or the cluster block services are too full.

To resolve this fault, add more block capacity or contact NetApp Support.

clusterFull

There is no more free storage space in the storage cluster.

To resolve this fault, add more storage.

clusterIOPSAreOverProvisioned

Storage cluster IOPS are over provisioned. The sum of all minimum QoS IOPS is greater than the expected IOPS of the cluster. The system cannot maintain minimum QoS for all volumes simultaneously.

To resolve this fault, lower the minimum QoS IOPS settings for volumes.

disableDriveSecurityFailed

A drive was unable to be secure disabled when the Encryption at Rest feature is turned off. The drive still has drive security enabled.

The reason that drive security could not be disabled is shown in the fault details; you might need to investigate the problem based on the reason. If you need to recover a disk that does not successfully disable security, use the following steps:

1. Logically remove the drive by moving it to "available" status.
2. Perform a secure erase on the drive.
3. Move the drive to "active" status.

If these steps do not resolve the issue, replace the drive.

disconnectedClusterPair

A cluster pair is disconnected or configured incorrectly.

Check the network connectivity of the cluster.

disconnectedRemoteNode

A remote node is 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 to be added in the storage cluster. In general, all storage 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

There is more than one storage cluster master candidate.

Contact NetApp Support for assistance.

enableDriveSecurityFailed

A drive was unable to be secure enabled when the Encryption at Rest feature is turned on.

Ensure that the correct key is being used to enable security. If you need to recover a disk that does not successfully enable security, use the following steps:

1. Logically remove the drive by moving it to "available" status.
2. Perform a secure erase on the drive.
3. Move the drive to "active" status.

If these steps do not resolve the issue, replace the drive.

ensembleDegraded

One of the ensemble nodes has lost network connectivity or power.

To resolve this fault, restore network connectivity or power to the affected node.

exception

An unusual fault has occurred. 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.

Contact NetApp Support for assistance.

fanSensor

A fan sensor has failed or is missing.

Replace any failed hardware in the node. If this does not resolve the issue, contact NetApp Support for assistance.

fibreChannelAccessDegraded

A Fibre Channel node is not responding to other nodes in the storage cluster via its storage IP address.

Check the network connectivity of the cluster.

fibreChannelAccessUnavailable

All Fibre Channel nodes are unresponsive. The node IDs are displayed.

Check the network connectivity of the cluster.

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.

fipsDrivesMismatch

A non-FIPS drive has been inserted into a FIPS storage node or a FIPS drive has been inserted into a non-FIPS storage node.

Remove or replace the drive or drives in question.

fipsDrivesOutOfCompliance

The system has detected that Encryption at Rest is disabled, or non-FIPS hardware is present in the storage cluster.

Enable Encryption at Rest or remove the non-FIPS hardware from the storage cluster.

fipsSelfTestFailure

The system has detected a failure during the FIPS 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.
- A node is using unsupported drive.
- There is a drive firmware mismatch.
- A drive's encryption capable state does not match its parent node.

Contact NetApp Support for assistance.

idPCertificateExpiration

The cluster’s service provider SSL certificate for use with a third-party identity provider 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` 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.

To resolve this fault, modify the bond modes in the per-node web UI.

inconsistentInterfaceConfiguration

The interface configuration is inconsistent.

To resolve this fault, ensure the node interfaces in the storage cluster are consistently configured.

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.

To resolve this fault, modify the MTU settings in the per-node web UI.

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.

To resolve this fault, modify the subnet mask in the Element (storage cluster) web UI.

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)

The root Certification Authority (CA) certificate is nearing expiration. 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.

To resolve this fault, update the certificate before it expires. Acquire a new certificate from the root CA with expiration date at least 30 days in the future. Use the `ModifyKeyServerKmip` API method to provide the updated root CA certificate.

kmipCertificateFault (Client certificate is nearing expiration)

The client certificate is nearing expiration. 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.

To resolve this fault, create a new CSR with the `GetClientCertificateSigningRequest` method. Have the CSR signed with an expiration greater than 30 days and then use the `ModifyKeyServerKmip` API method to replace the expiring KMIP client certificate with the new certificate.

kmipCertificateFault (Root Certification Authority (CA) certificate expired)

The root CA certificate has expired.

Acquire a new certificate from the root CA with expiration date at least 30 days in the future. Use the `ModifyKeyServerKmip` API method to provide the updated root CA certificate.

kmipCertificateFault (Client certificate expired)

The client certificate has expired.

Create a new CSR using the `GetClientCertificateSigningRequest` API method and have it signed making sure new expiration date is at least 30 days in the future. Use the `ModifyKeyServerKmip` API method to replace the expired client certificate with the new certificate.

kmipCertificateFault (Invalid root certification authority (CA) certificate)

The root CA certificate is invalid.

Make sure that the correct certificate was provided. If needed, reacquire the certificate from the root CA. Use the `ModifyKeyServerKmip` API method to install the correct certificate.

kmipCertificateFault (Invalid client certificate)

The client certificate is invalid.

Make sure that the correct KMIP client certificate is installed. The root CA of the client certificate should be installed on the external key management server. If you need to update the client certificate, use the `ModifyKeyServerKmip` API method to do so.

kmipServerFault (Connection failure)

One or more of the nodes cannot access the external key management server.

The key server ID is provided in the fault details. Ensure that the server is functional and reachable via the management network. If only some nodes are unable to access the external key management server, the nodes that are unable to reach the key server are listed in the fault details. Perform troubleshooting at the network or specific node level to determine why only some of the nodes can access the external key management server.

kmipServerFault (Authentication failure)

One or more of the nodes cannot authenticate with the external key management server.

Ensure that the correct root CA and KMIP client certificates are in use. If you need to update any of the certificates, use the `ModifyKeyServerKmip` method to install the correct certificate.

kmipServerFault (Server error)

The external key management server has an error.

The error details are provided in the fault details. You might need to troubleshoot the external key management server based on the error.

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:

Note: See the **Details** heading for 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.

See *Understanding cluster fullness levels* for more information.

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 network interface is not present.
- A duplicate network interface is present.
- A network interface is configured but down.
- A network interface restart is needed.

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.

To resolve this fault, check network connectivity and network cabling of the cluster. If there are no network problems, contact NetApp Support for a node replacement.

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 NTP server configuration, 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 has no input or the input is out of range.

To resolve this fault, verify that redundant power is supplied to all nodes. Contact NetApp Support if the issue persists.

provisionedSpaceTooFull

The overall provisioned capacity of the storage cluster is too full.

To resolve this fault, add more provisioned space, or delete and purge volumes or snapshots.

remoteRepAsyncDelayExceeded

The configured asynchronous delay for replication has been exceeded.

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, remove some snapshots on the remote 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

The Baseboard Management Controller (BMC) self-test failed or a sensor could not communicate with the 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 storage nodes or contact NetApp Support.

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 node or nodes.

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.

unbalancedMixedNodes

A single node accounts for more than one-third of the storage cluster's capacity.

Contact NetApp Support for assistance.

unresponsiveService

A system 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 network mask.
- 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 yet completely replicated and synchronized.

This fault is cleared when the synchronisation is complete.

If the fault persists, check for network connectivity issues and hardware errors.

volumesOffline

One or more volumes in the storage cluster are offline.

Contact NetApp Support for assistance.

Related concepts

Contact NetApp Support on page 614
If you need help with or have questions or comments about NetApp products, contact NetApp Support.

Related reference

GetClusterFullThreshold on page 128

You can use the `GetClusterFullThreshold` method to view the stages set for cluster fullness levels. This method returns all fullness metrics for the cluster.

ListClusterFaults on page 146

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.

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

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"
                }
            }
        ]
    }
    }
```

Contact NetApp Support

If you need help with or have questions or comments about NetApp products, contact NetApp Support.

- Web:
  mysupport.netapp.com
- Phone:
  - 888.4.NETAPP (888.463.8277) (US and Canada)
  - 00.800.44.638277 (EMEA/Europe)
  - +800.800.80.800 (Asia/Pacific)

Copyright and trademark

Copyright

Copyright © 2020 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark

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.

http://www.netapp.com/us/legal/netapptmlist.aspx