■ NetApp

Cluster API methods

Element Software

NetApp
November 19, 2025

Table of Contents

Cluster API methods

Element software cluster API methods enable you to manage the configuration and topology of the storage cluster and the nodes that belong to a storage cluster.

Some cluster API methods operate on nodes that are part of a cluster, or have been configured to join a cluster. You can add nodes to a new cluster or to an existing cluster. Nodes that are ready to be added to a cluster are in a "pending" state, which means they have been configured but not yet added to the cluster.

- AddNodes
- ClearClusterFaults
- CreateClusterInterfacePreference
- DeleteClusterInterfacePreference
- EnableFeature
- GetClusterCapacity
- GetClusterFullThreshold
- GetClusterHardwareInfo
- GetClusterInfo
- GetClusterInterfacePreference
- GetClusterMasterNodeID
- GetClusterStats
- GetClusterVersionInfo
- GetFeatureStatus
- GetLoginSessionInfo
- GetNodeHardwareInfo
- GetNodeStats
- ListActiveNodes
- ListAllNodes
- ListClusterFaults
- ListClusterInterfacePreferences
- ListEvents
- ListNodeStats
- ListISCSISessions
- ListServices
- ListPendingNodes
- ListPendingActiveNodes
- ModifyClusterFullThreshold
- ModifyClusterInterfacePreference
- RemoveNodes

- SetLoginSessionInfo

- Shutdown

Find more information

- SolidFire and Element Software Documentation

- Documentation for earlier versions of NetApp SolidFire and Element products

AddNodes

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

When a node that is not configured starts up for the first time, you are prompted to configure the node. Once you configure the node, it is registered as a "pending node" with the cluster. Storage clusters running Element software automatically image a node to the version on the cluster. When you add a pending node, the method response includes an asyncHandle value that you can use with the `GetAsyncResult` method to query the status of the automatic imaging process.

The process of adding a Fibre Channel node is the same as adding Element iSCSI storage nodes to a cluster. Fibre Channel nodes are registered in the system with a NodeID. When they become accessible, they are put in a "pending node" status. The `ListAllNodes` method will return the pendingNodeID for iSCSI nodes as well as any Fibre Channel nodes that are available to add to the cluster.

When you add a node to a cluster that you have configured for virtual networking, the system requires a sufficient number of virtual storage IP addresses to allocate a virtual IP to the new node. If there are no virtual IP addresses available for the new node, the `AddNode` operation fails. Use the `ModifyVirtualNetwork` method to add more storage IP addresses to your virtual network.

Once you add a node, any drives on the node are made available and you can add them using the `AddDrives` method to increase the storage capacity of the cluster.

> (i) It may take several seconds after adding a new node for it to start up and register its drives as available.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
autoInstall	If true, a return to factory image (RTFI) will be performed on the node upon adding. The default behavior is to perform RTFI. If the `cEnableAutoInstall` cluster constant is false, it takes priority over this parameter. If an upgrade is in progress, the RTFI process will not happen regardless of the value for this parameter.	boolean	None	No
pendingNodes	Pending NodeIDs for the nodes to be added. You can list all pending nodes using the ListPendingNodes method.	integer array	None	Yes

Return value

This method has the following return value:

Name	Description	Type
autoInstall	Whether or not the added nodes are being returned to the factory image.	boolean
nodes	An array of objects mapping the previous "pendingNodeID" to the "nodeID". When you add a pending node that is running an incompatible software version, this array includes an asyncHandle value that you can use with the GetAsyncResult method to query the status of the automatic imaging process.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
    "method": "AddNodes",
    "params": {
      "autoInstall" : true,
      "pendingNodes" : [1]
    },
    "id":1
}
```

Response example

This method returns a response similar to the following example:

```
{
  id: null,
  result: {
    autoInstall: true,
    nodes: [
      {
        activeNodeKey: "giAm2ep1hA",
        assignedNodeID: 6,
        asyncHandle: 3,
        cip: "10.10.5.106",
        mip: "192.168.133.106",
        pendingNodeID: 2,
        platformInfo: {
          chassisType: "R620",
          cpuModel: "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
          nodeMemoryGB: 72,
          nodeType: "SF3010"
        },
        sip: "10.10.5.106",
        softwareVersion: "9.0.0.1077"
      }
    ]
  }
}
```

New since version

9.6

Find more information

- [AddDrives](#)
- [GetAsyncResult](#)
- [ListAllNodes](#)
- [ModifyVirtualNetwork](#)

ClearClusterFaults

You can use the `ClearClusterFaults` method to clear information about both current and previously detected faults. Both resolved and unresolved faults can be cleared.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
faultTypes	Determines the types of faults to clear. Possible values: • current: Faults that are detected currently and have not been resolved. • resolved: Faults that were previously detected and resolved. • all: Both current and resolved faults. The fault status can be determined by the "resolved" field of the fault object.	string	resolved	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

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

New since version

9.6

CreateClusterInterfacePreference

The `CreateClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to create and store arbitrary information on the storage cluster. This method is for internal use.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference.	string	None	Yes
value	The value of the cluster interface preference.	string	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
      "method": "CreateClusterInterfacePreference",
      "params": {
          "name": "prefname",
          "value": "testvalue"
      },
   "id": 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

11.0

DeleteClusterInterfacePreference

The `DeleteClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to delete an existing cluster interface preference. This method is for internal use.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference to delete.	string	None	Yes

Return values

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
    "method": "DeleteClusterInterfacePreference",
    "params": {
        "name": "prefname"
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

11.0

EnableFeature

You can use the `EnableFeature` method to enable cluster features such as VVols that are disabled by default.

Parameter

This method has the following input parameter.

> (i) For systems running Element software 11.x, enabling virtual volumes before or after setting protection domain monitoring causes the cluster protection domains feature to function only at node level.

Name	Description	Type	Default value	Required
feature	Enable a cluster feature. Possible values: • `fips`: Enable FIPS 140-2 certified encryption for HTTPS communications. • `FipsDrives`: Enable FIPS 140-2 drive support for the storage cluster. • `SnapMirror`: Enable the SnapMirror replication cluster feature. • `vvols`: Enable the Element software VVols cluster feature.	string	None	Yes

Return value

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableFeature",
    "params": {
        "feature" : "vvols"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

GetClusterCapacity

You can use the `GetClusterCapacity` to return high-level capacity measurements for an entire storage cluster. This method returns fields that you can use to calculate the efficiency rates shown in the Element web UI. You can use the efficiency calculations in scripts to return the efficiency rates for thin provisioning, deduplication, compression, and overall efficiency.

Efficiency calculations

Use the following equations to calculate thin provisioning, deduplication, and compression. These equations apply to Element 8.2 and later.

- thinProvisioningFactor = (nonZeroBlocks + zeroBlocks) / nonZeroBlocks
- deDuplicationFactor = (nonZeroBlocks + snapshotNonZeroBlocks) / uniqueBlocks
- compressionFactor = (uniqueBlocks * 4096) / (uniqueBlocksUsedSpace * 0.93)

Overall efficiency rate calculation

Use the following equation to calculate overall cluster efficiency using the results of the thin provisioning, deduplication, and compression efficiency calculations.

- efficiencyFactor = thinProvisioningFactor * deDuplicationFactor * compressionFactor

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
clusterCapacity	Capacity measurements for the storage cluster.	clusterCapacity

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "clusterCapacity": {
      "activeBlockSpace": 236015557096,
      "activeSessions": 20,
      "averageIOPS": 0,
      "clusterRecentIOSize": 0,
      "currentIOPS": 0,
      "maxIOPS": 150000,
      "maxOverProvisionableSpace": 259189767127040,
      "maxProvisionedSpace": 51837953425408,
      "maxUsedMetadataSpace": 404984011161,
      "maxUsedSpace": 12002762096640,
      "nonZeroBlocks": 310080350,
      "peakActiveSessions": 20,
      "peakIOPS": 0,
      "provisionedSpace": 1357931085824,
      "snapshotNonZeroBlocks": 0,
      "timestamp": "2016-10-17T21:24:36Z",
      "totalOps": 1027407650,
      "uniqueBlocks": 108180156,
      "uniqueBlocksUsedSpace": 244572686901,
      "usedMetadataSpace": 8745762816,
      "usedMetadataSpaceInSnapshots": 8745762816,
      "usedSpace": 244572686901,
      "zeroBlocks": 352971938
    }
  }
}
```

New since version

9.6

GetClusterFullThreshold

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

> ⓘ   When a cluster reaches the Error stage of block cluster fullness, the maximum IOPS on all
> volumes are reduced linearly to the volume minimum IOPS as the cluster approaches the
> Critical stage. This helps prevent the cluster from reaching the Critical stage of block cluster
> fullness.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
blockFullness	The current computed level of block fullness of the cluster. • stage1Happy: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage2Aware: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage3Low: Your system cannot provide redundant data protection from two non-simultaneous node failures. Corresponds to the **Warning** state in the web UI. You can configure this level in the web UI (by default, the system triggers this alert at a capacity of 3% below the Error state). • stage4Critical: The system is not capable of providing redundant data protection from a single node failure. No new volumes or clones can be created. Corresponds to the **Error** state in the Element UI. • stage5CompletelyConsumed: Completely consumed. The cluster is read-only and iSCSI connections are maintained, but all writes are suspended. Corresponds to the **Critical** state in the Element UI.	string
fullness	Reflects the highest level of fullness between "blockFullness" and "metadataFullness".	string
maxMetadataOverProvisionFactor	A value representative of the number of times metadata space can be over provisioned relative to the amount of space available. For example, if there was enough metadata space to store 100 TiB of volumes and this number was set to 5, then 500 TiB worth of volumes could be created.	integer

Name	Description	Type
metadataFullness	The current computed level of metadata fullness of the cluster. • stage1Happy: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage2Aware: No alerts or error conditions. Corresponds to the **Healthy** state in the web UI. • stage3Low: Your system cannot provide redundant data protection from two non-simultaneous node failures. Corresponds to the **Warning** state in the web UI. You can configure this level in the web UI (by default, the system triggers this alert at a capacity of 3% below the Error state). • stage4Critical: The system is not capable of providing redundant data protection from a single node failure. No new volumes or clones can be created. Corresponds to the **Error** state in the Element UI. • stage5CompletelyConsumed: Completely consumed. The cluster is read-only and iSCSI connections are maintained, but all writes are suspended. Corresponds to the **Critical** state in the Element UI.	string
sliceReserveUsedThresholdPct	Error condition. A system alert is triggered if the reserved slice utilization is greater than this value.	integer
stage2AwareThreshold	Awareness condition. The value that is set for the stage 2 cluster threshold level.	integer
stage2BlockThresholdBytes	The number of bytes being used by the cluster at which a stage 2 condition will exist.	integer

Name	Description	Type
stage2MetadataThresholdBytes	The number of metadata bytes being used by the cluster at which a stage 2 fullness condition will exist.	
stage3BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 3 fullness condition will exist.	integer
stage3BlockThresholdPercent	The percent value set for stage 3. At this percent full, a warning is posted in the Alerts log.	integer
stage3LowThreshold	Error condition. The threshold at which a system alert is created due to low capacity on a cluster.	integer
stage3MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 3 fullness condition will exist.	integer
stage3MetadataThresholdPercent	The percent value set for stage3 of metadata fullness. At this percent full, a warning will be posted in the Alerts log.	integer
stage4BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 4 fullness condition will exist.	integer
stage4CriticalThreshold	Error condition. The threshold at which a system alert is created to warn about critically low capacity on a cluster.	integer
stage4MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 4 fullness condition will exist.	integer
stage5BlockThresholdBytes	The number of storage bytes used by the cluster at which a stage 5 fullness condition will exist.	integer
stage5MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 5 fullness condition will exist.	integer

Name	Description	Type
sumTotalClusterBytes	The physical capacity of the cluster, measured in bytes.	integer
sumTotalMetadataClusterBytes	The total amount of space that can be used to store metadata.	integer
sumUsedClusterBytes	The number of storage bytes used on the cluster.	integer
sumUsedMetadataClusterBytes	The amount of space used on volume drives to store metadata.	integer

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id":1,
  "result":{
    "blockFullness":"stage1Happy",
    "fullness":"stage3Low",
    "maxMetadataOverProvisionFactor":5,
    "metadataFullness":"stage3Low",
    "sliceReserveUsedThresholdPct":5,
    "stage2AwareThreshold":3,
    "stage2BlockThresholdBytes":2640607661261,
    "stage3BlockThresholdBytes":8281905846682,
    "stage3BlockThresholdPercent":5,
    "stage3LowThreshold":2,
    "stage4BlockThresholdBytes":8641988709581,
    "stage4CriticalThreshold":1,
    "stage5BlockThresholdBytes":12002762096640,
    "sumTotalClusterBytes":12002762096640,
    "sumTotalMetadataClusterBytes":404849531289,
    "sumUsedClusterBytes":45553617581,
    "sumUsedMetadataClusterBytes":31703113728
  }
}
```

New since version

9.6

Find more information

[ModifyClusterFullThreshold](#)

GetClusterHardwareInfo

You can use the `GetClusterHardwareInfo` method to retrieve the hardware status and information for all Fibre Channel nodes, iSCSI nodes and drives in the cluster. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
type	Include only one of the following types of hardware information in the response. Possible values: • drives: Lists only drive information in the response. • nodes: Lists only node information in the response. • all: Includes both drive and node information in the response. If this parameter is omitted, a type of all is assumed.	string	all	No

Return value

This method has the following return value:

Name	Description	Type
clusterHardwareInfo	Hardware information for all nodes and drives in the cluster. Each object in this output is labeled with the nodeID of the given node.	hardwareInfo

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetClusterHardwareInfo",
    "params": {
        "type": "all"
    },
    "id": 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

GetClusterHardwareInfo

GetClusterInfo

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

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
clusterInfo	Cluster information.	clusterInfo

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "clusterInfo": {
            "attributes": {},
            "defaultProtectionScheme": "doubleHelix",
            "enabledProtectionSchemes": [
                "doubleHelix"
            ],
            "encryptionAtRestState": "disabled",
            "ensemble": [
                "10.10.10.32",
                "10.10.10.34",
                "10.10.10.35",
                "10.10.10.36",
                "10.10.10.37"
            ],
            "mvip": "10.10.11.225",
            "mvipInterface": "team1G",
            "mvipNodeID": 3,
            "mvipVlanTag": "0",
            "name": "ClusterName",
            "repCount": 2,
            "softwareEncryptionAtRestState": "enabled",
            "supportedProtectionSchemes": [
                "doubleHelix"
            ],
            "svip": "10.10.10.111",
            "svipInterface": "team10G",
            "svipNodeID": 3,
            "svipVlanTag": "0",
            "uniqueID": "psmp",
            "uuid": "2f575d0c-36fe-406d-9d10-dbc1c306ade7"
        }
    }
}
```

New since version

9.6

GetClusterInterfacePreference

The `GetClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to get information about an existing cluster

interface preference. This method is for internal use.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference.	string	None	Yes

Return value

This method has the following return value:

Name	Description	Type
preference	The name and value of the requested cluster interface preference.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
    "method": "GetClusterInterfacePreference",
    "params": {
        "name": "prefname"
        },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "preference": {
            "name": "prefname",
            "value": "testvalue"
        }
    }
}
```

New since version

11.0

GetClusterMasterNodeID

You can use the `GetClusterMasterNodeID` method to retrieve the ID of the node that performs cluster-wide administration tasks and holds the storage virtual IP address (SVIP) and management virtual IP address (MVIP).

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
nodeID	ID of the master node.	integer

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

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

New since version

9.6

GetClusterStats

You can use the `GetClusterStats` method to retrieve high-level activity measurements for the cluster. Values returned are cumulative from the creation of the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
clusterStats	Cluster activity information.	clusterStats

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "clusterStats": {
      "actualIOPS": 9376,
      "averageIOPSize": 4198,
      "clientQueueDepth": 8,
      "clusterUtilization": 0.09998933225870132,
      "latencyUSec": 52,
      "normalizedIOPS": 15000,
      "readBytes": 31949074432,
      "readBytesLastSample": 30883840,
      "readLatencyUSec": 27,
      "readLatencyUSecTotal": 182269319,
      "readOps": 1383161,
      "readOpsLastSample": 3770,
      "samplePeriodMsec": 500,
      "servicesCount": 3,
      "servicesTotal": 3,
      "timestamp": "2017-09-09T21:15:39.809332Z",
      "unalignedReads": 0,
      "unalignedWrites": 0,
      "writeBytes": 8002002944,
      "writeBytesLastSample": 7520256,
      "writeLatencyUSec": 156,
      "writeLatencyUSecTotal": 231848965,
      "writeOps": 346383,
      "writeOpsLastSample": 918
    }
  }
}
```

New since version

9.6

GetClusterVersionInfo

You can use the `GetClusterVersionInfo` method to retrieve information about the Element software version running on each node in the cluster. This method also returns information about nodes that are currently in the process of upgrading software.

Cluster version info object members

This method has the following object members:

Name	Description	Type
nodeID	ID of the node.	integer
nodeInternalRevision	Internal software version of the node.	string
nodeVersion	Software version of the node.	string

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
clusterAPIVersion	The current API version on the cluster.	string
clusterVersion	Version of Element software currently running on the cluster.	string
clusterVersionInfo	List of nodes in the cluster with version information for each node.	JSON object array
pendingClusterVersion	If present, this is the version that the cluster software is currently being upgraded or reverted to.	string

Name	Description	Type
softwareVersionInfo	The state of an upgrade. Object members: • currentVersion: The current software version on a node. • nodeID: ID of the node being upgraded from currentVersion to pendingVersion. This field is 0 (zero) if there is no upgrade in progress. • packageName: Name of the software package being installed. • pendingVersion: The version of the software being installed. • startTime: The date and time the installation was started, in UTC+0 format.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
 "id": 1,
 "result": {
   "clusterAPIVersion": "6.0",
   "clusterVersion": "6.1382",
   "clusterVersionInfo": [
   {
       "nodeID": 1,
       "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
       "nodeVersion": "6.1382"
   },
   {
       "nodeID": 2,
       "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
       "nodeVersion": "6.1382"
   },
   {
       "nodeID": 3,
       "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
       "nodeVersion": "6.1382"
   },
   {
       "nodeID": 4,
       "nodeInternalRevision": "BuildType=Release Element=carbon
Release=carbon ReleaseShort=carbon Version=6.1382 sfdev=6.28
Repository=dev Revision=061511b1e7fb BuildDate=2014-05-28T18:26:45MDT",
       "nodeVersion": "6.1382"
   }
 ],
   "softwareVersionInfo": {
       "currentVersion": "6.1382",
       "nodeID": 0,
       "packageName": "",
       "pendingVersion": "6.1382",
       "startTime": ""
   }
 }
}
```

New since version

9.6

GetFeatureStatus

You can use the `GetFeatureStatus` method to retrieve the status of a cluster feature.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
feature	The status of a cluster feature. If no value is provided, the system returns a status of all features. Possible values: • Vvols: Retrieve status for the VVols cluster feature. • SnapMirror: Retrieve status for the SnapMirror replication cluster feature. • Fips: Retrieve status for the FIPS 140-2 encryption for HTTPS communication feature. • fipsDrives: Retrieve status for the FIPS 140-2 drive encryption feature.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
features	An array of feature objects indicating the feature name and its status. Object members: • feature: (string) The name of the feature. • enabled: (boolean) Whether the feature is enabled or not.	JSON object array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "features": [
            {
                "enabled": true,
                "feature": "Vvols"
            },
            {
                "enabled": true,
                "feature": "SnapMirror"
            },
            {
                "enabled": true,
                "feature": "Fips"
            },
            {
                "enabled": true,
                "feature": "FipsDrives"
            }
        ]
    }
}
```

New since version

9.6

GetLoginSessionInfo

You can use the `GetLoginSessionInfo` method to return the period of time a login authentication session is valid for both login shells and the TUI.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
loginSessionInfo	An object containing the authentication expiration period. Possible objects returned: • timeout: The time, in minutes, when this session will timeout and expire. Formatted in H:mm:ss. For example: 1:30:00, 20:00, 5:00. All leading zeros and colons are removed regardless of the format the timeout was entered.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
"id": 1,
  "result" : {
    "loginSessionInfo" : {
      "timeout" : "30:00"
    }
  }
}
```

New since version

9.6

GetNodeHardwareInfo

You can use the `GetNodeHardwareInfo` method to return all the hardware information

and status for the node specified. This generally includes manufacturers, vendors, versions, and other associated hardware identification information.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
nodeID	The ID of the node for which hardware information is being requested. Information about a Fibre Channel node is returned if a Fibre Channel node is specified.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
nodeHardwareInfo	Hardware information for the specified nodeID. Each object in this output is labeled with the nodeID of the given node.	hardwareInfo

Request example

Requests for this method are similar to the following example:

```
{
   "method": "GetNodeHardwareInfo",
   "params": {
      "nodeID": 1
   },
"id" : 1
}
```

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

GetNodeHardwareInfo (output for Fibre Channel nodes)

GetNodeHardwareInfo (output for iSCSI)

GetNodeStats

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

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
nodeID	Specifies the ID of the node for which statistics will be returned.	integer	None	Yes

Return value

This method has the following return value:

Name	Description	Type
nodeStats	Node activity information.	nodeStats

Request example

Requests for this method are similar to the following example:

```
{
    "method": "GetNodeStats",
    "params": {
        "nodeID": 5
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
    "id" : 1,
    "result" : {
      "nodeStats" : {
        "cBytesIn" : 9725856460404,
        "cBytesOut" : 16730049266858,
        "cpu" : 98,
        "mBytesIn" : 50808519,
        "mBytesOut" : 52040158,
        "networkUtilizationCluster" : 84,
        "networkUtilizationStorage" : 0,
        "sBytesIn" : 9725856460404,
        "sBytesOut" : 16730049266858,
        "timestamp" : "2012-05-16T19:14:37.167521Z",
        "usedMemory" : 41195708000
      }
    }
}
```

New since version

9.6

ListActiveNodes

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

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
nodes	List of active nodes in the cluster.	node array

Request example

Requests for this method are similar to the following example:

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

Response example

Due to the length of this response example, it is documented in a supplementary topic.

New since version

9.6

Find more information

ListActiveNodes

ListAllNodes

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

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
nodes	List of objects describing active nodes in the cluster.	node
pendingActiveNodes	List of objects describing pending active nodes for the cluster.	pendingActiveNode array
pendingNodes	List of objects describing pending nodes for the cluster.	pendingNode array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "nodes": [
            {
                "associatedFServiceID": 0,
                "associatedMasterServiceID": 1,
                "attributes": {},
                "chassisName": "CT5TV12",
                "cip": "10.1.1.1",
                "cipi": "Bond10G",
                "fibreChannelTargetPortGroup": null,
                "mip": "10.1.1.1",
                "mipi": "Bond1G",
                "name": "NLABP0704",
                "nodeID": 1,
                "nodeSlot": "",
                "platformInfo": {
                    "chassisType": "R620",
                    "cpuModel": "Intel",
                    "nodeMemoryGB": 72,
                    "nodeType": "SF3010",
                    "platformConfigVersion": "0.0.0.0"
                },
                "sip": "10.1.1.1",
                "sipi": "Bond10G",
                "softwareVersion": "11.0",
                "uuid": "4C4C4544-0054",
                "virtualNetworks": []
            }
        ],
        "pendingActiveNodes": [],
        "pendingNodes": []
    }
}
```

New since version

9.6

ListClusterFaults

You can use the `ListClusterFaults` method to list information about any faults
detected on the cluster. With this method, you can list both current faults as well as faults
that have been resolved. The system caches faults every 30 seconds.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
bestPractices	Include faults triggered by sub-optimal system configuration. Possible values: • true • false	boolean	None	No
faultTypes	Determines the types of faults returned. Possible values: • current: List active, unresolved faults. • resolved: List faults that were previously detected and resolved. • all: List both current and resolved faults. You can see the fault status in the "resolved" member of the fault object.	string	all	No

Return value

This method has the following return value:

Name	Description	Type
faults	An object describing the requested cluster faults.	fault

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ListClusterFaults",
    "params": {
      "faultTypes": "current",
      "bestPractices": true
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "faults": [
      {
        "blocksUpgrade": false,
        "clusterFaultID": 3,
        "code": "driveAvailable",
        "data": null,
        "date": "2024-04-03T22:22:56.660275Z",
        "details": "Node ID 1 has 6 available drive(s).",
        "driveID": 0,
        "driveIDs": [],
        "externalSource": "",
        "networkInterface": "",
        "nodeHardwareFaultID": 0,
        "nodeID": 1,
        "resolved": true,
        "resolvedDate": "2024-04-03T22:24:54.598693Z",
        "serviceID": 0,
        "severity": "warning",
        "type": "drive"
      },
      {
        "clusterFaultID": 9,
        "code": "disconnectedClusterPair",
        "data": null,
        "date": "2016-04-26T20:40:08.736597Z",
        "details": "One of the clusters in a pair may have become
misconfigured or disconnected.  Remove the local pairing and retry pairing
the clusters. Disconnected Cluster Pairs: []. Misconfigured Cluster Pairs:
[3]",
        "driveID": 0,
        "driveIDs": [],
        "nodeHardwareFaultID": 0,
        "nodeID": 0,
        "resolved": false,
        "resolvedDate": "",
        "serviceID": 0,
        "severity": "warning",
        "type": "cluster"
      }
    ]
  }
}
```

New since version

9.6

ListClusterInterfacePreferences

The `ListClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to list the existing cluster interface preferences stored on the system. This method is for internal use.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
preferences	A list of cluster interface objects currently stored on the storage cluster, each containing the name and value of the preference.	JSON object array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
        "preferences": [
            {
                "name": "prefname",
                "value": "testvalue"
            }
        ]
    }
}
```

New since version

11.0

ListEvents

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

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
driveID	Specifies that only events with this drive ID will be returned.	integer	0	No
endEventID	Identifies the end of a range of event IDs to return.	integer	(unlimited)	No
endPublishTime	Specifies that only events published earlier than this time will be returned.	string	0	No
endReportTime	Specifies that only events reported earlier than this time will be returned.	string	0	No

Name	Description	Type	Default value	Required
eventType	Specifies the type of events to return. See event for possible event types.	string	0	No
maxEvents	Specifies the maximum number of events to return.	integer	(unlimited)	No
nodeID	Specifies that only events with this node ID will be returned.	integer		
serviceID	Specifies that only events with this service ID will be returned.			
startEventID	Identifies the beginning of a range of events to return.	integer	0	No
startPublishTime	Specifies that only events published after this time will be returned.	string	0	No
startReportTime	Specifies that only events reported after this time will be returned.	string	0	No

Return value

This method has the following return value:

Name	Description	Type
events	List of events.	event array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id":1,
  "result":{
    "events":[
        {
            "details":
              {
                  "paramGCGeneration":1431550800,
                  "paramServiceID":2
              },
            "driveID":0,
            "eventID":2131,
            "eventInfoType":"gcEvent",
            "message":"GC Cluster Coordination Complete",
            "nodeID":0,
            "serviceID":2,
            "severity":0,
            "timeOfPublish":"2015-05-13T21:00:02.361354Z",
            "timeOfReport":"2015-05-13T21:00:02.361269Z"
        },{
            "details":
                    {

"eligibleBS":[5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,24,25,26,27,28,29,30
,31,40,41,42,43,44,45,46,47,52,53,54,55,56,57,58,59,60],
                    "generation":1431550800,
                    "participatingSS":[23,35,39,51]
                },
            "driveID":0,
            "eventID":2130,
            "eventInfoType":"gcEvent",
            "message":"GCStarted",
            "nodeID":0,
            "serviceID":2,
```

```
            "severity":0,
            "timeOfPublish":"2015-05-13T21:00:02.354128Z",
            "timeOfReport":"2015-05-13T21:00:02.353894Z"
        },{
            "details":"",
            "driveID":0,
            "eventID":2129,
            "eventInfoType":"tSEvent",
            "message":"return code:2 t:41286 tt:41286 qcc:1 qd:1 qc:1 vrc:1
 tt:2 ct:Write etl:524288",
            "nodeID":0,
            "serviceID":0,
            "severity":0,
            "timeOfPublish":"2015-05-13T20:45:21.586483Z",
            "timeOfReport":"2015-05-13T20:45:21.586311Z"
        }
    ]
  }
}
```

New since version

9.6

ListNodeStats

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

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
nodeStats	Storage node activity information.	nodeStats

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
    "id": 1,
    "result": {
      "nodeStats": {
        "nodes": [
          {
          "cBytesIn": 46480366124,
          "cBytesOut": 46601523187,
          "cpu": 0,
          "mBytesIn": 59934129,
          "mBytesOut": 41620976,
          "networkUtilizationCluster": 0,
          "networkUtilizationStorage": 0,
          "nodeID": 1,
          "sBytesIn": 46480366124,
          "sBytesOut": 46601523187,
          "timestamp": 1895558254814,
          "usedMemory": 31608135680
          }
        ]
      }
    }
}
```

New since version

9.6

ListISCSISessions

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

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
sessions	Information about each iSCSI session.	session

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
  {
   "id": 1,
   "result": {
     "sessions": [
       {
         "accountID": 1,
         "accountName": "account1",
         "authentication": {
             "authMethod": "CHAP",
             "chapAlgorithm": "SHA3_256",
             "chapUsername": "iqn.1994-05.com."redhat:1cf11f3eed3",
             "direction": "two-way"
         },
         "createTime": "2022-10-03T22:02:49.121723Z",
         "driveID": 23,
         "driveIDs": [23],
         "initiator": null,
         "initiatorIP": "10.1.1.1:37138",
         "initiatorName": "iqn.2010-01.net.solidfire.eng:c",
         "initiatorPortName": "iqn.2010-
01.net.solidfire.eng:c,i,0x23d860000",
         "initiatorSessionID": 9622126592,
         "msSinceLastIscsiPDU": 243,
         "msSinceLastScsiCommand": 141535021,
         "nodeID": 3,
         "serviceID": 6,
         "sessionID": 25769804943,
         "targetIP": "10.1.1.2:3260",
         "targetName": "iqn.2010-01.com.solidfire:a7sd.3",
         "targetPortName": "iqn.2010-01.com.solidfire:a7sd.3,t,0x1",
         "virtualNetworkID": 0,
         "volumeID": 3,
         "volumeInstance": 140327214758656
       }
       ...
     ]
   }
}
```

New since version

9.6

ListServices

You can use the `ListServices` method to list services information for nodes, drives, current software, and other services that are running on the cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
services	Services that are running on drives and nodes.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
"id": 1,
"result": {
    "services": [
        {
            "drive": {
                "assignedService": 22,
                "asyncResultIDs": [],
                "attributes": {},
                "capacity": 300069052416,
                "customerSliceFileCapacity": 0,
                "driveID": 5,
                "driveStatus": "assigned",
                "driveType": "block",
                "failCount": 0,
```

```
            "nodeID": 4,
            "reservedSliceFileCapacity": 0,
            "serial": "scsi-SATA_INTEL_SSDSC2",
            "slot": 3
        },
        "drives": [
            {
                "assignedService": 22,
                "asyncResultIDs": [],
                "attributes": {},
                "capacity": 300069052416,
                "customerSliceFileCapacity": 0,
                "driveID": 5,
                "driveStatus": "assigned",
                "driveType": "Block",
                "failCount": 0,
                "nodeID": 4,
                "reservedSliceFileCapacity": 0,
                "serial": "scsi-SATA_INTEL_SSDSC2",
                "slot": 3
            }
        ],
        "node": {
            "associatedFServiceID": 0,
            "associatedMasterServiceID": 1,
            "attributes": {},
            "cip": "10.117.63.18",
            "cipi": "Bond10G",
            "fibreChannelTargetPortGroup": null,
            "mip": "10.117.61.18",
            "mipi": "Bond1G",
            "name": "node4",
            "nodeID": 4,
            "nodeSlot": "",
            "platformInfo": {
                "chassisType": "R620",
                "cpuModel": "Intel(R) Xeon(R) CPU",
                "nodeMemoryGB": 72,
                "nodeType": "SF3010",
                "platformConfigVersion": "10.0"
            },
            "sip": "10.117.63.18",
            "sipi": "Bond10G",
            "softwareVersion": "10.0",
            "uuid": "4C4C4544-0053",
            "virtualNetworks": []
```

```
            },
            "service": {
                "associatedBV": 0,
                "associatedTS": 0,
                "associatedVS": 0,
                "asyncResultIDs": [
                    1
                ],
                "driveID": 5,
                "driveIDs": [
                    5
                ],
                "firstTimeStartup": true,
                "ipcPort": 4008,
                "iscsiPort": 0,
                "nodeID": 4,
                "serviceID": 22,
                "serviceType": "block",
                "startedDriveIDs": [],
                "status": "healthy"
            }
        }
    ]
}
```

New since version

9.6

ListPendingNodes

You can use the `ListPendingNodes` method to list the pending storage nodes in the system. Pending nodes are storage nodes that are running and configured to join the storage cluster but have not yet been added using the AddNodes API method.

IPv4 and IPv6 management addresses

Note that `ListPendingNodes` does not list pending nodes that have different address types for the management IP address (MIP) and management virtual IP address (MVIP). For example, if a pending node has an IPv6 MVIP and an IPv4 MIP, `ListPendingNodes` will not include the node as part of the result.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
pendingNodes	List of pending nodes in the cluster.	pendingNode array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 3,
  "result": {
    "pendingNodes": [
      {
        "assignedNodeID": 0,
        "cip": "10.26.65.101",
        "cipi": "Bond10G",
        "compatible": true,
        "mip": "172.26.65.101",
        "mipi": "Bond1G",
        "name": "VWC-EN101",
        "pendingNodeID": 1,
        "platformInfo": {
          "chassisType": "R620",
          "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
          "nodeMemoryGB": 72,
          "nodeType": "SF3010"
        },
        "sip": "10.26.65.101",
        "sipi": "Bond10G",
        "softwareVersion": "9.0.0.1554",
        "uuid": "4C4C4544-0048-4410-8056-C7C04F395931"
      }
    ]
  }
}
```

New since version

9.6

Find more information

AddNodes

ListPendingActiveNodes

You can use the `ListPendingActiveNodes` method to list nodes in the cluster that are in the PendingActive state, between pending and active states. Nodes in this state are being returned to the factory image.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
pendingActiveNodes	List of objects detailing information about all PendingActive nodes in the system.	pendingActiveNode array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  id: null,
  result: {
    pendingActiveNodes: [
    {
      activeNodeKey: "5rPHP3lTAO",
      assignedNodeID: 5,
      asyncHandle: 2,
      cip: "10.10.5.106",
      mip: "192.168.133.106",
      pendingNodeID: 1,
      platformInfo: {
        chassisType: "R620",
        cpuModel: "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
        nodeMemoryGB: 72,
        nodeType: "SF3010"
      },
      sip: "10.10.5.106",
      softwareVersion: "9.0.0.1077"
    }
    ]
  }
}
```

New since version

9.6

ModifyClusterFullThreshold

You can use the `ModifyClusterFullThreshold` method to change the level at which the system generates an event when the storage cluster approaches a certain capacity utilization. You can use the threshold setting to indicate the acceptable amount of utilized block storage before the system generates a warning.

For example, if you want to be alerted when the system reaches 3% below the "Error" level block storage utilization, enter a value of "3" for the stage3BlockThresholdPercent parameter. If this level is reached, the system sends an alert to the Event Log in the Cluster Management Console.

Parameters

This method has the following input parameters:

ⓘ You must select at least one parameter.

Name	Description	Type	Default value	Required
maxMetadataOverProvisionFactor	A value representative of the number of times metadata space can be over provisioned relative to the amount of space available. For example, if there was enough metadata space to store 100 TiB of volumes and this number was set to 5, then 500 TiB worth of volumes could be created.	integer	5	No
stage2AwareThreshold	The number of nodes of capacity remaining in the cluster before the system triggers a capacity notification.	integer	None	No
stage3BlockThresholdPercent	The percentage of block storage utilization below the "Error" threshold that causes the system to trigger a cluster "Warning" alert.	integer	None	No
stage3MetadataThresholdPercent	The percentage of metadata storage utilization below the "Error" threshold that causes the system to trigger a cluster "Warning" alert.	integer	None	No

Return values

This method has the following return values:

Name	Description	Type

blockFullness	The current computed level of block fullness of the cluster.	string
	• 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.	
fullness	Reflects the highest level of fullness between "blockFullness" and "metadataFullness".	string
maxMetadataOverProvisionFactor	A value representative of the number of times metadata space can be over provisioned relative to the amount of space available. For example, if there was enough metadata space to store 100 TiB of volumes and this number was set to 5, then 500 TiB worth of volumes could be created.	integer

metadataFullness	The current computed level of metadata fullness of the cluster.	string
	• 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.	
sliceReserveUsedThresholdPct	Error condition. A system alert is triggered if the reserved slice utilization is greater than the sliceReserveUsedThresholdPct value returned.	integer
stage2AwareThreshold	Awareness condition. The value that is set for "Stage 2" cluster threshold level.	integer
stage2BlockThresholdBytes	The number of bytes being used by the cluster at which a stage 2 fullness condition will exist.	integer

stage2MetadataThresholdBytes	The number of metadata bytes being used by the cluster at which a stage 2 fullness condition will exist.	
stage3BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 3 fullness condition will exist.	integer
stage3BlockThresholdPercent	The percent value set for stage 3. At this percent full, a warning is posted in the Alerts log.	integer
stage3LowThreshold	Error condition. The threshold at which a system alert is created due to low capacity on a cluster.	integer
stage3MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 3 fullness condition will exist.	
stage4BlockThresholdBytes	The number of storage bytes being used by the cluster at which a stage 4 fullness condition will exist.	integer
stage4CriticalThreshold	Error condition. The threshold at which a system alert is created to warn about critically low capacity on a cluster.	integer
stage4MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 4 fullness condition will exist.	
stage5BlockThresholdBytes	The number of storage bytes used by the cluster at which a stage 5 fullness condition will exist.	integer
stage5MetadataThresholdBytes	The number of metadata bytes used by the cluster at which a stage 5 fullness condition will exist.	
sumTotalClusterBytes	The physical capacity of the cluster, measured in bytes.	integer
sumTotalMetadataClusterBytes	The total amount of space that can be used to store metadata.	integer

sumUsedClusterBytes	The number of storage bytes used on the cluster.	integer
sumUsedMetadataClusterBytes	The amount of space used on volume drives to store metadata.	integer

Request example

Requests for this method are similar to the following example:

```
{
    "method" : "ModifyClusterFullThreshold",
    "params" : {
             "stage3BlockThresholdPercent" : 3
             },
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "blockFullness": "stage1Happy",
    "fullness": "stage3Low",
    "maxMetadataOverProvisionFactor": 5,
    "metadataFullness": "stage3Low",
    "sliceReserveUsedThresholdPct": 5,
    "stage2AwareThreshold": 3,
    "stage2BlockThresholdBytes": 2640607661261,
    "stage3BlockThresholdBytes": 8281905846682,
    "stage3BlockThresholdPercent": 3,
    "stage3LowThreshold": 2,
    "stage4BlockThresholdBytes": 8641988709581,
    "stage4CriticalThreshold": 1,
    "stage5BlockThresholdBytes": 12002762096640,
    "sumTotalClusterBytes": 12002762096640,
    "sumTotalMetadataClusterBytes": 404849531289,
    "sumUsedClusterBytes": 45553617581,
    "sumUsedMetadataClusterBytes": 31703113728
  }
}
```

New since version

9.6

ModifyClusterInterfacePreference

The `ModifyClusterInterfacePreference` method enables systems integrated with storage clusters running Element software to change an existing cluster interface preference. This method is for internal use.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
name	The name of the cluster interface preference to modify.	string	None	Yes
value	The new value of the cluster interface preference.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
    "method": "ModifyClusterInterfacePreference",
    "params": {
    "name": "testname",
    "value": "newvalue"
},
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

11.0

RemoveNodes

You can use `RemoveNodes` to remove one or more nodes that should no longer participate in the cluster.

Before removing a node, you must remove all drives the node contains using the `RemoveDrives` method. You cannot remove a node until the `RemoveDrives` process has completed and all data has been migrated away from the node. After you remove a node, it registers itself as a pending node. You can add the node again or shut it down (shutting the node down removes it from the pending node list).

Cluster master node removal

If you use `RemoveNodes` to remove the cluster master node, the method might time out before returning a response. If the method call fails to remove the node, make the method call again. Note that if you are removing the cluster master node along with other nodes, you should use a separate call to remove the cluster master node by itself.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
ignoreEnsembleTole ranceChange	Ignore changes to the ensemble's node failure tolerance when removing nodes. If the storage cluster uses data protection schemes that tolerate multiple node failures and removing the nodes would decrease the node failure tolerance of the ensemble, the node removal normally fails with an error. You can set this parameter to true to disable the ensemble tolerance check so that the node removal succeeds.	boolean	false	No
nodes	List of NodeIDs for the nodes to be removed.	integer array	None	Yes

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
    "method": "RemoveNodes",
    "params": {
      "nodes"  : [3,4,5]
    },
    "id" : 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

SetLoginSessionInfo

You can use the `SetLoginSessionInfo` method to set the period of time that a login authentication for a session is valid. After the login period elapses without activity on the system, the authentication expires. New login credentials are required for continued access to the cluster after the login period has elapsed.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
timeout	Cluster authentication expiration period. Formatted in HH:mm:ss. For example: 01:30:00, 00:90:00, and 00:00:5400 can all be used to equal a 90 minute timeout period. The minimum timeout value is 1 minute. When a value is not provided, or is set to zero, the login session has no timeout value.	string	30 minutes	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetLoginSessionInfo",
  "params": {
    "timeout" : "01:30:00"
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

Shutdown

You can use the `Shutdown` method to restart or shutdown the nodes in a cluster. You can shut down a single node, multiple nodes, or all of the nodes in the cluster using this method.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
nodes	List of NodeIDs for the nodes to be restarted or shut down.	integer array	None	Yes

Name	Description	Type	Default value	Required
option	Action to take for the cluster. Possible values: • restart: Restarts the cluster. • halt: Performs a full power-off.	string	restart	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

```
{
    "method": "Shutdown",
    "params": {
      "nodes": [
         2,
         3,
         4
      ],
      "option": "halt"
    },
    "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
     "failed": [],
     "successful": [
          6
      [
    }
}
```

New since version

9.6