



Node API methods

Element Software

NetApp

November 12, 2025

This PDF was generated from https://docs.netapp.com/us-en/element-software-128/api/reference_element_api_checkpingonvlan.html on November 12, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Node API methods	1
CheckPingOnVlan	1
Parameters	1
Return values	2
Request example	3
Response example	3
New since version	4
CheckProposedNodeAdditions	4
Parameter	5
Return values	5
Request example	6
Response example	7
New since version	7
CreateClusterSupportBundle	7
Parameters	7
Return values	9
Request example	9
Response example	9
New since version	9
CreateSupportBundle	10
Parameters	10
Return values	10
Request example	11
Response example	11
New since version	12
DeleteAllSupportBundles	12
Parameters	12
Return values	12
Request example	12
Response example	13
New since version	13
DisableMaintenanceMode	13
Parameters	13
Return values	13
Request example	15
Response example	15
New since version	16
Find more information	16
DisableSsh	16
Parameter	16
Return value	16
Request example	16
Response example	17

EnableMaintenanceMode	17
Parameters	17
Return values	18
Request example	19
Response example	19
New since version	20
Find more information	20
EnableSsh	20
Parameter	20
Return value	20
Request example	20
Response example	21
GetClusterConfig	21
Parameters	21
Return value	21
Request example	21
Response example	22
New since version	22
GetClusterState	22
Parameters	22
Return values	22
Request example	23
Response example	23
New since version	23
GetConfig	23
Parameters	24
Return values	24
Request example	24
Response example	24
New since version	24
Find more information	24
GetDriveConfig	25
Parameters	25
Return value	25
Request example	25
Response example	25
GetHardwareConfig	27
Parameters	27
Return value	27
Request example	27
Response example	27
New since version	29
GetHardwareInfo	29
Parameter	29
Return value	29

Request example	30
Response example	30
New since version	31
GetIpmiConfig	31
Parameter	31
Return values	32
Request example	32
Response example	32
New since version	36
GetIpmiInfo	36
Parameters	36
Return value	36
Request example	36
Response example	36
New since version	39
GetNetworkConfig	39
Parameters	40
Return value	40
Request example	40
Response example	40
New since version	40
Find more information	40
GetNetworkInterface	40
Parameters	40
Return value	41
Request example	41
Response example	42
New since version	43
GetNodeActiveTlsCiphers	43
Parameter	44
Return values	44
Request example	44
Response example	44
GetNodeFipsDrivesReport	45
Parameter	45
Return values	45
Request example	46
Response example	46
New since version	46
GetNodeSSLCertificate	46
Parameters	47
Return values	47
Request example	47
Response example	47
GetNodeSupportedTlsCiphers	48

Parameter	49
Return values	49
Request example	49
Response example	49
GetPatchInfo	50
Parameters	50
Return values	51
Request example	51
Response example	51
New since version	52
GetPendingOperation	52
Parameters	52
Return values	52
Request example	53
Response example	53
New since version	53
GetSshInfo	53
Parameters	54
Return value	54
Request example	54
Response example	54
ListDriveHardware	54
Parameters	55
Return value	55
Request example	55
Response example	55
New since version	56
Find more information	57
ListNetworkInterfaces	57
Parameter	57
Return value	57
Request example	57
Response example	58
New since version	59
ListNetworkInterfaceStats	59
Parameter	60
Return value	60
Request example	60
Response example	60
New since version	61
ListTests	61
Parameters	61
Return value	61
Request example	62
Response example	62

New since version	62
ListUtilities	62
Parameters	62
Return value	63
Request example	63
Response example	63
New since version	63
RemoveNodeSSLCertificate	63
Parameters	64
Return values	64
Request example	64
Response example	64
ResetDrives	64
Parameters	64
Return value	65
Request example	65
Response example	65
New since version	66
ResetNode	66
Parameters	66
Return values	67
Request example	67
Response example	67
New since version	69
ResetNodeSupplementalTlsCiphers	69
Parameter	69
Return values	69
Request example	69
Response example	69
RestartNetworking	69
Parameter	70
Return values	70
Request example	70
Response example	70
New since version	70
RestartServices	71
Parameters	71
Return values	71
Request example	71
Response example	72
New since version	72
SetClusterConfig	72
Parameter	72
Return value	73
Request example	73

Response example	73
New since version	74
SetConfig	74
Parameter	74
Return value	75
Request example	75
Response example	76
New since version	76
Find more information	76
SetNetworkConfig	76
Parameter	76
Return value	77
Request example	77
Response example	78
New since version	78
Find more information	78
SetNodeSSLCertificate	78
Parameters	78
Return values	79
Request example	79
Response example	80
SetNodeSupplementalTlsCiphers	81
Parameter	81
Return values	81
Request example	81
Response example	82
Shutdown	82
Parameters	83
Return value	83
Request example	83
Response example	83
New since version	84
TestConnectEnsemble	84
Parameters	84
Return value	84
Request example	85
Response example	85
New since version	85
TestConnectMvip	85
Parameter	85
Return value	86
Request example	89
Response example	89
New since version	90
TestConnectSvip	90

Parameter	90
Return value	91
Request example	94
Response example	94
New since version	95
TestDrives	95
Parameters	95
Return value	96
Request example	96
Response example	96
New since version	96
TestHardwareConfig	96
Parameters	96
Return value	97
Request example	97
Response example	97
New since version	98
Find more information	98
TestLocateCluster	98
Parameters	98
Return value	98
Request example	98
Response example	98
New since version	99
TestLocalConnectivity	99
Parameters	99
Return value	99
Request example	100
Response example	100
New since version	102
TestNetworkConfig	102
Parameters	102
Return value	102
Request example	102
Response example 1	103
Response example 2	103
Response example 3	104
New since version	104
Find more information	104
TestPing	104
Parameters	104
Return value	106
Request example	106
Response example	107
New since version	108

TestRemoteConnectivity	108
Parameters	109
Return value	109
Request example	109
Response example	109
New since version	111

Node API methods

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

Name	Description	Type	Default value	Required
packetSize	Specifies the number of bytes to send in the ICMP packet that is sent to each IP. The number of bytes must be less than the maximum MTU specified in the network configuration.	integer	None	No
pingTimeoutMsec	Specifies the number of milliseconds to wait for each individual ping response.	integer	500 ms	No
prohibitFragmentation	Enables the DF (Do not Fragment) flag for the ICMP packets.	boolean	false	No
sourceAddressV4	The source IPv4 address to use in the ICMP ping packets.	string	None	Yes
sourceAddressV6	The source IPv6 address to use in the ICMP ping packets.	string	None	Yes
totalTimeoutSec	Specifies the time in seconds the ping should wait for a system response before issuing the next ping attempt or ending the process.	integer	5	No
virtualNetworkTag	The VLAN ID to use when sending the ping packets.	integer	None	Yes

Return values

This method has the following return values:

Name	Description	Type
result	List of each IP the node was able to communicate with and ping response statistics.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CheckPingOnVlan",
  "params": {
    "interface": "Bond10G",
    "virtualNetworkTag": 4001,
    "sourceAddressV4": "192.168.41.4",
    "hosts": "192.168.41.2"
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```

{
  "id": 1,
  "result": {
    "192.168.41.2": {
      "individualResponseCodes": [
        "Success",
        "Success",
        "Success",
        "Success",
        "Success"
      ],
      "individualResponseTimes": [
        "00:00:00.000373",
        "00:00:00.000098",
        "00:00:00.000097",
        "00:00:00.000074",
        "00:00:00.000075"
      ],
      "individualStatus": [
        true,
        true,
        true,
        true,
        true
      ],
      "interface": "Bond10G",
      "responseTime": "00:00:00.000143",
      "sourceAddressV4": "192.168.41.4",
      "successful": true,
      "virtualNetworkTag": 4001
    }
  }
}

```

New since version

11.1

CheckProposedNodeAdditions

You can use the `CheckProposedNodeAdditions` method to test a set of storage nodes to see if you can add them to a storage cluster without errors or best practice violations.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
nodes	A list of storage IP addresses of storage nodes that are ready to be added to a storage cluster.	string array	None	Yes

Return values

This method has the following return values:

Name	Description	Type
proposedClusterValid	Indicates whether or not the proposed storage nodes would make up a valid storage cluster. Possible values: • true • false	boolean

proposedClusterErrors	Errors that would occur if a storage cluster was created using the proposed storage nodes. Possible error codes: • nodesNoCapacity: Nodes did not have any useable capacity. • nodesTooLarge: Nodes constitute too large a portion of cluster capacity for the active protection scheme. • nodesConnectFailed: Could not connect to nodes to query hardware configuration. • nodesQueryFailed: Could not query nodes for hardware configuration. • nodesClusterMember: IP addresses for nodes are already in use in the cluster. • nonFipsNodeCapable: Unable to add a non-FIPS capable node to the storage cluster while the FIPS 140-2 drive encryption feature is enabled. • nonFipsDrivesCapable: Unable to add a node with non-FIPS-capable drives to the cluster while the FIPS 140-2 drive encryption feature is enabled.	string array
-----------------------	--	--------------

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CheckProposedNodeAdditions",
  "params": {
    "nodes": [
      "192.168.1.11",
      "192.168.1.12",
      "192.168.1.13",
      "192.168.1.14"
    ]
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "proposedClusterValid": true,
    "proposedClusterErrors": [ ]
  }
}
```

New since version

11.0

CreateClusterSupportBundle

You can use the `CreateClusterSupportBundle` on the management node to gather support bundles from all nodes in a cluster. The individual node support bundles are compressed as tar.gz files. The cluster support bundle is a tar file containing the node support bundles. You can only run this method on a management node; it does not work when run on a storage node.

Parameters



You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```


This method has the following input parameters:

Name	Description	Type	Default value	Required
allowIncomplete	Allows the script to continue to run if bundles cannot be gathered from one or more of the nodes.	boolean	None	No
bundleName	Unique name for each support bundle created. If no name is provided, then "supportbundle" and the node name are used as the file name	string	None	No
mvip	The MVIP of the cluster. Bundles are gathered from all nodes in the cluster. This parameter is required if the nodes parameter is not specified.	string	None	Yes
nodes	The IP addresses of the nodes from which to gather bundles. Use either nodes or mvip, but not both, to specify the nodes from which to gather bundles. This parameter is required if mvip is not specified.	string array	None	Yes
password	The cluster admin password. Note: This password is visible as text when entered.	string	None	Yes
username	The cluster admin user name.	string	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CreateClusterSupportBundle",
  "params": {
    "bundlename": "clusterbundle",
    "mvip": "132.119.120.100"
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id":1,
  "result":{
    "details":{
      "bundleName":"clusterbundle",
      "extraArgs":"",
      "files":[
        "/tmp/supportbundles/clusterbundle.cl-4SD5.tar"
      ],
      "output":"timeout -s KILL 1790s
/usr/local/bin/sfclustersupportbundle --quiet --name=\"clusterbundle\"
--target-directory=\"/tmp/solidfire-dtemp.MM7f0m\" --user=\"admin\"
--pass=\"admin\" --mvip=132.119.120.100"
    },
    "duration":"00:00:24.938127",
    "result":"Passed"
  }
}
```

New since version

9.6

CreateSupportBundle

You can use `CreateSupportBundle` to create a support bundle file under the node's directory. After creation, the bundle is stored on the node as a tar file (gz compression option is available via the `extraArgs` parameter.)

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
bundleName	Unique name for the support bundle. If no name is provided, then "supportbundle" and the node name are used as the file name.	string	None	No
extraArgs	Use '--compress gz' to create the support bundle as a tar.gz file.	string	None	No
timeoutSec	The number of seconds the support bundle script runs.	integer	1500	No

Return values

This method has the following return values:

Name	Description	Type
------	-------------	------

details	The details of the support bundle. Possible values: • bundleName: The name specified in the <code>CreateSupportBundleAPI</code> method. If no name was specified, "supportbundle" is used. • extraArgs: The arguments passed with this method. • files: A list of the support bundle files that the system created. • output: The command line output from the script that created the support bundle. • timeoutSec: The number of seconds the support bundle script runs before stopping. • url: URL to the support bundle created.	JSON object
duration	The time used to create the support bundle in the format: HH:MM:SS.ssssss.	string
result	The success or failure of the support bundle operation.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "CreateSupportBundle",
  "params": {
    "extraArgs": "--compress gz"
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "details": {
      "bundleName": "supportbundle",
      "extraArgs": "--compress gz",
      "files": [
        "supportbundle.nodehostname.tar.gz"
      ],
      "output": "timeout -s KILL 1500s /sf/scripts/sfsupportbundle --quiet
--compress gz /tmp/solidfire-dtemp.1L6bdX/supportbundle<br><br>Moved
'/tmp/solidfire-dtemp.1L6bdX/supportbundle.nodehostname.tar.gz' to
/tmp/supportbundles",
      "timeoutSec": 1500,
      "url": [

        "https://nodeIP:442/config/supportbundles/supportbundle.nodehostname.tar.g
z"
      ]
    },
    "duration": "00:00:43.101627",
    "result": "Passed"
  }
}
```

New since version

9.6

DeleteAllSupportBundles

You can use the `DeleteAllSupportBundles` method to delete all support bundles generated with the `CreateSupportBundle` API method.

Parameters

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

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

New since version

9.6

DisableMaintenanceMode

You can use the `DisableMaintenanceMode` method to take a storage node out of maintenance mode. You should only disable maintenance mode after you have completed maintenance and the node is online.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
nodes	List of storage node IDs to take out of maintenance mode.	integer array	None	Yes

Return values

This method has the following return values:

Name	Description	Type
------	-------------	------

asyncHandle	You can use the GetAsyncResult method to retrieve this asyncHandle and determine when the maintenance mode transition is complete.	integer
currentMode	The current maintenance mode state of the node. Possible values: • Disabled: No maintenance has been requested. • FailedToRecover: The node failed to recover from maintenance mode. • Unexpected: The node was found to be offline, but was in the Disabled mode. • RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. • PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. • ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)

requestedMode	The requested maintenance mode state of the node. Possible values: • Disabled: No maintenance has been requested. • FailedToRecover: The node failed to recover from maintenance mode. • Unexpected: The node was found to be offline, but was in the Disabled mode. • RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. • PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. • ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)
---------------	---	--------------------------

Request example

Requests for this method are similar to the following example:

```
{
  "method": "DisableMaintenanceMode",
  "params": {
    "nodes": [6]
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:


```
{
  "id": 1,
  "result":
    {
      "requestedMode": "Disabled",
      "asyncHandle": 1,
      "currentMode": "Enabled"
    }
}
```

New since version

12.2

Find more information

[NetApp HCI storage maintenance mode concepts](#)

DisableSsh

You can use the `DisableSsh` method to disable the SSH service for a single storage node. This method does not affect the cluster-wide SSH service timeout duration.

Parameter

This method has no input parameter.

Return value

This method has the following return value:

Name	Description	Type
enabled	The status of the SSH service for this node.	boolean

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

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

EnableMaintenanceMode

You can use the `EnableMaintenanceMode` method to prepare a storage node for maintenance. Maintenance scenarios include any task that requires the node to be powered off or restarted.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
<code>forceWithUnresolvedFaults</code>	Force maintenance mode to be enabled for this node even with blocking cluster faults present.	boolean	False	No
<code>nodes</code>	The list of node IDs to put in maintenance mode. Only one node at a time is supported.	integer array	None	Yes
<code>perMinutePrimarySwapLimit</code>	The number of primary slices to swap per minute. If not specified, all primary slices will be swapped at once.	integer	None	No

Name	Description	Type	Default value	Required
timeout	Specifies how long maintenance mode should remain enabled before it is automatically disabled. Formatted as a time string (for example, HH:mm:ss). If not specified, maintenance mode will remain enabled until explicitly disabled.	string	None	No

Return values

This method has the following return values:

Name	Description	Type
asyncHandle	You can use the GetAsyncResult method to retrieve this asyncHandle and determine when the maintenance mode transition is complete.	integer
currentMode	The current maintenance mode state of the node. Possible values: - Disabled: No maintenance has been requested. - FailedToRecover: The node failed to recover from maintenance mode. - RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. - PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. - ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)

requestedMode	The requested maintenance mode state of the node. Possible values: • Disabled: No maintenance has been requested. • FailedToRecover: The node failed to recover from maintenance mode. • RecoveringFromMaintenance: The node is in the process of recovering from maintenance mode. • PreparingForMaintenance: Actions are being taken to prepare a node to have maintenance performed. • ReadyForMaintenance: The node is ready for maintenance to be performed.	MaintenanceMode (string)
---------------	--	--------------------------

Request example

Requests for this method are similar to the following example:

```
{
  "method": "EnableMaintenanceMode",
  "params": {
    "forceWithUnresolvedFaults": False,
    "nodes": [6],
    "perMinutePrimarySwapLimit" : 40,
    "timeout" : "01:00:05"
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "requestedMode": "ReadyForMaintenance",
    "asyncHandle": 1,
    "currentMode": "Disabled"
  }
}
```

New since version

12.2

Find more information

[NetApp HCI storage maintenance mode concepts](#)

EnableSsh

You can use the `EnableSsh` method to enable the Secure Shell (SSH) service for a single node. This method does not affect the cluster-wide SSH timeout duration, and does not exempt the node from having SSH disabled by the global SSH timeout.

Parameter

This method has no input parameter.

Return value

This method has the following return value:

Name	Description	Type
enabled	The status of the SSH service for this node.	boolean

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

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

GetClusterConfig

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

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
cluster	Cluster configuration information the node uses to communicate with the cluster.	cluster

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "cluster": {
      "cipi": "Bond10G",
      "cluster": "ClusterName",
      "ensemble": [
        "1:10.30.65.139",
        "2:10.30.65.140",
        "3:10.30.65.141"
      ],
      "fipsDriveConfiguration": true,
      "mipi": "Bond1G",
      "name": "xxx-en142",
      "nodeID": 4,
      "pendingNodeID": 0,
      "role": "Storage",
      "sipi": "Bond10G",
      "state": "Active",
      "version": "9.1.0"
    }
  }
}
```

New since version

9.6

GetClusterState

You can use the `GetClusterState` API method to indicate if a node is part of a cluster or not.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
cluster	Name of the cluster.	string
state	• Available: Node has not been configured with a cluster name. • Pending: Node is pending for a specific named cluster and can be added. • Active: Node is an active member of a cluster and may not be added to another cluster.	string

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" :
    "cluster" : "Cluster101"
    "state" : "Active"
}
```

New since version

9.6

GetConfig

You can use the `GetConfig` API method to get all configuration information for a node. This API method includes the same information available in both the `GetClusterConfig` and `GetNetworkConfig` API methods.

Parameters

This method has no input parameters.

Return values

This method has the following return value:

Name	Description	Type
config	The configuration details of the cluster. This object contains: • cluster: Cluster information that identifies how the storage node communicates with the storage cluster it is associated with. • network (all interfaces): Network connection types and current settings for each network interface of the node.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

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

New since version

9.6

Find more information

- [GetClusterConfig](#)
- [GetNetworkConfig](#)
- [GetConfig](#)

GetDriveConfig

You can use the `GetDriveConfig` method to get drive information for expected slice and block drive counts as well as the number of slices and block drives that are currently connected to the node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
driveConfig	Information on the drives that are connected to the node.	drive

Request example

Requests for this method are similar to the following example:

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

Response example

Responses for this method are similar to the following example. Due to length, the response contains information for one drive of one storage node only.

```

{
  "id": 1,
  "result": {
    "driveConfig": {
      "drives": [
        {
          "canonicalName": "sda",
          "connected": true,
          "dev": 2052,
          "devPath": "/dev/sdimm0p4",
          "driveType": "Slice",
          "name": "scsi-SATA_VRFSD3400GNCVMT205581853-
part4",
          "path": "/dev/sda4",
          "pathLink": "/dev/sdimm0p4",
          "product": "VRFSD3400GNCVMTKS1",
          "scsiCompatId": "scsi-
SATA_VRFSD3400GNCVMT205581853-part4",
          "scsiState": "Running",
          "securityAtMaximum": false,
          "securityEnabled": false,
          "securityFrozen": true,
          "securityLocked": false,
          "securitySupported": true,
          "serial": "205581853",
          "size": 299988156416,
          "slot": -1,
          "uuid": "9d4b198b-5ff9-4f7c-04fc-
3bc4e2f38974",
          "vendor": "Viking",
          "version": "612ABBF0"
        }
      ],
      "numBlockActual": 10,
      "numBlockExpected": 10,
      "numSliceActual": 1,
      "numSliceExpected": 1,
      "numTotalActual": 11,
      "numTotalExpected": 11
    }
  }
}

```

GetHardwareConfig

You can use the `GetHardwareConfig` method to get the hardware configuration information for a node. This configuration data is intended for internal use. To get a more useful live system hardware component inventory, use the `GetHardwareInfo` method instead.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
hardwareConfig	List of hardware information and current settings.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

Responses for this method are similar to the following example.

```
{
  "id": 1,
  "result": {
    "hardwareConfig": {
      "biosRevision": "1.0",
      "biosVendor": [
        "NetApp",
        "SolidFire"
      ],
      "biosVersion": "1.1.2",
      "blockDriveSizeBytes": 300069052416,
      "blockDrives": [
```

```

        "/dev/slot0",
        "/dev/slot1",
        "/dev/slot2",
        "/dev/slot3",
        "/dev/slot4",
        "/dev/slot5",
        "/dev/slot6",
        "/dev/slot7",
        "/dev/slot8",
        "/dev/slot9"
    ],
    "blockServiceFormat": "Standard",
    "bmcFirmwareRevision": "1.6",
    "bmcIpmiVersion": "2.0",
    "chassisType": "R620",
    "cpuCores": 6,
    "cpuCoresEnabled": 6,
    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
    "cpuThreads": 12,
    "driveSizeBytesInternal": 400088457216,
    "fibreChannelFirmwareRevision": "",
    "fibreChannelModel": "",
    "fibreChannelPorts": {},
    "idracVersion": "1.06.06",
    "ignoreFirmware": [],
    "memoryGB": 72,
    "memoryMhz": 1333,
    "networkDriver": [
        "bnx2x"
    ],
    "nicPortMap": {
        "PortA": "eth2",
        "PortB": "eth3",
        "PortC": "eth0",
        "PortD": "eth1"
    },
    "nodeType": "SF3010",
    "numCpu": 2,
    "numDrives": 10,
    "numDrivesInternal": 1,
    "nvramTempMonitorEnable": false,
    "rootDrive": "/dev/sdimm0",
    "scsiBusExternalDriver": "mpt3sas",
    "scsiBusInternalDriver": "ahci",
    "sliceDriveSizeBytes": 299988156416,
    "sliceDrives": [

```

```

        "/dev/sdimm0p4"
    ],
    "slotOffset": 0,
    "solidfireDefaults": {
        "bufferCacheGB": 12,
        "configuredIops": 50000,
        "cpuDmaLatency": -1,
        "driveWriteThroughputMBPerSleep": 10,
        "maxDriveWriteThroughputMBPerSec": 175,
        "maxIncomingSliceSyncs": 10,
        "postCallbackThreadCount": 8,
        "sCacheFileCapacity": 100000000,
        "sliceFileLogFileCapacity": 5000000000
    }
}
}
}

```

New since version

9.6

GetHardwareInfo

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

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
force	Set this "force" parameter to true to run on all nodes in the cluster.	boolean	false	No

Return value

This method has the following return value:

Name	Description	Type
hardwareInfo	Hardware information for the node.	hardwareInfo

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "hardwareInfo": {
      "bus": {
        "core_DMI:0200": {
          "description": "Motherboard",
          "physid": "0",
          "product": "0A47AA",
          "serial": "..AB123456C12354.",
          "version": "C07"
        }
      },
      "driveHardware": [
        {
          "canonicalName": "sdh",
          "connected": true,
          "dev": 2160,
          "devPath": "/dev/disk/by-path/pci-0000:41:00.0-sas-0x500056b37789abf0-lun-0",
          "driveEncryptionCapability": "fips",
          "driveType": "Block",
          "lifeRemainingPercent": 92,
          "lifetimeReadBytes": 175436696911872,
          "lifetimeWriteBytes": 81941097349120,
          "name": "scsi-SATA_INTEL_SSDSC2BB3BTWL12345686300AAA",
          "path": "/dev/sdh",
          "pathLink": "/dev/disk/by-path/pci-0000:41:00.0-sas-0x500056b37789abf0-lun-0",
          "powerOnHours": 17246,

```

```

        "product": "INTEL SSDAA2AA300A4",
        "reallocatedSectors": 0,
        "reserveCapacityPercent": 100,
        "scsiCompatId": "scsi-SATA_INTEL_SSDSC2BB3BTWL12345686300AAA",
        "scsiState": "Running",
        "securityAtMaximum": false,
        "securityEnabled": false,
        "securityFrozen": false,
        "securityLocked": false,
        "securitySupported": true,
        "serial": "AAAA33710886300AAA",
        "size": 300069052416,
        "slot": 1,
        "smartSsdWriteCapable": false,
        "uuid": "aea178b9-c336-6bab-a61d-87b615e8120c",
        "vendor": "Intel",
        "version": "D2010370"
    },
    ...
]
}
}
}

```

New since version

9.6

GetIpmiConfig

You can use the `GetIpmiConfig` method to retrieve hardware sensor information from sensors that are in your node.

Parameter

This method has the following input parameter:

Name	Description	Type
chassisType	Used to display information for each node chassis type. Possible values: - all: returns sensor information for each chassis type. {chassis type}: returns sensor information for a specified chassis type.	string

Return values

This method has the following return values:

Name	Description	Type
sensorName	Name of the sensor that has been found.	string
uniqueSensorID	Unique identifier for the sensor.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetIpmiConfig",
  "params": {
    "chassisType": "all"
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "nodes": [
      {
        "nodeID": 1,
        "result": {
```

```
"ipmiConfig": {
  "C220M4": [
    {
      "sensorName": "Fan1A RPM",
      "uniqueSensorID": "29.1:0xf"
    },
    {
      "sensorName": "Fan1B RPM",
      "uniqueSensorID": "29.1:0x10"
    },
    {
      "sensorName": "Fan2A RPM",
      "uniqueSensorID": "29.2:0x11"
    },
    {
      "sensorName": "Fan2B RPM",
      "uniqueSensorID": "29.2:0x12"
    },
    {
      "sensorName": "Fan3A RPM",
      "uniqueSensorID": "29.3:0x13"
    },
    {
      "sensorName": "Fan3B RPM",
      "uniqueSensorID": "29.3:0x14"
    },
    {
      "sensorName": "Fan4A RPM",
      "uniqueSensorID": "29.4:0x15"
    },
    {
      "sensorName": "Fan4B RPM",
      "uniqueSensorID": "29.4:0x16"
    },
    {
      "sensorName": "Fan5A RPM",
      "uniqueSensorID": "29.5:0x17"
    },
    {
      "sensorName": "Fan5B RPM",
      "uniqueSensorID": "29.5:0x18"
    },
    {
      "sensorName": "Fan6A RPM",
      "uniqueSensorID": "29.6:0x19"
    },
  ],
}
```

```

    {
      "sensorName": "Fan6B RPM",
      "uniqueSensorID": "29.6:0x1a"
    },
    {
      "sensorName": "Exhaust Temp",
      "uniqueSensorID": "7.1:0x1"
    },
    {
      "sensorName": "Inlet Temp",
      "uniqueSensorID": "7.1:0x4"
    },
    {
      "sensorName": "PS1",
      "uniqueSensorID": "10.1:0x26"
    },
    {
      "sensorName": "PS2",
      "uniqueSensorID": "10.2:0x2c"
    }
  ],
  "R620": [
    {
      "sensorName": "Fan1A RPM",
      "uniqueSensorID": "7.1:0x30"
    },
    {
      "sensorName": "Fan1B RPM",
      "uniqueSensorID": "7.1:0x31"
    },
    {
      "sensorName": "Fan2A RPM",
      "uniqueSensorID": "7.1:0x32"
    },
    {
      "sensorName": "Fan2B RPM",
      "uniqueSensorID": "7.1:0x33"
    },
    {
      "sensorName": "Fan3A RPM",
      "uniqueSensorID": "7.1:0x34"
    },
    {
      "sensorName": "Fan3B RPM",
      "uniqueSensorID": "7.1:0x35"
    },
  ],

```

```

{
  "sensorName": "Fan4A RPM",
  "uniqueSensorID": "7.1:0x36"
},
{
  "sensorName": "Fan4B RPM",
  "uniqueSensorID": "7.1:0x37"
},
{
  "sensorName": "Fan5A RPM",
  "uniqueSensorID": "7.1:0x38"
},
{
  "sensorName": "Fan5B RPM",
  "uniqueSensorID": "7.1:0x39"
},
{
  "sensorName": "Fan6A RPM",
  "uniqueSensorID": "7.1:0x3a"
},
{
  "sensorName": "Fan6B RPM",
  "uniqueSensorID": "7.1:0x3b"
},
{
  "sensorName": "Fan7A RPM",
  "uniqueSensorID": "7.1:0x3c"
},
{
  "sensorName": "Fan7B RPM",
  "uniqueSensorID": "7.1:0x3d"
},
{
  "sensorName": "Exhaust Temp",
  "uniqueSensorID": "7.1:0x1"
},
{
  "sensorName": "Inlet Temp",
  "uniqueSensorID": "7.1:0x4"
},
{
  "sensorName": "PS1",
  "uniqueSensorID": "10.1:0x62"
},
{
  "sensorName": "PS2",

```

```
        "uniqueSensorID": "10.2:0x63"
      }
    ],
  }
}
```

New since version

9.6

GetIpmiInfo

You can use the `GetIpmiInfo` method to display a detailed reporting of sensors (objects) for node fans, intake and exhaust temperatures, and power supplies that are monitored by the system.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
sensors	Detailed information from each sensor within a node.	JSON object array

Request example

Requests for this method are similar to the following example:

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

Response example

Due to the length of the returned response for this API method, portions of the response have been intentionally eliminated from this document. What is included are the portions of the hardware information that the system monitors in order to ensure the node is running at optimum performance.

```
{
  "id": 1,
```

```

"result": {
  "ipmiInfo": {
    "sensors": [
      {
        "entityID": "7.1 (System Board)",
        "sensorID": "0x72",
        "sensorName": "SEL",
        "sensorType": "Event Logging Disabled",
        "uniqueSensorID": "7.1:0x72"
      },
      {
        "assertionsEnabled": [ "General Chassis intrusion" ],
        "deassertionsEnabled": [ "General Chassis intrusion" ],
        "entityID": "7.1 (System Board)", "sensorID": "0x73",
        "sensorName": "Intrusion",
        "sensorType": "Physical Security",
        "uniqueSensorID": "7.1:0x73"
      },
      {THIS ENTIRE SECTION IS REPEATED FOR EACH FAN IN THE SYSTEM
        "assertionEvents": [],
        "assertionsEnabled": [],
        "deassertionsEnabled": [],
        "entityID": "7.1 (System Board)",
        "eventMessageControl": "Per-threshold",
        "lowerCritical": "720.000",
        "lowerNonCritical": "840.000",
        "maximumSensorRange": "Unspecified",
        "minimumSensorRange": "Unspecified",
        "negativeHysteresis": "600.000",
        "nominalReading": "10080.000",
        "normalMaximum": "23640.000",
        "normalMinimum": "16680.000",
        "positiveHysteresis": "600.000",
        "readableThresholds": "lcr lnc",
        "sensorID": "0x30",
        "sensorName": "Fan1A RPM",
        "sensorReading": "4440 (+/- 120) RPM",
        "sensorType": "Fan",
        "settableThresholds": "",
        "status": "ok",
        "thresholdReadMask": "lcr lnc",
        "uniqueSensorID": "7.1:0x30"
      },
      .
      .
      .
    ]
  }
}

```

{THIS ENTIRE SECTION IS REPEATED FOR THE EXHAUST TEMPERATURE
OF EACH NODE

```
"assertionEvents": [],  
"assertionsEnabled": [],  
"entityID": "7.1 (System Board)",  
"eventMessageControl": "Per-threshold",  
"lowerCritical": "3.000",  
"lowerNonCritical": "8.000",  
"maximumSensorRange": "Unspecified",  
"minimumSensorRange": "Unspecified",  
"negativeHysteresis": "1.000",  
"nominalReading": "23.000",  
"normalMaximum": "69.000",  
"normalMinimum": "11.000",  
"positiveHysteresis": "1.000",  
"readableThresholds": "lcr lnc unc ucr",  
"sensorID": "0x1",  
"sensorName": "Exhaust Temp",  
"sensorReading": "44 (+/- 1) degrees C",  
"sensorType": "Temperature",  
"settableThresholds": "",  
"status": "ok",  
"uniqueSensorID": "7.1:0x1",  
"upperCritical": "75.000",  
"upperNonCritical": "70.000"  
},
```

{THIS ENTIRE SECTION IS REPEATED FOR THE INLET TEMPERATURE OF
EACH NODE

```
"assertionEvents": [],  
"assertionsEnabled": [],  
"deassertionsEnabled": [],  
"entityID": "7.1 (System Board)",  
"eventMessageControl": "Per-threshold",  
"lowerCritical": "-7.000",  
"lowerNonCritical": "3.000",  
"maximumSensorRange": "Unspecified",  
"minimumSensorRange": "Unspecified",  
"negativeHysteresis": "1.000",  
"nominalReading": "23.000",  
"normalMaximum": "69.000",  
"normalMinimum": "11.000",  
"positiveHysteresis": "1.000",  
"readableThresholds": "lcr lnc unc ucr",  
"sensorID": "0x4",  
"sensorName": "Inlet Temp",  
"sensorReading": "20 (+/- 1) degrees C",
```

```

        "sensorType": "Temperature",
        "settableThresholds": "lcr lnc unc ucr",
        "status": "ok",
        "thresholdReadMask": "lcr lnc unc ucr",
        "uniqueSensorID": "7.1:0x4",
        "upperCritical": "47.000",
        "upperNonCritical": "42.000"
    },
    {THIS ENTIRE SECTION IS REPEATED FOR EACH POWER SUPPLY ON EACH
NODE
        "assertionEvents": [],
        "assertionsEnabled": [],
        "entityID": "10.2 (Power Supply)",
        "eventMessageControl": "Per-threshold",
"maximumSensorRange": "Unspecified",
        "minimumSensorRange": "Unspecified",
        "negativeHysteresis": "Unspecified",
        "nominalReading": "0.000",
        "normalMaximum": "0.000",
        "positiveHysteresis": "Unspecified",
        "readableThresholds": "No Thresholds",
        "sensorID": "0x6d",
        "sensorName": "Voltage 2",
        "sensorReading": "118 (+/- 0) Volts",
        "sensorType": "Voltage",
        "settableThresholds": "No Thresholds", "status": "ok",
"uniqueSensorID": "10.2:0x6d"
    },
    .
    .
    .
    }
]
}
}
}
}

```

New since version

9.6

GetNetworkConfig

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

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
network	Network connection types and current settings for each network interface of the node.	network (all interfaces)

Request example

Requests for this method are similar to the following example:

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

Response example

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

New since version

9.6

Find more information

[GetNetworkConfig](#)

GetNetworkInterface

You can use the `GetNetworkInterface` method to get information about a network interface on a node.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
interface	The name of the interface to get information about for each node. Possible values: • Bond1G • Bond10G	string	None	No
force	Set this parameter to true to run on all nodes in the cluster.	boolean	false	No

Return value

This method has the following return value:

Name	Description	Type
nodes	An array of objects describing the interface for each storage node in the storage cluster. Each object within the array contains the following items: • nodeID: (integer) The ID of the storage node in the storage cluster the interface information applies to. • result: (networkInterface) Interface configuration information for this storage node.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetNetworkInterface",
  "params": {
    "interface": "Bond1G",
    "force": true
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "nodes": [
      {
        "nodeID": 1,
        "result": {
          "interface": {
            "address": "10.117.64.32",
            "addressV6": "::",
            "broadcast": "10.117.79.255",
            "macAddress": "90:b1:1c:42:e0:1e",
            "mtu": 1500,
            "name": "Bond1G",
            "namespace": false,
            "netmask": "255.255.240.0",
            "status": "UpAndRunning",
            "type": "BondMaster",
            "virtualNetworkTag": 0
          }
        }
      },
      {
        "nodeID": 2,
        "result": {
          "interface": {
            "address": "10.117.64.35",
            "addressV6": "::",
            "broadcast": "10.117.79.255",
            "macAddress": "d4:ae:52:7a:ae:23",
            "mtu": 1500,
            "name": "Bond1G",
            "namespace": false,
            "netmask": "255.255.240.0",
            "status": "UpAndRunning",
            "type": "BondMaster",
            "virtualNetworkTag": 0
          }
        }
      },
      {
        "nodeID": 3,
```

```

    "result": {
      "interface": {
        "address": "10.117.64.39",
        "addressV6": "::",
        "broadcast": "10.117.79.255",
        "macAddress": "c8:1f:66:f0:9d:17",
        "mtu": 1500,
        "name": "Bond1G",
        "namespace": false,
        "netmask": "255.255.240.0",
        "status": "UpAndRunning",
        "type": "BondMaster",
        "virtualNetworkTag": 0
      }
    },
    {
      "nodeID": 4,
      "result": {
        "interface": {
          "address": "10.117.64.107",
          "addressV6": "::",
          "broadcast": "10.117.79.255",
          "macAddress": "b8:ca:3a:f5:24:f8",
          "mtu": 1500,
          "name": "Bond1G",
          "namespace": false,
          "netmask": "255.255.240.0",
          "status": "UpAndRunning",
          "type": "BondMaster",
          "virtualNetworkTag": 0
        }
      }
    }
  ]
}

```

New since version

9.6

GetNodeActiveTlsCiphers

You can use the `GetNodeActiveTlsCiphers` method on a single node to get a list of the TLS ciphers that are currently accepted on this node. You can use this method on

management and storage nodes.

Parameter

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
mandatoryCiphers	List of mandatory TLS cipher suites for the node. These are ciphers which are always active on the node.	string
supplementalCiphers	List of supplemental TLS cipher suites for the node.	string

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
    "mandatoryCiphers": [
      "DHE-RSA-AES256-SHA256",
      "DHE-RSA-AES256-GCM-SHA384",
      "ECDHE-RSA-AES256-SHA384",
      "ECDHE-RSA-AES256-GCM-SHA384"
    ],
    "supplementalCiphers": [
      "DHE-RSA-AES128-SHA256",
      "DHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES128-SHA256",
      "ECDHE-RSA-AES128-GCM-SHA256"
    ]
  }
}
```

GetNodeFipsDrivesReport

You can use the `GetNodeFipsDrivesReport` method to check the FIPS 140-2 drive encryption capability status of a single node in the storage cluster. You must run this method against an individual storage node.

Parameter

This method has no input parameter.

Return values

This method has the following return values:

Name	Description	Type
fipsDrives	A JSON object containing the status of FIPS 140-2 feature support for this node. Possible values: • None: Node is not FIPS capable. • Partial: Node is FIPS capable but not all drives in the node are FIPS drives. • Ready: Node is FIPS capable and all drives in the node are FIPS drives (or no drives are present).	string

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "fipsDrives": "None"
  }
}
```

New since version

11.5

GetNodeSSLCertificate

You can use the `GetNodeSSLCertificate` method to retrieve the SSL certificate that

is currently active on the management node.

Parameters



You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
certificate	The full PEM-encoded text of the certificate.	string
details	The decoded information of the certificate.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD\nnVQQGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBACUUDFZlZ2FzLCBCYXJ5ITEhMB8G\nnA1UEChMYV2hhbC
BIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\nnaGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXNwHhcNMTCwMzA4MjI1MDI2WhcN\nnMjcwMzA2MjI1MDI2WjCBgzELMAkGA1
UEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYD\nnVQQHFAXWZWdhcywgQmFieSExITAfBgNVBAoTGF
```



```

doYXQgSGFwcGVucyBpbiBWZWdh\ncy4uLjEtMCsGCSqGSib3DQEJARYed2hhdGhhcHBlbnNAdm
VnYXNzdGF5c2luLnZl\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAE8U+28f
nLKQNWEMMR\n6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FH
P4\nt6P/Si0aYr4ovx15wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nmrgp
GZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jy10DmVNU\nnZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\nn7dg9cyXu4afXcgKy14Ji
NBvqbBjhGJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nnq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBrvvBRPno5S34zGRhrnDJyTsdnEbTCB\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrn
DJyTsdnEbaGBiasBhjCBgzELMAkG\na1UEBhMCVVMxCzAJBgNVBAgTAK5WMRUwEwYDVQQHFAxW
ZWdhcywgQmFieSExITAf\nBgNVBAoTGFdoYXQgSGFwcGVucyBpbiBWZWdhcy4uLjEtMCsGCSqG
Sib3DQEJARYe\nd2hhdGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYD
VR0T\nBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s7lmQPECwVLfiE/ndtIbnpe\nnMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7Jyfs3714rRolFpX6N\nniebEgAae5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dH0lQ+WYnoO/eIMy0coT26JB15H\nnDEwvdl+DwkxnS1cx1v
ERv51glgua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nnHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\nn7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDg
l0QoQDWNDoTerBs80nwiouA==\n-----END CERTIFICATE-----\n",
    "details": {
      "issuer":
"/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org",
      "modulus":
"F14FB6F1F9CB290356116311E9A91E0CAB9E852A52EFDA1D2C68A0235F2A94257F0146396
4B8EAB138C1BD325546FE38CA809380DAF1DFA53B1473F8B7A3FF4A2D1A62BE28BF1979C03
A44337432CB924F07B25E94E07A003EDF9A24F078FDB41D162966F63E533ECB6041429AB82
9199405DE239221C047B4B284E75F3A2554FA8F9760EB28D41903B7E76CA573D1D71DC9FA9
5BFE3CA5D0399535467471A430026212DC99A8CB1FB38FF61AE162AAFB64AA4C05FB6D7D05
DF01C77D79D99479CCF1F113E4DFFD03E2BA952EDD83D7325EEE1A7D77202B2D78262341BE
A6C18E1809B44EFAC80CBAAD31EED313378E376471BF58F2688DCF117E002ABE8AD6B",
      "notAfter": "2027-03-06T22:50:26Z",
      "notBefore": "2017-03-08T22:50:26Z",
      "serial": "CC1B221598E37FF3",
      "sha1Fingerprint":
"1D:70:7A:6F:18:8A:CD:29:50:C7:95:B1:DD:5E:63:21:F4:FA:6E:21",
      "subject":
"/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org"
    }
  }
}

```

GetNodeSupportedTlsCiphers

You can use the `GetNodeSupportedTlsCiphers` method on a single node to get a list of the TLS ciphers that are currently supported on this node. You can use this method on management and storage nodes.

Parameter

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
mandatoryCiphers	List of mandatory TLS cipher suites for the node. These are ciphers which are always active on the node.	string
defaultSupplementalCiphers	List of default supplemental TLS cipher suites for the node. The supplemental ciphers are restored to this list when you run the <code>ResetNodeSupplementalTlsCiphers</code> API method.	string
supportedSupplementalCiphers	List of available supplemental TLS cipher suites which you can configure with the <code>SetNodeSupplementalTlsCiphers</code> API method.	string

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```

{
  "id" : 1,
  "result" : {
    "defaultSupplementalCiphers": [
      "DHE-RSA-AES128-SHA256",
      "DHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES128-SHA256",
      "ECDHE-RSA-AES128-GCM-SHA256"
    ],
    "mandatoryCiphers": [
      "DHE-RSA-AES256-SHA256",
      "DHE-RSA-AES256-GCM-SHA384",
      "ECDHE-RSA-AES256-SHA384",
      "ECDHE-RSA-AES256-GCM-SHA384"
    ],
    "supportedSupplementalCiphers": [
      "DHE-RSA-AES128-SHA256",
      "DHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES128-SHA256",
      "ECDHE-RSA-AES128-GCM-SHA256",
      "DHE-RSA-AES256-SHA",
      "ECDHE-RSA-AES256-SHA",
      "DHE-RSA-CAMELLIA256-SHA",
      "DHE-RSA-AES128-SHA",
      "ECDHE-RSA-AES128-SHA",
      "DHE-RSA-CAMELLIA128-SHA"
    ]
  }
}

```

GetPatchInfo

You can use the `GetPatchInfo` method to get information about Element software patches installed on a storage node.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
force	Force the method to run on all nodes in the storage cluster. You only need this when you issue the API to a cluster IP address instead of a single node. Possible values: • true • false	boolean	false	No

Return values

This method has the following return values:

Name	Description	Type
patches	Object containing information about the patches installed on this node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "GetPatchInfo",
  "params": {
    "force": false,
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "patches": {
      "SUST936": {
        "date": "Wed 09 Dec 2020 10:41:59 PM UTC",
        "description": "BMC fixes",
        "newFiles": [
          "None"
        ],
        "patchedFiles": [
          "Patched_file_1.bin",
          "Patched_file_2.dat",
          "Patched_file_3.tgz"
        ]
      }
    }
  }
}
```

New since version

12.3

GetPendingOperation

You can use the `GetPendingOperation` method to detect an operation on a node that is currently in progress. This method can also be used to report back when an operation has completed.

Parameters

This method has no input parameters.

Return values

This method has the following return values:

Name	Description	Type
pending	Possible values: • true: The operation is still in progress. • false: The operation is no longer in progress.	boolean
operation	Name of operation that is in progress or has completed.	string

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
    "pendingOperation" : {
      "pending" : "true",
      "operation" : "TestDrivesInternal",
    }
  }
}
```

New since version

9.6

GetSshInfo

You can use the `GetSshInfo` method to query the status of the SSH service on a single

node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
result	The status of the SSH service for this node.	boolean

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

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

ListDriveHardware

You can use the `ListDriveHardware` method to list all the drives connected to a node. When used on individual nodes, this method returns drive hardware information. When used on the cluster master node MVIP, this method returns information for all drives on all nodes.

Parameters



The "securitySupported": true line of the method response does not imply that the drives are capable of encryption; only that the security status can be queried. If you have a node type with a model number ending in "-NE", commands to enable security features on these drives will fail.

This method has the following parameter:

Name	Description	Type	Default value	Required
force	Set to true to run this method on all nodes.	boolean	None	No

Return value

This method has the following return value:

Name	Description	Type
driveHardware	Returned drive hardware information for the node.	JSON object array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:


```

{
  "id": 1,
  "result": {
    "driveHardware": [
      {
        "canonicalName": "sda",
        "connected": true,
        "dev": 2048,
        "devPath": "/dev/slot0",
        "driveEncryptionCapability": "fips",
        "driveType": "Slice",
        "lifeRemainingPercent": 98,
        "lifetimeReadBytes": 0,
        "lifetimeWriteBytes": 14012129542144,
        "name": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
        "path": "/dev/sda",
        "pathLink": "/dev/slot0",
        "powerOnHours": 15489,
        "product": "SAMSUNG MZ7GE240HMGR-00003",
        "reallocatedSectors": 0,
        "reserveCapacityPercent": 100,
        "scsiCompatId": "scsi-SATA_SAMSUNG_MZ7GE24S1M9NWAG501251",
        "scsiState": "Running",
        "securityAtMaximum": false,
        "securityEnabled": true,
        "securityFrozen": false,
        "securityLocked": false,
        "securitySupported": true,
        "serial": "S1M9NWAG501251",
        "size": 240057409536,
        "slot": 0,
        "uncorrectableErrors": 0,
        "uuid": "789aa05d-e49b-ff4f-f821-f60eed8e43bd",
        "vendor": "Samsung",
        "version": "EXT1303Q"
      }
    ]
  }
}

```

New since version

9.6

Find more information

[EnableEncryptionAtRest](#)

ListNetworkInterfaces

You can use the `ListNetworkInterfaces` method to list information about each network interface on a node. This API method is intended for use on individual nodes; user ID and password authentication is required for access to individual nodes. However, you can use this method on the cluster if the parameter `force` is given the value `true` in the method call. When the parameter is used on the cluster, all interfaces are listed.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
<code>force</code>	Possible values: • <code>true</code>: Information about all network interfaces in the cluster is returned. • <code>false</code>: No information is returned.	boolean	None	No

Return value

This method has the following return value:

Name	Description	Type
<code>interfaces</code>	A list of configuration information for each network interface of the storage node (or entire storage cluster, if <code>force = true</code>).	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
            },
          ]
        }
      }
    ]
  }
}
```


Parameter

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
networkInterfaceStats	A list of network statistics information, such as the number of dropped packets and various types of network errors, for each network interface of a storage node.	networkInterfaceStats array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```

{
  "networkInterfaceStats": [
    {
      "rxErrors": 1,
      "rxPackets": 1,
      "txErrors": 1,
      "rxDropped": 1,
      "txCarrierErrors": 1,
      "rxOverErrors": 1,
      "rxMissedErrors": 1,
      "txPackets": 1,
      "name": "if_name",
      "rxLengthErrors": 1,
      "collisions": 1,
      "rxFifoErrors": 1,
      "txBytes": 1,
      "rxBytes": 1,
      "rxFrameErrors": 1,
      "rxCrcErrors": 1,
      "txFifoErrors": 1
    }
  ]
}

```

New since version

12.3

ListTests

You can use the `ListTests` method to list the tests that are available to run on a node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
tests	List of tests that can be performed on the node.	string array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "tests": [
      "TestConnectEnsemble",
      "TestConnectMvip",
      "TestConnectSvip",
      "TestDrives",
      "TestHardwareConfig",
      "TestLocateCluster",
      "TestPing",
      "TestLocalConnectivity",
      "TestRemoteConnectivity",
      "TestNetworkConfig"
    ]
  }
}
```

New since version

9.6

ListUtilities

You can use the `ListUtilities` method to list the operations that are available to run on a node.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
utilities	List of utilities currently available to run on the node.	string array

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "utilities": [
      "ResetDrives",
      "ResetNode",
      "RestartNetworking",
      "RestartServices",
      "CreateSupportBundle",
      "DeleteAllSupportBundles",
      "CreateClusterSupportBundle"
    ]
  }
}
```

New since version

9.6

RemoveNodeSSLCertificate

You can use the `RemoveNodeSSLCertificate` method to remove the user SSL

certificate and private key for the management node. After the certificate and private key are removed, the management node is configured to use the default certificate and private key.

Parameters



You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method" : "RemoveNodeSSLCertificate",
  "params" : {},
  "id" : 3
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 3,
  "result" : {}
}
```

ResetDrives

You can use the `ResetDrives` method to proactively initialize drives and remove all data currently residing on a drive. The drive can then be reused in an existing node or used in an upgraded node.

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
drives	List of device names (not driveIDs) to reset.	string	None	Yes
force	Set to true to reset the drive.	boolean	None	Yes

Return value

This method has the following return value:

Name	Description	Type
details	Details of drives that are being reset.	JSON object array

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ResetDrives",
  "params": {
    "drives" : "slot3",
    "force" : true
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```

{
  "id": 1,
  "result": {
    "details": {
      "drives": [
        {
          "drive": "slot3",
          "returnCode": 0,
          "stderr": " * Unlocking /dev/slot9 .[ ok ]\ * Setting master
password /dev/slot9 .[ ok ]\ * Secure erasing /dev/slot9 (hdparm)
[tries=0/1] .....[ ok ]",
          "stdout": ""
        }
      ]
    },
    "duration": "00:00:28.501269",
    "result": "Passed"
  }
}

```

New since version

9.6

ResetNode

You can use the `ResetNode` method to reset a node to the factory settings. All data, packages (software upgrades, etc), configurations, and log files are deleted from the node when you call this method. However, network settings for the node are preserved during this operation. Nodes that are participating in a cluster cannot be reset to the factory settings.

Parameters

The `ResetNode` API can only be used on nodes that are in an "Available" state. It cannot be used on nodes that are "Active" in a cluster, or in a "Pending" state.

CAUTION:

This method clears any customer data that is on the node.

This method has the following input parameters:

Name	Description	Type	Default value	Required
build	Used to specify the URL to a remote Element software image to which the node will be reset.	URL	None	No
force	Set to true to reset the node.	boolean	None	Yes
options	Used to enter specifications for running the reset operations. Details are be provided by NetApp Support, if required.	JSON object	None	No

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "ResetNode",
  "params": {
    "build" : "file:///sf/rtfi/image/filesystem.squashfs",
    "force" : true
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": null,
  "result": {
    "rtfiInfo": {
      "build": "file:///sf/rtfi/image/filesystem.squashfs",
      "generation": "9",
      "options": {
```

```

    "edebug": "",
    "sf_auto": "0",
    "sf_bond_mode": "ActivePassive",
    "sf_check_hardware": "0",
    "sf_disable_otpw": "0",
    "sf_fa_host": "",
    "sf_hostname": "SF-FA18",
    "sf_inplace": "1",
    "sf_inplace_die_action": "kexec",
    "sf_inplace_safe": "0",
    "sf_keep_cluster_config": "0",
    "sf_keep_data": "0",
    "sf_keep_hostname": "0",
    "sf_keep_network_config": "0",
    "sf_keep_paths": "\"/var/log/hardware.xml\"",
    "sf_max_archives": "5",
    "sf_nvram_size": "",
    "sf_oldroot": "",
    "sf_postinst_erase_root_drive": "0",
    "sf_root_drive": "",
    "sf_rtfti_cleanup_state": "",
    "sf_secure_erase": "1",
    "sf_secure_erase_retries": "5",
    "sf_slice_size": "",
    "sf_ssh_key": "1",
    "sf_ssh_root": "1",
    "sf_start_rtfti": "1",
    "sf_status_httpserver": "1",
    "sf_status_httpserver_stop_delay": "5m",
    "sf_status_inject_failure": "",
    "sf_status_json": "0",
    "sf_support_host": "sfsupport.solidfire.com",
    "sf_test_hardware": "0",
    "sf_upgrade": "0",
    "sf_upgrade_firmware": "0",
    "sf_upload_logs_url": ""
  },
  "statusUrlAll": "http://192.168.130.20/status/all.json",
  "statusUrlCurrent": "http://192.168.130.20/status/current.json"
}
}
}

```

New since version

9.6

ResetNodeSupplementalTlsCiphers

You can use the `ResetNodeSupplementalTlsCiphers` method to restore the list of supplemental TLS ciphers to the default. You can use this command on management nodes.

Parameter



You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has no input parameters.

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

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

RestartNetworking

You can use the `RestartNetworking` method to restart the networking services on a

node.

CAUTION:

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

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
force	Set to true to restart networking services on a node.	boolean	None	Yes

Return values

This method has no return values.

Request example

Requests for this method are similar to the following example:

```
{
  "method": "RestartNetworking",
  "params": {
    "force" : true
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

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

New since version

9.6

RestartServices

You can use the `RestartServices` method to restart the services on a node.

Parameters

CAUTION:

This method causes temporary node services interruption.

This method has the following input parameters:

Name	Description	Type	Default value	Required
force	Set to true to restart services on a node.	boolean	None	Yes
service	Service name to be restarted.	string	None	No
action	Action to perform on the service (start, stop, restart).	string	None	No

Return values

This method has the following return values:

Name	Description	Type
details	The output of the service restart procedure, including errors (if any).	JSON object
duration	The time, in seconds, it took to restart services to the node.	string
result	Results of the restart.	string

Request example

Requests for this method are similar to the following example:


```
{
  "method": "RestartServices",
  "params": {
    "force" : true
    "action" : restart,
  }
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "details": "solidfire stop/waiting\nsolidfire start/running, process
7284\n",
    "duration": "00:00:02.541594",
    "result": "Passed"
  }
}
```

New since version

9.6

SetClusterConfig

You can use the `SetClusterConfig` method to set the configuration that a node uses to communicate with the cluster it is associated with. To display the current cluster interface settings for a node, run the `GetClusterConfig` API method.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
cluster	Configuration attributes that should be changed during this method call. Only the fields you want changed need to be added to this method as members in this parameter.	cluster	None	No

Return value

This method has the following return value:

Name	Description	Type
cluster	Configuration information the node uses to communicate with the cluster.	cluster

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetClusterConfig",
  "params": {
    "cluster": {
      "name": "myhost",
      "mipi": "Bond10G"
    },
    "id" : 1
  }
}
```

Response example

This method returns a response similar to the following example:

```

{
  "id" : 1,
  "result" : {
    "cluster" : {
      "cipi" : "Bond10G",
      "cluster" : "QoS",
      "ensemble" : [
        "1:10.10.5.42",
        "2:10.10.5.43",
        "3:10.10.5.44",
        "4:10.10.5.46",
        "5:10.10.5.47"
      ],
      "hostname" : "myhost",
      "mipi" : "Bond10G",
      "nodeID" : 1,
      "sipi" : "Bond10G",
      "state" : "Active"
    }
  }
}

```

New since version

9.6

SetConfig

You can use the `SetConfig` method to set the network and cluster information for the node. This method includes the same settings in a single API method that are available using both `SetClusterConfig` and `SetNetworkConfig` methods. Only the fields you want changed need to be included with this method.

Parameter

CAUTION:

Changing the bond-mode on a node can cause a temporary loss of network connectivity.

This method has the following input parameters:

Name	Description	Type	Default value	Required
cluster	Cluster information that identifies how the storage node communicates with the storage cluster it is associated with.	cluster	None	No
network	Network connection types and current settings for each network interface of the node.	network (all interfaces)	None	No

Return value

This method has the following return value:

Name	Description	Type
config	The new and current configuration for the node. This object contains: • cluster: Cluster information that identifies how the storage node communicates with the storage cluster it is associated with. • network (all interfaces): Network connection types and current settings for each network interface of the node.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetConfig",
  "params": {
    "cluster": {
      "name": "MyHostname"
    },
    "network": {
      "Bond10G": {
        "bond-mode": "ALB"
      }
    }
  }
}
```

Response example

The response from this method is the same as the return for the `GetConfig` method. All fields for the object display and updated values are seen when `SetConfig` is used.

New since version

9.6

Find more information

- [SetClusterConfig](#)
- [SetNetworkConfig](#)
- [GetConfig](#)

SetNetworkConfig

You can use the `SetNetworkConfig` method to set the network configuration for a node. To display the current network settings for a node, run the `GetNetworkConfig` API method.

Parameter

CAUTION:

Changing the bond-mode on a node can cause a temporary loss of network connectivity.

This method has the following input parameter:

Name	Description	Type	Default value	Required
network	An object containing node network settings to modify. You only need to add the fields you want changed to this method as attributes in this parameter.	network (all interfaces)	None	No

Return value

This method has the following return value:

Name	Description	Type
network	The new and current network configuration for the node.	network (all interfaces)

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetNetworkConfig",
  "params": {
    "network": {
      "Bond10G": {
        "bond-mode": "ALB"
      },
      "Bond1G": {
        "netmask": "255.255.224.0"
      },
      "eth0": {
        "method": "bond"
      },
      "lo": {
        "method": "loopback"
      }
    }
  }
}
```

Response example

The response from this method is the same as the response from the `GetNetworkConfig` method. The method displays all members for each object and includes the new values for any changed members.

New since version

9.6

Find more information

- [GetNetworkConfig](#)
- [GetNetworkConfig](#)

SetNodeSSLCertificate

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



After using the API, you must reboot the management node.

Parameters



You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has the following input parameters:

Name	Description	Type	Default value	Required
certificate	The PEM-encoded text version of the certificate. Note: When setting a node or cluster certificate, the certificate must include the extendedKeyUsage extension for serverAuth. This extension allows the certificate to be used without error on common operating systems and browsers. If the extension is not present, the API will reject the certificate as invalid.	string	None	Yes
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+28fnLKQNWEMMR6akeDKuehSpS79odLGigI18qlCV/AUY5\nZLjqsT
jBvTJVRv44yoCTgNrx36U7FHP4t6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP
20HRYpZvY+Uz7LYEFCmrpgGZQF3iOSIcBHtLKE5186JVT6j5dg\n6yjjUGQO352ylc9HXHcn6lb
/jy10DmVNUZ0caQwAmIS3Jmoyx+zj/Ya4WKq+2SqTA\nX7bX0F3wHHfXnZlHnM8fET5N/9A+K6
lS7dg9cyXu4afXcgKy14JiNBvqbBjhGJtE\n76yAy6rTHu0xM3jjdkcb9Y8miNzxF+ACq+itaw
IDAQABAoIBAH1jlIZr6/sltqVW\n00qVC/49dyNu+KWVSq92ti9rFe7hBPueh9gklh78hP9Qli
tLkir3YK4GFsTFUMux\n7z1NRCxA/4LrmLSkAjW2kRXDfVl2bwZq0ua9NefGw9208D2OZvbuOx
k7Put2p6se\nfngNzSjf2SI5DIX3UME5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7w
nq\nt+Am5/Vu1LI6rGiG6oHEW0oGSuH1lesIyXXa2hqkU+1+iF2iGRMTiXac4C8d11NU\nwGIR
CXFJAmsAQ+hQm7pmtsKdEqumj/PIoGXf0BoFVEWaIJIMEgnfuLZp8IelJQXn\nsSFJbk2ECgYEA
```



```
+d5ooU4thZXylWHUZqomaxyzOruA1T53UeH69HiFTrLjvfwuaiqj\nlHzPlhms6hxexwzldzAp
gog/NOM+2bAc0rn0dqvtV4doejt1DZKRqrNCf/cuN2QX\njaCJC1CWau3sEHCckLOhWeY4HaPS
oWq0GKLMkKkKDChB4nWUYg3gSWQkCgYEA9zuN\nHW8GPS+yjixeKXmkK00x/vvxzR+J5HH5znaI
Hss48THyhZxpLr+v30Hy2h0yAlBS\nny5Ja6wsomb0mVe4NxVtVawg2E9vVvTa1UC+TNmFBBuL
RPfjcnjDerrSuQ5lYY+M\nnC9MJtXGfhp//G0bzwsRzZxOBsUJb15tpaZIs9MCgYAJricpkKjM
0xlZ1jdVXsos\nnPilnbho4qLngrzuUuxKXEPEnzBxUOqCpwQgdzZLYYw788TCVVIVXLEYem2s0
7dDA\nnDTo+WrzQNkvC6IgqtXH1RgqegIoG1VbgQsbsYmDhdaQ+os4+AOeQXw3vgAhJ/qNJ\nnjQ
4Ttw3ylt7FYkRH26ACWQKBgQC74Zmf4JuRLAo5WSZFxpcmMvtnlvdutqUH4kXA\nnzPssy6t+QE
La1fFbAXkZ5Pg1ITK752aiaX6KQNG6qRsA3VS1J6drD9/2AofOQU17\n+n+jOkGzmmoXf49Zj3iS
akwg0ZbQNGXNxEsCAUr0BYAobPp9/fB4PbtUs99fvtocFr\nnjS562QKBgCb+JMDP5q7jpUuspj
0obd/ZS+MsomE+gFAMBJ71KFQ7KuoNezNFO+ZE\nn3rnR8AqAm4VMzqRahs2PWNe2H14J4hKu96
qNpNHbsW1NjXdAL9P7oqQIrhGLVdhX\nnInDXvTgXMDmoet4BKnfTelrXFKHgGqXJoczq4JWzGS
IHNgvkrH60\n-----END RSA PRIVATE KEY-----\n",
    "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD\nnVQQGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBACUUDFZlZ2FzLCBCYWJ5ITEhMB8G\nnA1UEChMYV2hhbDc
BIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\nnaGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN\nnMjcwMzA4MjI1MDI2WjCBGzELMAkGA1
UEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYD\nnVQQHFAxWZWdhcywgQmFieSExITAFBgNVBAoTGF
doYXQgSGFwcGVucyBpbWZldh\nncy4uLjEtMCsGCSqGSIb3DQEJARYed2hhbGhhcHBlbnNAdm
VnYXNzdGF5c2luLnZl\nnZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA8U+28f
nLQKNWEWMMR\nn6akeDKuehSpS79odLGigI18q1CV/AUY5ZLjqsTjBvTJVRv44yoCTGnRx36U7FH
P4\nt6P/Si0aYr4ovx15wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nnmrgp
GZQF3iOSIcBhtLKE5186JVT6j5dg6yjUGQ0352ylc9HXHcn6lb/jy10DmVNU\nnZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\nn7dg9cyXu4afXcgKy14Ji
NBvqbBjhGJtE76yAy6rThu0xM3jjdkcb9Y8miNzx+AC\nnq+itawIDAQABO4HrMIHoMB0GA1Ud
DgQWBBrvvBRPno5S34zGRhrnDjYtsdnEbTCB\nnuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrn
DjYtsdnEbaGBiaSBhjCBGzELMAkG\nnA1UEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYDVQQHFAxW
ZWdhcywgQmFieSExITAF\nnBgNVBAoTGFdoYXQgSGFwcGVucyBpbWZldhcy4uLjEtMCsGCSqG
SIb3DQEJARYe\nnd2hhbGhhcHBlbnNAdmVnYXNzdGF5c2luLnZlZ2FzgkAzBsiFZjJf/MwDAYD
VR0T\nnBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe\nnMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7Jyfs3714rRolFpX6N\nniebEgAaE5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dH0lQ+WYnoO/eIMy0coT26JB15H\nnDEwvdl+DwkxnS1cx1v
ERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nnHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\nn7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAJJDg
l0QoQDWNDoTerBsZ80nwiouA==\n-----END CERTIFICATE-----\n"
    },
    "id" : 2
}

```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 2,
  "result" : {}
}
```

SetNodeSupplementalTlsCiphers

You can use the `SetNodeSupplementalTlsCiphers` method to specify the list of supplemental TLS ciphers. You can use this command on management nodes.

Parameter



You must call this method against the management node. For example:

```
https://<management node IP>:442/json-rpc/10.0
```

This method has the following input parameter:

Name	Description	Type	Default value	Required
supplementalCiphers	The supplemental cipher suite names using the OpenSSL naming scheme. Use of cipher suite names is case-insensitive.	string	None	Yes

Return values

This method has the following return values:

Name	Description	Type
mandatoryCiphers	List of mandatory TLS cipher suites for the node. These are ciphers which are always active on the node.	string
supplementalCiphers	List of supplemental TLS cipher suites for the node.	string

Request example

Requests for this method are similar to the following example:

```
{
  "method": "SetNodeSupplementalTlsCiphers",
  "params": {
    "supplementalCiphers": [
      "DHE-RSA-AES128-SHA256",
      "DHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES128-SHA256",
      "ECDHE-RSA-AES128-GCM-SHA256"
    ]
  },
  "id": 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id" : 1,
  "result" : {
    "mandatoryCiphers": [
      "DHE-RSA-AES256-SHA256",
      "DHE-RSA-AES256-GCM-SHA384",
      "ECDHE-RSA-AES256-SHA384",
      "ECDHE-RSA-AES256-GCM-SHA384"
    ],
    "supplementalCiphers": [
      "DHE-RSA-AES128-SHA256",
      "DHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES128-SHA256",
      "ECDHE-RSA-AES128-GCM-SHA256"
    ]
  }
}
```

Shutdown

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

Parameters

This method has the following input parameters:

Name	Description	Type	Default value	Required
nodes	List of NodeIDs for the nodes to be restarted or shut down.	integer array	None	Yes
option	Action to take for the cluster. Possible values: • restart: Restarts the cluster. • halt: Performs a full power-off.	string	restart	No

Return value

This method has no return value.

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```

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

```

New since version

9.6

TestConnectEnsemble

You can use the `TestConnectEnsemble` method to verify connectivity with a specified database ensemble. By default it uses the ensemble for the cluster the node is associated with. Alternatively you can provide a different ensemble to test connectivity.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
ensemble	A comma-separated list of ensemble node cluster IP addresses for connectivity testing.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
details	Objects returned: <code>nodes</code>: (object) A list of each ensemble node in the test and the results of the tests. <code>duration</code>: (string) The time required to run the test. <code>result</code>: (string) The results of the entire test.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "details": {
      "nodes": {
        "1:10.10.20.70": "Passed",
        "2:10.10.20.71": "Passed",
        "3:10.10.20.72": "Passed",
        "4:10.10.20.73": "Passed",
        "5:10.10.20.74": "Passed"
      }
    },
    "duration": "00:00:00:756072",
    "result": "Passed"
  }
}
```

New since version

9.6

TestConnectMvip

You can use the `TestConnectMvip` method to test the management connection to the storage cluster. The test pings the MVIP and executes a simple API method to verify connectivity.

Parameter

This method has the following input parameter:

Name	Description	Type	Default value	Required
mvip	You can pass this value to test the management connection of a different MVIP. You do not need to use this value when testing the connection to the target cluster.	string	None	No

Return value

This method has the following return value:

Name	Description	Type
details	Information about the test operation (JSON object): • connected: Indicates if the test could connect to the MVIP (boolean) • mvip: The MVIP tested against (string) • pingBytes: Details of the ping tests with 56 bytes and 1500 bytes (object) ◦ 56: Results of the 56 Byte ping test (JSON object): ▪ individualResponseTimes: List of response times from each ensemble node (string array) ▪ individualStatus: List of ping status from each ensemble node (boolean array) ▪ responseTime: Average ping response time (string) ▪ successful: Indicates if the ping test was successful (boolean) ◦ 1500: Results of the 1500 byte ping test (JSON object): ▪ individualResponseTimes: List of response times from each ensemble node (string array) ▪ individualStatus: List of ping status from each ensemble node (boolean array) ▪ responseTime: Average ping response time (string) ▪ successful: Whether the ping test was successful (boolean) duration: Length of time required to run the test (string)	JSON object

Request example

Requests for this method are similar to the following example as a whole (string)

```
{
  "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)	string

Request example

Requests for this method are similar to the following example. Result of the test as a whole (string)

```
{
  "method": "TestConnectSvip",
  "params": {
    "svip" : "172.27.62.50"
  },
  "id" : 1
}
```

Response example

This method returns a response similar to the following example:

```
{
  "id": 1,
  "result": {
    "details": {
      "connected": true,
      "pingBytes": {
        "56": {
          "individualResponseTimes": [
            "00:00:00.000152",
            "00:00:00.000132",
            "00:00:00.000119",
            "00:00:00.000114",
            "00:00:00.000112"
          ],
          "individualStatus": [
            true,
            true,
            true,
            true,
            true
          ],
          "responseTime": "00:00:00.000126",
          "successful": true
        },
        "9000": {
          "individualResponseTimes": [
            "00:00:00.000295",
            "00:00:00.000257",
            "00:00:00.000172",
```

```

        "00:00:00.000172",
        "00:00:00.000267"
    ],
    "individualStatus": [
        true,
        true,
        true,
        true,
        true
    ],
    "responseTime": "00:00:00.000233",
    "successful": true
    }
},
"svip": "172.27.62.50"
},
"duration": "00:00:00.421907",
"result": "Passed"
}
}

```

New since version

9.6

TestDrives

You can use the `TestDrives` method to run a hardware validation on all drives on the node. This method detects hardware failures on the drives and reports any in the results of the validation tests.

Parameters

You can only use the `TestDrives` method on nodes that are not "Active" in a cluster.



This test takes approximately 10 minutes.

This method has the following input parameters:

Name	Description	Type	Default value	Required
force	Set to true to test the drives on the node.	boolean	None	Yes

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



These test are not intended to detect hardware failures.

This method has the following input parameters:

Name	Description	Type	Default value	Required
clean	Starts the hardware configuration test with a clean cache. Possible values: • true: Deletes the cached test results file and reruns the tests. • false: Retrieves a cached test results.	boolean	false	No
force	The force parameter must be included in this method to successfully reset the node.	boolean	None	Yes

Return value

This method has the following return value:

Name	Description	Type
details	Hardware configuration details.	JSON object

Request example

Requests for this method are similar to the following example:

```
{
  "method": "TestHardwareConfig",
  "params": {
    "force": true
  },
  "id" : 1
}
```

Response example

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

New since version

9.6

Find more information

[TestHardwareConfig](#)

TestLocateCluster

You can use the `TestLocateCluster` method to validate that the node can locate the cluster specified in the cluster configuration. The output validates that the cluster has been created and lists the nodes in the cluster ensemble.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
details	Information about the test operation success or failure.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

This method returns a response similar to the following example:

```

{
  "id": 1,
  "result": {
    "details": {
      "complete": true,
      "ensemble": {
        "nodes": [
          {
            "IP": "10.10.5.94",
            "nodeID": 1
          },
          {
            "IP": "10.10.5.107",
            "nodeID": 2
          },
          {
            "IP": "10.10.5.108",
            "nodeID": 3
          }
        ]
      },
      "version": "5.749"
    },
    "duration": "0.0384478sec",
    "result": "Passed"
  }
}

```

New since version

9.6

TestLocalConnectivity

You can use the `TestLocalConnectivity` method to ping the Cluster IP (CIP) of each node in an active cluster.

Parameters

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
details	Individual ping response times for each node in the local, active cluster.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example

Requests for this method are similar to the following example:

```
{
  "id": null,
  "result": {
    "details": {
      "10.26.86.17": {
        individualResponseTimes: [
          "00:00:00.006868",
          "00:00:00.005933",
          "00:00:00.006655",
          "00:00:00.006584",
          "00:00:00.006334"
        ],
        individualStatus: [
          true,
          true,
          true,
          true,
          true
        ],
        responseTime: "00:00:00.006475",
        successful: true
      },
      "10.26.86.18": {
        individualResponseTimes: [
          "00:00:00.006201",
```

```

        "00:00:00.006187",
        "00:00:00.005990",
        "00:00:00.006029",
        "00:00:00.005917"],
    individualStatus: [
        true,
        true,
        true,
        true,
        true
    ],
    "responseTime": "00:00:00.006065",
    "successful": true
},

    "10.26.86.19": {
    individualResponseTimes: [
        "00:00:00.005988",
        "00:00:00.006948",
        "00:00:00.005981",
        "00:00:00.005964",
        "00:00:00.005942"
    ],
    individualStatus: [
        "true",
        "true",
        true,
        true,
        true
    ],
    responseTime: "00:00:00.006165",
    successful: true,
},

    "10.26.86.20": {
    individualResponseTimes: [
        "00:00:00.005926",
        "00:00:00.006072",
        "00:00:00.005675",
        "00:00:00.009904",
        "00:00:00.006225"
    ],
    "individualStatus": [
        true,
        true,
        true,
        true,
        true
    ]
}

```

```

        ],
        responseTime: "00:00:00.006760",
        successful: true
    }
},
"duration": "00:00:00.595982",
"result": "Passed"
}
}

```

New since version

9.6

TestNetworkConfig

You can use the `TestNetworkConfig` method to test that the configured network settings match the network settings being used on the system.

Parameters

When you configure a node with the `SetNetworkConfig` method, in the UI or TUI, the configuration is validated and stored. The `TestNetworkConfig` API test uses the stored configuration for post-validation logic. For example, in the event of a power outage or network failure, you can use this API method to ensure a node is running with the most currently stored network configuration. This validates that there are no errors in the configuration and that the current configuration is in use.

This test is designed to only show failures in the response output. If there are no errors, this test does not return any output. See the following response examples.

This method has no input parameters.

Return value

This method has the following return value:

Name	Description	Type
details	Contains any errors found when validating the currently stored network settings with the running network configuration.	JSON object

Request example

Requests for this method are similar to the following example:

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

Response example 1

If no errors are detected, then no responses are returned.

```
{
  "id" : 1,
  "result": {
    "details": {
      "network": {...}
    },
    "duration": "00:00:00.144514",
    "result": "Passed"
  }
}
```

Response example 2

Example of an MTU Mismatch.

```
{
  "id" : 1,
  "result":
  {
    "details" :
    {
      "error":
      {
        "message" : "Network configuration mismatch on Bond10G:
Incorrect MTU expectedMTU=[1500]  actualMTU=[9600]", name:
"xAssertionFailure"
      }
    },
    "duration": "0.125213sec",
    "result": "Failed"
  }
}
```


Response example 3

Example of a missing static route.

```
{
  "id": 1,
  "result":
  {
    "details" :
    {
      "error":
      {
        "message" : "Network configuration mismatch on Bond1G: Routing
table missing route=[192.168.137.2 via 192.168.159.254 dev Bond1G]", name:
"xAssertionFailure"
      }
    },
    "duration" : "0.128547sec",
    "result" : "Failed"
  }
}
```

New since version

9.6

Find more information

[SetNetworkConfig](#)

TestPing

You can use the `TestPing` method to test network connectivity to all nodes in the cluster on both 1G and 10G interfaces using ICMP packets. The test uses the appropriate MTU sizes for each packet based on the MTU settings in the network configuration. `TestPing` does not create a temporary VLAN interface.

Parameters

This method has the following input parameter:

Name	Description	Type	Default value	Required
attempts	Specifies the number of times the system should repeat the test ping.	integer	5	No
hosts	Specifies a comma-separated list of addresses or hostnames of devices to ping. If no hosts are specified, the method pings the hosts in the storage cluster.	string	None	No
interface	The existing (base) interface from which the pings should be sent. Possible values: • Bond10G: Send pings from the Bond10G interface. • Bond1G: Send pings from the Bond1G interface.	string	None	No
packetSize	Specifies the number of bytes to send in the ICMP packet that is sent to each IP. The number of bytes must be less than the maximum MTU specified in the network configuration.	integer	None	No
pingTimeoutMsec	Specifies the number of milliseconds to wait for each individual ping response.	integer	500 milliseconds	No

Name	Description	Type	Default value	Required
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

Copyright information

Copyright © 2025 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.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, 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) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.