



节点 **API** 方法 Element Software

NetApp
January 15, 2024

目录

节点 API 方法	1
了解更多信息	2
CheckPingOnVlan	2
CheckProposedNodeAdditions	5
CreateClusterSupportBundle	7
CreateSupportBundle	9
DeleteAllSupportBundles	12
禁用维护模式	12
DisableSsh	15
EnableMaintenanceMode	16
EnableSsh	18
GetClusterConfig	19
GetClusterState	20
getConfig	21
GetDriveConfig	23
GetHardwareConfig	25
GetHardwareInfo	27
GetIpmiConfig	29
GetIpmiInfo	34
GetNetworkConfig	37
GetNetworkInterface	38
GetNodeActiveTlsCiphers	41
GetNodeFipsDrivesReport	42
GetNodeSSLCertificate	43
GetNodeSupportedTlsCiphers	45
GetPatchInfo	47
GetPendingOperation	49
GetSshInfo	50
ListDriveHardware	51
ListNetworkInterfaces	54
ListNetworkInterfaceStats	56
ListTests	58
ListUtilities	59
RemoveNodeSSLCertificate	60
ResetDrives	61
ResetNode	63
ResetNodeSupplementalTlsCiphers	65
RestartNetworking	66
RestartServices	67
SetClusterConfig	69
SetConfig	70
SetNetworkConfig	72

SetNodeSSLCertificate	73
SetNodeSupplementalTlsCiphers	76
关闭	77
TestConnectEnsemble	79
TestConnectMvip	80
TestConnectSvip	84
测试驱动器	88
TestHardwareConfig	89
TestLocateCluster	91
测试本地连接	92
TestNetworkConfig	95
TestPing	97
测试远程连接	100

节点 API 方法

您可以使用节点 API 方法配置单个节点。这些方法可在需要配置，已配置但尚未加入集群或正在积极加入集群的单个节点上运行。通过节点 API 方法，您可以查看和修改各个节点以及用于与节点通信的集群网络的设置。您必须对单个节点运行这些方法；不能对集群地址运行每个节点 API 方法。

- [CheckPingOnVlan](#)
- [CheckProposedNodeAdditions](#)
- [CreateClusterSupportBundle](#)
- [CreateSupportBundle](#)
- [DeleteAllSupportBundles](#)
- [禁用维护模式](#)
- [DisableSsh](#)
- [EnableMaintenanceMode](#)
- [EnableSsh](#)
- [GetClusterConfig](#)
- [GetClusterState](#)
- [getconfig](#)
- [GetDriveConfig](#)
- [GetHardwareConfig](#)
- [GetHardwareInfo](#)
- [GetIpmiConfig](#)
- [GetIpmiInfo](#)
- [GetNetworkConfig](#)
- [GetNetworkInterface](#)
- [GetNodeActiveTlsCiphers](#)
- [GetNodeFipsDrivesReport](#)
- [GetNodeSSLCertificate](#)
- [GetNodeSupportedTlsCiphers](#)
- [GetPendingOperation](#)
- [GetSshInfo](#)
- [ListDriveHardware](#)
- [ListNetworkInterfaces](#)
- [ListTests](#)
- [ListUtilities](#)
- [RemoveNodeSSLCertificate](#)
- [ResetDrives](#)

- [ResetNode](#)
- [ResetNodeSupplementalTlsCiphers](#)
- [RestartNetworking](#)
- [RestartServices](#)
- [SetClusterConfig](#)
- [SetConfig](#)
- [SetNetworkConfig](#)
- [SetNodeSSLCertificate](#)
- [SetNodeSupplementalTlsCiphers](#)
- [关闭](#)
- [TestConnectEnsemble](#)
- [TestConnectMvip](#)
- [TestConnectSvip](#)
- [测试驱动器](#)
- [TestHardwareConfig](#)
- [TestLocateCluster](#)
- [测试本地连接](#)
- [TestNetworkConfig](#)
- [TestPing](#)
- [测试远程连接](#)

了解更多信息

- ["SolidFire 和 Element 软件文档"](#)
- ["早期版本的 NetApp SolidFire 和 Element 产品的文档"](#)

CheckPingOnVlan

在执行部署前网络验证时，您可以使用 `CheckPingOnVlan` 方法测试临时 VLAN 上的网络连接。`CheckPingOnVlan` 创建临时 VLAN 接口，使用 VLAN 接口将 ICMP 数据包发送到存储集群中的所有节点，然后删除该接口。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
尝试次数	指定系统应重复执行测试 ping 的次数。	整型	5.	否

Name	Description	Type	默认值	Required
主机	指定要 ping 的设备的地址或主机名列表，以逗号分隔。	string	集群中的节点	否
接口	应从中发送 ping 的现有（基本）接口。 可能值： <ul style="list-style-type: none"> • Bond10G：从绑定 10G 接口发送 ping。 • Bond1G：从绑定 1G 接口发送 ping。 	string	无	是的。
packetSize	指定要在发送到每个 IP 的 ICMP 数据包中发送的字节数。字节数必须小于网络配置中指定的最大 MTU。	整型	无	否
PingTimeoutMsec	指定等待每个 ping 响应的毫秒数。	整型	500 毫秒	否
禁止分段	为 ICMP 数据包启用 DF（Do not Fragment）标志。	boolean	false	否
sourceAddressV4	要在 ICMP ping 数据包中使用的源 IPv4 地址。	string	无	是的。
sourceAddressV6	要在 ICMP ping 数据包中使用的源 IPv6 地址。	string	无	是的。
总计超时秒	指定在发出下一次 ping 尝试或结束此过程之前 ping 应等待系统响应的时间（以秒为单位）。	整型	5.	否
virtualNetworkTag	发送 ping 数据包时要使用的 VLAN ID。	整型	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	列出节点能够与之通信的每个 IP 以及 ping 响应统计信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```

{
  "id": 1,
  "result": {
    "details": {
      "192.168.41.2": {
        "individualResponseCodes": [
          "Success",
          "Success",
          "Success",
          "Success",
          "Success"
        ],
        "individualResponseTimes": [
          "00:00:00.000373",
          "00:00:00.000098",
          "00:00:00.000097",
          "00:00:00.000074",
          "00:00:00.000075"
        ],
        "individualStatus": [
          true,
          true,
          true,
          true,
          true
        ],
        "interface": "Bond10G",
        "responseTime": "00:00:00.000143",
        "sourceAddressV4": "192.168.41.4",
        "successful": true,
        "virtualNetworkTag": 4001
      }
    },
    "duration": "00:00:00.244379",
    "result": "Passed"
  }
}

```

自版本以来的新增功能

11.1

CheckProposedNodeAdditions

您可以使用 `CheckProposedNodeAdditions` 方法测试一组存储节点，以确定是否可以

将其添加到存储集群中而不会出现错误或违反最佳实践的情况。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
nodes	可随时添加到存储集群的存储节点的存储 IP 地址列表。	string array	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
建议的 ClusterValid	指示建议的存储节点是否构成有效的存储集群。可能值： <ul style="list-style-type: none">• true• false	boolean
PropostedClusterErrors	如果使用建议的存储节点创建存储集群，则会发生错误。可能的错误代码： <ul style="list-style-type: none">• nodesNoCapacity：节点没有任何可用容量。• nodesTooLarge：对于活动保护方案，节点在集群容量中所占的比例过大。• nodesConnectFailed：无法连接到节点以查询硬件配置。• nodesQueryFailed：无法查询节点以获取硬件配置。• nodesClusterMember：节点的 IP 地址已在集群中使用。• nonFipsNodeCapable：启用 FIPS 140-2 驱动器加密功能后，无法向存储集群添加不支持 FIPS 的节点。• nonFipsDrivesCapable：启用 FIPS 140-2 驱动器加密功能后，无法将具有不支持 FIPS 的驱动器的节点添加到集群中。	string array

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

11.0

CreateClusterSupportBundle

您可以在管理节点上使用 `CreateClusterSupportBundle` 从集群中的所有节点收集支持包。各个节点支持包将压缩为 `tar.gz` 文件。集群支持包是一个包含节点支持包的 `tar` 文件。此方法只能在管理节点上运行；在存储节点上运行时不起作用。

Parameters



您必须对管理节点调用此方法。例如：

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

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
allowIncomplete	如果无法从一个或多个节点收集捆绑包，则允许脚本继续运行。	boolean	无	否
捆绑包名称	创建的每个支持包的唯一名称。如果未提供名称，则使用 "supportbundle" 和 节点名称作为文件名	string	无	否
mvip	集群的 MVIP。从集群中的所有节点收集捆绑包。如果未指定 nodes 参数，则需要此参数。	string	无	是的。
nodes	要从中收集捆绑包的节点的 IP 地址。使用节点或 mvip 指定要从中收集捆绑包的节点，但不能同时使用这两者。如果未指定 mvip，则需要此参数。	string array	无	是的。
password	集群管理员密码。* 注：* 输入密码时，此密码以文本形式显示。	string	无	是的。
username	集群管理员用户名。	string	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

CreateSupportBundle

您可以使用 `CreateSupportBundle` 在节点的目录下创建支持包文件。创建后，该捆绑包将以 tar 文件的形式存储在节点上（可以通过 `extraArgs` 参数使用 `gz` 压缩选项。）

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
捆绑包名称	支持包的唯一名称。如果未提供名称，则使用 "supportbundle" 和节点名称作为文件名。	string	无	否
非凡的 Args	使用 "-compress gz" 将支持包创建为 tar.gz 文件。	string	无	否
超时秒	支持包脚本运行的秒数。	整型	1500	否

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	支持包的详细信息。可能值： <ul style="list-style-type: none">• bundleName：在 CreateSupportBundleAPI 方法中指定的名称。如果未指定名称，则使用 "supportbundle"。• extraArgs：使用此方法传递的参数。• files：系统创建的支持包文件的列表。• output：创建支持包的脚本的命令行输出。• timeoutSec：支持包脚本在停止之前运行的秒数。• url：指向创建的支持包的 URL。	JSON 对象
Duration	创建支持包所用的时间，格式为 HH：MM：SS.ssssss。	string
结果	支持包操作的成功或失败。	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
  }
}
```

自版本以来的新增功能

9.6

DeleteAllSupportBundles

您可以使用 `DDeleteAllSupportBundles` 方法删除使用 `CreateSupportBundle` API 方法生成的所有支持包。

Parameters

此方法没有输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

禁用维护模式

您可以使用 `DisableMaintenanceMode` 方法将存储节点退出维护模式。只有在完成维护且节点处于联机状态后，才应禁用维护模式。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
nodes	要退出维护模式的存储节点 ID 列表。	整型数组	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
异步处理	您可以使用 GetAsyncResult 方法检索此 asyncHandle 并确定维护模式过渡完成的时间。	整型
当前模式	节点的当前维护模式状态。可能值： <ul style="list-style-type: none">• Disabled：尚未请求任何维护。• FailedToRecover：节点无法从维护模式恢复。• Unexpected：发现节点脱机，但处于 Disabled 模式。• RecoveringFromMaintenance：节点正在从维护模式恢复。• PreparingForMaintenance：正在执行操作以准备节点以执行维护。• ReadyForMaintenance：节点已准备好执行维护。	维护模式（字符串）

重新要求模式	节点请求的维护模式状态。可能值： <ul style="list-style-type: none">• Disabled：尚未请求任何维护。• FailedToRecover：节点无法从维护模式恢复。• Unexpected：发现节点脱机，但处于 Disabled 模式。• RecoveringFromMaintenance：节点正在从维护模式恢复。• PreparingForMaintenance：正在执行操作以准备节点以执行维护。• ReadyForMaintenance：节点已准备好执行维护。	维护模式（字符串）
--------	--	-----------

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

12.2

了解更多信息

["NetApp HCI 存储维护模式概念"](#)

DisableSsh

您可以使用 `DisableSsh` 方法为单个存储节点禁用 SSH 服务。此方法不会影响集群范围的 SSH 服务超时持续时间。

参数

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
enabled	此节点的 SSH 服务状态。	boolean

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

EnableMaintenanceMode

您可以使用 `EnableMaintenanceMode` 方法准备要维护的存储节点。维护情形包括需要关闭或重新启动节点的任何任务。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
<code>forceWithUnresolvedFaults</code>	强制为此节点启用维护模式，即使存在阻止集群故障也是如此。	boolean	false	否
<code>nodes</code>	要置于维护模式的节点 ID 列表。一次仅支持一个节点。	整型数组	无	是的。
<code>perMinutePrimarySwapLimit</code>	每分钟要交换的主分区数。如果未指定，则会一次性交换所有主分区。	整型	无	否
超时	指定维护模式在自动禁用之前应保持启用多长时间。格式化为时间字符串（例如 HH : mm : ss ）。如果未指定，维护模式将保持启用状态，直到显式禁用为止。	string	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
异步处理	您可以使用 <code>GetAsyncResult</code> 方法检索此 <code>asyncHandle</code> 并确定维护模式过渡完成的时间。	整型

当前模式	节点的当前维护模式状态。可能值： <ul style="list-style-type: none">• Disabled：尚未请求任何维护。• FailedToRecover：节点无法从维护模式恢复。• RecoveringFromMaintenance：节点正在从维护模式恢复。• PreparingForMaintenance：正在执行操作以准备节点以执行维护。• ReadyForMaintenance：节点已准备好执行维护。	维护模式（字符串）
重新要求模式	节点请求的维护模式状态。可能值： <ul style="list-style-type: none">• Disabled：尚未请求任何维护。• FailedToRecover：节点无法从维护模式恢复。• RecoveringFromMaintenance：节点正在从维护模式恢复。• PreparingForMaintenance：正在执行操作以准备节点以执行维护。• ReadyForMaintenance：节点已准备好执行维护。	维护模式（字符串）

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

12.2

了解更多信息

["NetApp HCI 存储维护模式概念"](#)

EnableSsh

您可以使用 `EnableSsh` 方法为单个节点启用安全 Shell （SSH） 服务。此方法不会影响集群范围的 SSH 超时持续时间，也不会使节点免于因全局 SSH 超时而禁用 SSH 。

参数

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
enabled	此节点的 SSH 服务状态。	boolean

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

GetClusterConfig

您可以使用 `GetClusterConfig` API 方法返回有关节点用于与其集群通信的集群配置的信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
集群	节点用于与集群通信的集群配置信息。	集群

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
    }
  }
}
```

自版本以来的新增功能

9.6

GetClusterState

您可以使用 `GetClusterState` API 方法指示节点是否属于集群。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
集群	集群的名称。	string
state	<ul style="list-style-type: none"> • Available：节点尚未配置集群名称。 • Pending：节点正在等待特定命名集群的使用，可以添加。 • Active：节点是集群的活动成员，不能添加到其他集群。 	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

getconfig

您可以使用 `getconfig` API 方法获取节点的所有配置信息。此 API 方法包含在 `GetClusterConfig` 和 `GetNetworkConfig` API 方法中提供的相同信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
配置	集群的配置详细信息。此对象包含： <ul style="list-style-type: none">• 集群：集群信息，用于标识存储节点如何与其关联的存储集群进行通信。• 网络（所有接口）：节点的每个网络接口的网络连接类型和当前设置。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例的长度较长，因此它会记录在一个补充主题中。

自版本以来的新增功能

9.6

了解更多信息

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

GetDriveConfig

您可以使用 `GetDriveConfig` 方法获取预期分区和块驱动器计数以及当前连接到节点的分区和块驱动器数量的驱动器信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
driveConfig	有关连接到节点的驱动器的信息。	驱动器

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应类似于以下示例。由于篇幅限制，此响应仅包含一个存储节点中一个驱动器的信息。

```

{
  "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

您可以使用 GetHardwareConfig 方法获取节点的硬件配置信息。此配置数据仅供内部使用。要获取更有用的实时系统硬件组件清单，请改用 GetHardwareInfo 方法。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
硬件配置	硬件信息和当前设置的列表。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应类似于以下示例。

```
{
  "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
        }
    }
}

```

自版本以来的新增功能

9.6

GetHardwareInfo

您可以使用 `GetHardwareInfo` 方法获取单个节点的实时硬件信息和状态。硬件信息通常包括制造商，供应商，版本，驱动器以及其他相关的标识信息。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
【强制】	将此 "force" 参数设置为 true 可在集群中的所有节点上运行。	boolean	false	否

返回值

此方法具有以下返回值：

Name	Description	Type
hardwareInfo	节点的硬件信息。	hardwareInfo

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
    },
    ...
]
}
}
}

```

自版本以来的新增功能

9.6

GetIpmiConfig

您可以使用 `GetIpmiConfig` 方法从节点中的传感器检索硬件传感器信息。

参数

此方法具有以下输入参数：

Name	Description	Type
chassisType	<p>用于显示每个节点机箱类型的信息。可能值：</p> <ul style="list-style-type: none"> all：返回每个机箱类型的传感器信息。 { chassis type }：返回指定机箱类型的传感器信息。 	string

返回值

此方法具有以下返回值：

Name	Description	Type
sensorName	已找到的传感器的名称。	string
uniqueSensorID	传感器的唯一标识符。	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
        }
    ],
}

```

自版本以来的新增功能

9.6

GetIpmiInfo

您可以使用 `GetIpmiInfo` 方法显示系统监控的节点风扇，进气和排气温度以及电源的传感器（对象）的详细报告。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
传感器	节点中每个传感器的详细信息。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此 API 方法返回的响应较长，因此本文档特意删除了部分响应。其中包括系统为确保节点以最佳性能运行而监控的部分硬件信息。

```
{
  "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"
    },
    .
    .
    .
    }
]
}
}
}
}

```

自版本以来的新增功能

9.6

GetNetworkConfig

您可以使用 GetNetworkConfig 方法显示节点的网络配置信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
网络	节点的每个网络接口的网络连接类型和当前设置。	网络（所有接口）

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例的长度较长，因此它会记录在一个补充主题中。

自版本以来的新增功能

9.6

了解更多信息

[GetNetworkConfig](#)

GetNetworkInterface

您可以使用 `GetNetworkInterface` 方法获取有关节点上网络接口的信息。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
接口	用于获取每个节点的相关信息的接口的名称。可能值： <ul style="list-style-type: none">• 绑定 1 G• 绑定 10G	string	无	否
【强制】	将此参数设置为 true 可在集群中的所有节点上运行。	boolean	false	否

返回值

此方法具有以下返回值：

Name	Description	Type
nodes	<p>一个对象数组，用于描述存储集群中每个存储节点的接口。阵列中的每个对象都包含以下项：</p> <ul style="list-style-type: none"> • <code>nodeID</code>：（整型）存储集群中存储节点的 ID 接口信息适用场景。 • 结果网络接口）此存储节点的接口配置信息。 	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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
    }
}
}
```

自版本以来的新增功能

9.6

GetNodeActiveTlsCiphers

您可以在单个节点上使用 `GetNodeActiveTlsCiphers` 方法来获取此节点上当前接受的 TLS 密码列表。您可以在管理和存储节点上使用此方法。

参数

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
MandatoryCiphers	节点的必需 TLS 密码套件列表。这些密码在节点上始终处于活动状态。	string
补充密码	节点的补充 TLS 密码套件列表。	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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

您可以使用 `GetNodeFipsDrivesReport` 方法检查存储集群中单个节点的 FIPS 140-2 驱动器加密功能状态。您必须对单个存储节点运行此方法。

参数

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
fipsDrives	<p>一个 JSON 对象，其中包含此节点的 FIPS 140-2 功能支持状态。可能值：</p> <ul style="list-style-type: none"> • None：节点不支持 FIPS。 • Partial：节点支持 FIPS，但节点中并非所有驱动器都是 FIPS 驱动器。 • Ready：节点支持 FIPS，并且节点中的所有驱动器均为 FIPS 驱动器（或不存在任何驱动器）。 	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

11.5

GetNodeSSLCertificate

您可以使用 `GetNodeSSLCertificate` 方法检索管理节点上当前处于活动状态的 SSL 证书。

Parameters



您必须对管理节点调用此方法。例如：

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

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
certificate	证书的 PEM 编码的完整文本。	string
详细信息	证书的解码信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBBQUAMIGDMQswCQYD\nVQQGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBACUUDFZlZ2FzLCBCYWJ5ITEhMB8G\nA1UEChMYV2hhcC
BIYXBwZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN\nnMjcwMzA2MjI1MDI2WjCBgzELMAkGA1
UEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYD\nVQQHFAxWZWdhcywgQmFieSEExITAfBgNVBAoTGF
doYXQqSGFwcGVucyBpbWZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXNzdGF5c2luLnZl\nnZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAE8U+28f
```

```

nLKQNWEMMR\n6akeDKuehSpS79odLGigI18q1CV/AUY5ZLjqsTjBvTJVRv44yoCTgNr36U7FH
P4\nt6P/Si0aYr4ovx15wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nmrgp
GZQF3iOSIcBHtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jyl0DmVNU\nZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\n7dg9cyXu4afXcgKy14Ji
NBvqbBjhgJtE76yAy6rThu0xM3jjdkcb9Y8miNzxF+AC\nq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBrvBRPno5S34zGRhrnDJyTsdnEbTCB\nuAYDVR0jBIGwMIGtgBRvvBRPno5S34zGRhrn
DJyTsdnEbaGBiaSBhjCBgzELMAkG\nA1UEBhMCVVMxCzAJBgNVBAgTAk5WMRUwEwYDVQHFAXW
ZWdhcywgQmFieSExITAf\nBgNVBAoTGfdoYXQgSGFwcGVucyBpbWZwdhcy4uLjEtMCsGCSqG
SIb3DQEJARYe\nd2hhdGhhcHBlnNAdmVnYXNzdGF5c2luLnZlZ2FzggkAzBsiFZjjf/MwDAYD
VR0T\nBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s7lmQPECwVLfiE/ndtIbnpe\nmMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7Jyfs37l4rRolFpX6N\nniebEgAaE5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dH0lQ+WYnoO/eIMY0coT26JB15H\nnDEwvdl+DwkxnS1cx1v
ERv5lglgua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nnHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\n7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDg
l0QoQDWNDoTerBs80nwioA==\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

您可以在单个节点上使用 `GetNodeSupportedTlsCiphers` 方法来获取此节点当前支持的 TLS 密码列表。您可以在管理和存储节点上使用此方法。

参数

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
MandatoryCiphers	节点的必需 TLS 密码套件列表。这些密码在节点上始终处于活动状态。	string
defaultSupplementalCiphers	节点的默认补充 TLS 密码套件列表。运行 ResetNodeSupplementalTlsCiphers API 方法时，补充密码将还原到此列表中。	string
支持的 SupplementalCiphers	可使用 SetNodeSupplementalTlsCiphers API 方法配置的可用补充 TLS 密码套件列表。	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```

{
  "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

您可以使用 `GetPatchInfo` 方法获取有关存储节点上安装的 Element 软件修补程序的信息。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
强制	<p>强制此方法在存储集群中的所有节点上运行。只有在将 API 问题描述为集群 IP 地址而不是单个节点时，才需要执行此操作。可能值：</p> <ul style="list-style-type: none"> • true • false 	boolean	false	否

返回值

此方法具有以下返回值：

Name	Description	Type
修补程序	包含有关此节点上安装的修补程序的信息的对象。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```

{
  "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"
        ]
      }
    }
  }
}

```

自版本以来的新增功能

12.3

GetPendingOperation

您可以使用 `GetPendingOperation` 方法检测节点上当前正在进行的操作。此方法还可用于在操作完成后进行报告。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
待定	可能值： <ul style="list-style-type: none"> • true：操作仍在进行中。 • false 操作不再进行。	boolean

Name	Description	Type
操作	正在进行或已完成的操作的名称。	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

GetSshInfo

您可以使用 `GetSshInfo` 方法查询单个节点上 SSH 服务的状态。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
结果	此节点的 SSH 服务状态。	boolean

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

ListDriveHardware

您可以使用 ListDriveHardware 方法列出连接到节点的所有驱动器。在单个节点上使用时，此方法将返回驱动器硬件信息。在集群主节点 MVIP 上使用时，此方法将返回所有节点上所有驱动器的信息。

Parameters



方法响应中的 "securitySupported" : true 行并不表示驱动器能够加密；只能查询安全状态。如果您的节点类型的型号以 "-NE" 结尾，则用于在这些驱动器上启用安全功能的命令将失败。

此方法具有以下参数：

Name	Description	Type	默认值	Required
【强制】	设置为 true 可在所有节点上运行此方法。	boolean	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
驱动器硬件	返回节点的驱动器硬件信息。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```

{
  "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"
      }
    ]
  }
}

```

自版本以来的新增功能

9.6

了解更多信息

[EnableEncryptionAtRest](#)

ListNetworkInterfaces

您可以使用 `ListNetworkInterfaces` 方法列出有关节点上每个网络接口的信息。此 API 方法适用于单个节点；访问单个节点需要用户 ID 和密码身份验证。但是，如果在方法调用中为参数 `force` 指定了值 `true`，则可以在集群上使用此方法。如果在集群上使用此参数，则会列出所有接口。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
【强制】	可能值： <ul style="list-style-type: none">• <code>true</code>：返回有关集群中所有网络接口的信息。• <code>false</code>：不返回任何信息。	boolean	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
接口	存储节点（或整个存储集群，如果 <code>force = true</code> ）的每个网络接口的配置信息列表。	网络接口 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "nodes": [
      {
        "nodeID": 1,
        "result": {
          "interfaces": [
            {
              "address": "10.117.80.32",
              "addressV6": "::",
              "broadcast": "10.117.95.255",
              "macAddress": "90:b1:1c:42:e0:1a",
              "mtu": 9000,
              "name": "Bond10G",
              "namespace": false,
              "netmask": "255.255.240.0",
              "status": "UpAndRunning",
              "type": "BondMaster",
              "virtualNetworkTag": 0
            },
            {
              "address": "10.117.64.32",
              "addressV6": "::",
              "broadcast": "10.117.79.255",
              "macAddress": "90:b1:1c:42:e0:1e",
              "mtu": 1500,
              "name": "Bond1G",
              "namespace": false,
              "netmask": "255.255.240.0",
              "status": "UpAndRunning",
              "type": "BondMaster",
              "virtualNetworkTag": 0
            },
            {
              "address": "0.0.0.0",
              "addressV6": "::",
              "broadcast": "0.0.0.0",
              "macAddress": "90:b1:1c:42:e0:1a",
              "mtu": 9000,
              "name": "eth0",
              "namespace": false,
```

自版本以来的新增功能

ListNetworkInterfaceStats

参数

返回值

56

Name	Description	Type
网络接口统计信息	存储节点的每个网络接口的网络统计信息列表，例如丢弃的数据包数量和各种类型的网络错误。	网络接口统计信息 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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
    }
  ]
}
```

自版本以来的新增功能

12.3

ListTests

您可以使用 `ListTests` 方法列出可在节点上运行的测试。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
测试	可对节点执行的测试列表。	string array

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

ListUtilities

您可以使用 ListUtilities 方法列出可在节点上运行的操作。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
实用程序	列出当前可在节点上运行的实用程序。	string array

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

RemoveNodeSSLCertificate

您可以使用 `RemoveNodeSSLCertificate` 方法删除管理节点的用户 SSL 证书和专用密钥。删除证书和专用密钥后，管理节点将配置为使用默认证书和专用密钥。

Parameters



您必须对管理节点调用此方法。例如：

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

此方法没有输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

ResetDrives

您可以使用 `ResetDrives` 方法主动初始化驱动器并删除当前驻留在驱动器上的所有数据。然后，可以在现有节点中重复使用该驱动器，也可以在升级后的节点中使用该驱动器。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
驱动器	要重置的设备名称（非驱动器 ID）列表。	string	无	是的。
【强制】	设置为 true 可重置驱动器。	boolean	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	正在重置的驱动器的详细信息。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

ResetNode

您可以使用 `ResetNode` 方法将节点重置为出厂设置。调用此方法时，所有数据，软件包（软件升级等），配置和日志文件都会从节点中删除。但是，此操作期间会保留节点的网络设置。不能将加入集群的节点重置为出厂设置。

Parameters

`ResetNode` API 只能用于处于 " 可用 " 状态的节点。它不能用于集群中的 " 活动 " 节点或处于 " 待定 " 状态的节点。

小心

此方法将清除节点上的任何客户数据。

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
build	用于指定要将节点重置到的远程 Element 软件映像的 URL 。	URL	无	否
【强制】	设置为 true 可重置节点。	boolean	无	是的。
选项	用于输入运行重置操作的规格。如果需要，NetApp 支持部门将提供详细信息。	JSON 对象	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": null,
  "result": {
    "rtfiInfo": {
      "build": "file:///sf/rtfi/image/filesystem.squashfs",
      "generation": "9",
      "options": {
        "edebug": "",
        "sf_auto": "0",
        "sf_bond_mode": "ActivePassive",
        "sf_check_hardware": "0",
        "sf_disable_otpw": "0",
        "sf_fa_host": "",
        "sf_hostname": "SF-FA18",
        "sf_inplace": "1",
        "sf_inplace_die_action": "kexec",
        "sf_inplace_safe": "0",
        "sf_keep_cluster_config": "0",
        "sf_keep_data": "0",
        "sf_keep_hostname": "0",
        "sf_keep_network_config": "0",
        "sf_keep_paths": "\"/var/log/hardware.xml\"",
        "sf_max_archives": "5",
        "sf_nvram_size": "",
        "sf_oldroot": "",
        "sf_postinst_erase_root_drive": "0",
        "sf_root_drive": "",
        "sf_rtfi_cleanup_state": "",
        "sf_secure_erase": "1",
        "sf_secure_erase_retries": "5",
        "sf_slice_size": "",

```

```

    "sf_ssh_key": "1",
    "sf_ssh_root": "1",
    "sf_start_rtfi": "1",
    "sf_status_httpserver": "1",
    "sf_status_httpserver_stop_delay": "5m",
    "sf_status_inject_failure": "",
    "sf_status_json": "0",
    "sf_support_host": "sfsupport.solidfire.com",
    "sf_test_hardware": "0",
    "sf_upgrade": "0",
    "sf_upgrade_firmware": "0",
    "sf_upload_logs_url": ""
  },
  "statusUrlAll": "http://192.168.130.20/status/all.json",
  "statusUrlCurrent": "http://192.168.130.20/status/current.json"
}
}
}

```

自版本以来的新增功能

9.6

ResetNodeSupplementalTlsCiphers

您可以使用 `ResetNodeSupplementalTlsCiphers` 方法将补充 TLS 密码列表还原为默认值。您可以在管理节点上使用此命令。

参数



您必须对管理节点调用此方法。例如：

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

此方法没有输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

RestartNetworking

您可以使用 `RestartNetworking` 方法重新启动节点上的网络服务。

小心

此方法将重新启动节点上的所有网络服务，从而导致网络连接暂时断开。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
【强制】	设置为 true 可在节点上重新启动网络服务。	boolean	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

RestartServices

您可以使用 `RestartServices` 方法重新启动节点上的服务。

Parameters

小心

此方法会导致节点服务暂时中断。

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
【强制】	设置为 true 可重新启动节点上的服务。	boolean	无	是的。
服务	要重新启动的服务名称。	string	无	否
Action	要对服务执行的操作（启动，停止，重新启动）。	string	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	此服务的输出将重新启动操作步骤，包括错误（如果有）。	JSON 对象
Duration	重新启动节点服务所用的时间（以秒为单位）。	string
结果	重新启动的结果。	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

SetClusterConfig

您可以使用 SetClusterConfig 方法设置节点用于与其关联集群通信的配置。要显示节点的当前集群接口设置，请运行 GetClusterConfig API 方法。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
集群	在此方法调用期间应更改的配置属性。只需将要更改的字段作为此参数的成员添加到此方法中。	集群	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
集群	节点用于与集群通信的配置信息。	集群

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：


```

{
  "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"
    }
  }
}

```

自版本以来的新增功能

9.6

SetConfig

您可以使用 SetConfig 方法设置节点的网络和集群信息。此方法在一个 API 方法中包含与使用 SetClusterConfig 和 SetNetworkConfig 方法相同的设置。此方法只需要包含要更改的字段。

参数

小心

更改节点上的绑定模式会发生原因导致网络连接暂时断开。

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
集群	用于标识存储节点与其关联存储集群通信的集群信息。	集群	无	否

Name	Description	Type	默认值	Required
网络	节点的每个网络接口的网络连接类型和当前设置。	网络（所有接口）	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
配置	<p>节点的新配置和当前配置。此对象包含：</p> <ul style="list-style-type: none"> • 集群：集群信息，用于标识存储节点如何与其关联的存储集群进行通信。 • 网络（所有接口）：节点的每个网络接口的网络连接类型和当前设置。 	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应与 getconfig 方法的返回相同。使用 SetConfig 时，将显示对象的所有字段以及更新的值。

自版本以来的新增功能

9.6

了解更多信息

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

SetNetworkConfig

您可以使用 `SetNetworkConfig` 方法设置节点的网络配置。要显示节点的当前网络设置，请运行 `GetNetworkConfig` API 方法。

参数

小心

更改节点上的绑定模式会发生原因导致网络连接暂时断开。

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
网络	包含要修改的节点网络设置的对象。您只需在此参数中将要更改的字段作为属性添加到此方法中。	网络（所有接口）	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
网络	节点的新网络配置和当前网络配置。	网络（所有接口）

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应与 `GetNetworkConfig` 方法的响应相同。该方法将显示每个对象的所有成员，并包括任何已更改成员的新值。

自版本以来的新增功能

9.6

了解更多信息

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

SetNodeSSLCertificate

您可以使用 `SetNodeSSLCertificate` 方法为管理节点设置用户 SSL 证书和专用密钥。



使用API后、必须重新启动管理节点。

Parameters



您必须对管理节点调用此方法。例如：

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

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
certificate	PEM 编码的证书文本版本。* 注意：* 设置节点或集群证书时，证书必须包括 serverAuth 的 extendedKeyUsage 扩展。通过此扩展，可以在通用操作系统和浏览器上正确使用此证书。如果扩展名不存在，API 将拒绝证书无效。	string	无	是的。
privateKey	PEM 编码的专用密钥文本版本。	string	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method" : "SetNodeSSLCertificate",
  "params" : {
    "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEEowIBAAKCAQEA8U+28fnLKQNWEMR6akeDKuehSpS79odLGigI18qlCV/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/4LrmLSkAjW2kRXDfVl2bwZq0ua9NefGw92O8D2OZvbuOx
k7Put2p6se\nfgNzSjf2SI5DIX3UMe5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7w
nq\nt+Am5/Vu1LI6rGiG6oHEW0oGSuH1lesIyXXa2hqkU+1+iF2iGRMTiXac4C8d11NU\nnWGIR
CXFJAmsAQ+hQm7pmtsKdEqumj/PIoGXf0BoFVEWaIJIMEgnfuLZp8IelJQXn\nsSFJbk2ECgYEA
+d5ooU4thZXylWHUZqomaxyzOruA1T53UeH69HiFTrLjvfwuaiqj\nlHzPlhms6hxexwz1dzAp
gog/NOM+2bAc0rn0dqvtV4doejt1DZKRqrNCf/cuN2QX\nnjaCJC1CWau3sEHCckLOhWeY4HaPS
```



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

SetNodeSupplementalTlsCiphers

您可以使用 `SetNodeSupplementalTlsCiphers` 方法指定补充 TLS 密码列表。您可以在管理节点上使用此命令。

参数



您必须对管理节点调用此方法。例如：

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

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
补充密码	使用 OpenSSL 命名方案的补充密码套件名称。使用密码套件名称不区分大小写。	string	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
MandatoryCiphers	节点的必需 TLS 密码套件列表。这些密码在节点上始终处于活动状态。	string
补充密码	节点的补充 TLS 密码套件列表。	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
    ]
  }
}
```

关闭

您可以使用 `Shutdown` 方法重新启动或关闭集群中的节点。您可以使用此方法关闭单个节点，多个节点或集群中的所有节点。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
nodes	要重新启动或关闭的节点的节点 ID 列表。	整型数组	无	是的。
选项	对集群采取的操作。可能值：* restart：重新启动集群。* halt：执行完全关闭。	string	重新启动	否

返回值

此方法没有返回值。=此方法的请求示例请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

TestConnectEnsemble

您可以使用 `TestConnectEnsemble` 方法验证与指定数据库集合的连接。默认情况下，它会对与节点关联的集群使用集合。或者，您也可以提供其他集合来测试连接。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
集合	用于连接测试的集合节点集群 IP 地址列表，以英文逗号分隔。	string	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	<p>返回的对象：</p> <ul style="list-style-type: none">• <code>nodes</code>：（对象）列出测试中的每个集合节点以及测试结果。• <code>duration</code>：（字符串）运行测试所需的时间。• <code>result</code>：（字符串）整个测试的结果。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
  }
}
```

自版本以来的新增功能

9.6

TestConnectMvip

您可以使用 `TestConnectMvip` 方法测试与存储集群的管理连接。此测试将对 MVIP 执行 ping 操作，并执行一种简单的 API 方法来验证连接。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
mvip	您可以传递此值来测试其他 MVIP 的管理连接。在测试与目标集群的连接时，无需使用此值。	string	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	<p>有关测试操作的信息（JSON 对象）：</p> <ul style="list-style-type: none"> • <code>connected</code>：指示测试能否连接到 MVIP（布尔值） • <code>mVIP</code>：用于测试的 MVIP（字符串） • <code>pingBytes</code>：56 字节和 1500 字节 ping 测试的详细信息（对象） <ul style="list-style-type: none"> ◦ 56：56 字节 ping 测试的结果（JSON 对象）： <ul style="list-style-type: none"> ▪ <code>individualResponseTimes</code>：列出每个集合节点的响应时间（字符串数组） ▪ <code>individualStatus</code>：列出每个集合节点的 ping 状态（布尔数组） ▪ <code>reresponeTime</code>：平均 ping 响应时间（字符串） ▪ <code>successful</code>：指示 ping 测试是否成功（布尔值） ◦ 1500：1500 字节 ping 测试的结果（JSON 对象）： <ul style="list-style-type: none"> ▪ <code>individualResponseTimes</code>：列出每个集合节点的响应时间（字符串数组） ▪ <code>individualStatus</code>：列出每个集合节点的 ping 状态（布尔数组） ▪ <code>reresponeTime</code>：平均 ping 响应时间（字符串） ▪ <code>successful</code>：ping 测试是否成功（布尔值） ◦ <code>duration</code>：运行测试所需的时间长度（字符串） ◦ <code>result</code>：整体测试结果（字符串） 	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "TestConnectMvip",
  "params": {
    "mvip" : "172.27.62.50"
  },
  "id":1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
}
}

```

自版本以来的新增功能

9.6

TestConnectSvip

您可以使用 `TestConnectSvip` 方法测试与存储集群的存储连接。此测试将使用 ICMP 数据包对 SVIP 执行 ping 操作，如果成功，则会作为 iSCSI 启动程序进行连接。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
Svip	您可以传递此值来测试其他 SVIP 的管理连接。在测试与目标集群的连接时，无需使用此值。	string	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	<p>有关测试操作的信息（JSON 对象）：</p> <ul style="list-style-type: none"> • <code>connected</code>：指示测试能否连接到 SVIP（布尔值） • <code>sVIP</code>：用于测试的 SVIP（字符串） • <code>pingBytes</code>：56 字节和 9000 字节 ping 测试的详细信息（对象） <ul style="list-style-type: none"> ◦ 56：56 字节 ping 测试的结果（JSON 对象）： <ul style="list-style-type: none"> ▪ <code>individualResponseTimes</code>：列出每个集合节点的响应时间（字符串数组） ▪ <code>individualStatus</code>：列出每个集合节点的 ping 状态（布尔数组） ▪ <code>reresponeTime</code>：平均 ping 响应时间（字符串） ▪ <code>successful</code>：指示 ping 测试是否成功（布尔值） ◦ 9000：9000 字节 ping 测试的结果（JSON 对象）： <ul style="list-style-type: none"> ▪ <code>individualResponseTimes</code>：列出每个集合节点的响应时间（字符串数组） ▪ <code>individualStatus</code>：列出每个集合节点的 ping 状态（布尔数组） ▪ <code>reresponeTime</code>：平均 ping 响应时间（字符串） ▪ <code>successful</code>：指示 ping 测试是否成功（布尔值） ◦ <code>duration</code>：运行测试所需的时间长度（字符串） ◦ <code>result</code>：整体测试结果（字符串） 	string

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
}
}

```

自版本以来的新增功能

9.6

测试驱动器

您可以使用 `TestDrives` 方法对节点上的所有驱动器运行硬件验证。此方法可检测驱动器上的硬件故障，并在验证测试结果中报告任何故障。

Parameters

您只能在集群中非 " 活动 " 节点上使用 `TestDrives` 方法。



此测试大约需要 10 分钟。

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
【强制】	设置为 true 以测试节点上的驱动器。	boolean	无	是的。
minutes	指定测试运行的分钟数。	整型	10	否

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	有关测试操作成功或失败的信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "TestDrives",
  "params": {
    "force": true,
    "minutes" : 10
  },
  "id" : 1
}
```

响应示例

此方法将返回一个表，其中包含节点中每个驱动器的测试结果。

自版本以来的新增功能

9.6

TestHardwareConfig

您可以使用 `TestHardwareConfig` 方法在节点上执行硬件测试。测试选项包括验证硬件配置，固件版本以及所有驱动器是否都存在。

Parameters



这些测试不用于检测硬件故障。

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
清理	使用干净的缓存启动硬件配置测试。可能值： <ul style="list-style-type: none"> • true：删除缓存的测试结果文件并重新运行测试。 • false：检索缓存的测试结果。 	boolean	false	否
【强制】	要成功重置节点，此方法必须包含 force 参数。	boolean	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	硬件配置详细信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例的长度较长，因此它会记录在一个补充主题中。

自版本以来的新增功能

9.6

了解更多信息

[TestHardwareConfig](#)

TestLocateCluster

您可以使用 `TestLocateCluster` 方法验证节点是否可以找到在集群配置中指定的集群。输出将验证是否已创建集群，并列出了集群集合中的节点。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	有关测试操作成功或失败的信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
  }
}
```

自版本以来的新增功能

9.6

测试本地连接

您可以使用 `TestLocalConnectivity` 方法对活动集群中每个节点的集群 IP（CIP）执行 ping 操作。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	本地活动集群中每个节点的单个 ping 响应时间。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的请求类似于以下示例：

```
{
  "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"
}
}
```

自版本以来的新增功能

9.6

TestNetworkConfig

您可以使用 TestNetworkConfig 方法测试已配置的网络设置是否与系统上正在使用的网络设置匹配。

Parameters

在 UI 或 TUI 中使用 SetNetworkConfig 方法配置节点时，系统会验证并存储此配置。TestNetworkConfig API 测试使用存储的配置进行验证后逻辑。例如，如果发生断电或网络故障，您可以使用此 API 方法确保节点运行的网络配置为最新存储的网络配置。此操作将验证配置中没有错误，并且当前配置正在使用中。

此测试仅用于在响应输出中显示故障。如果没有错误，则此测试不会返回任何输出。请参见以下响应示例。

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	包含在使用正在运行的网络配置验证当前存储的网络设置时发现的任何错误。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例 1

如果未检测到错误，则不会返回任何响应。

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

响应示例 2

MTU 不匹配示例。

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

响应示例 3

缺少静态路由的示例。

```
{
  "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"
  }
}
```

自版本以来的新增功能

9.6

了解更多信息

[SetNetworkConfig](#)

TestPing

您可以使用 `TestPing` 方法测试使用 ICMP 数据包在 1G 和 10G 接口上与集群中所有节点的网络连接。此测试会根据网络配置中的 MTU 设置对每个数据包使用适当的 MTU 大小。`TestPing` 不会创建临时 VLAN 接口。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
尝试次数	指定系统应重复执行测试 ping 的次数。	整型	5.	否

Name	Description	Type	默认值	Required
主机	指定要 ping 的设备 的地址或主机名列表 ，以逗号分隔。如果 未指定任何主机，则 此方法将对存储集群 中的主机执行 ping 操作。	string	无	否
接口	应从中发送 ping 的 现有（基本）接口。 可能值： <ul style="list-style-type: none"> • Bond10G：从绑 定 10G 接口发送 ping。 • Bond1G：从绑 定 1G 接口发送 ping。 	string	无	否
packetSize	指定要在发送到每个 IP 的 ICMP 数据包 中发送的字节数。字 节数必须小于网络配 置中指定的最大 MTU。	整型	无	否
PingTimeoutMsec	指定等待每个 ping 响应的毫秒数。	整型	500 毫秒	否
禁止分段	为 ICMP 数据包启用 DF （ Do not Fragment ） 标志。	boolean	false	否
sourceAddressV4	要在 ICMP ping 数据 包中使用的源 IPv4 地址。	string	无	否
sourceAddressV6	要在 ICMP ping 数据 包中使用的源 IPv6 地址。	string	无	否
总计超时秒	指定在发出下一次 ping 尝试或结束此过 程之前 ping 应等待 系统响应的的时间（以 秒为单位）。	整型	5.	否

Name	Description	Type	默认值	Required
virtualNetworkTag	发送 ping 数据包时要使用的 VLAN ID。 。	整型	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	列出节点能够与之通信的每个 IP 以及 ping 响应统计信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "TestPing",
  "params": {
    "interface": "Bond1G",
    "hosts": "192.168.0.1"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```

{
  "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"
  }
}

```

自版本以来的新增功能

5.0

测试远程连接

您可以使用 `TestRemoteConnectivity` 方法对远程集群的每个节点执行 ping 操作并检

查远程集合数据库连接。要使用此方法返回有用的结果，必须对集群配对。如果远程数据库连接失败，系统的响应将列出例外情况。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
详细信息	每个节点的单个 ping 响应时间。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "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"
}
```

自版本以来的新增功能

9.6

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。