



## **Node API 方法**

### Element Software

NetApp  
November 12, 2025

This PDF was generated from [https://docs.netapp.com/zh-cn/element-software-128/api/reference\\_element\\_api\\_checkpingonvlan.html](https://docs.netapp.com/zh-cn/element-software-128/api/reference_element_api_checkpingonvlan.html) on November 12, 2025. Always check docs.netapp.com for the latest.

# 目录

|                |    |
|----------------|----|
| Node API 方法    | 1  |
| 检查 VLAN 的 Ping | 1  |
| 参数             | 1  |
| 返回值            | 2  |
| 请求示例           | 2  |
| 响应示例           | 2  |
| 自版本以来的新版本      | 3  |
| 检查拟议节点添加       | 3  |
| 参数             | 4  |
| 返回值            | 4  |
| 请求示例           | 5  |
| 响应示例           | 5  |
| 自版本以来的新版本      | 5  |
| 创建集群支持包        | 5  |
| 参数             | 5  |
| 返回值            | 6  |
| 请求示例           | 6  |
| 响应示例           | 7  |
| 自版本以来的新版本      | 7  |
| 创建支持包          | 7  |
| 参数             | 8  |
| 返回值            | 8  |
| 请求示例           | 9  |
| 响应示例           | 9  |
| 自版本以来的新版本      | 9  |
| 删除所有支持包        | 10 |
| 参数             | 10 |
| 返回值            | 10 |
| 请求示例           | 10 |
| 响应示例           | 10 |
| 自版本以来的新版本      | 10 |
| 禁用维护模式         | 10 |
| 参数             | 11 |
| 返回值            | 11 |
| 请求示例           | 12 |
| 响应示例           | 12 |
| 自版本以来的新版本      | 13 |
| 查找更多信息         | 13 |
| 禁用SSH          | 13 |

|           |    |
|-----------|----|
| 参数        | 13 |
| 返回值       | 13 |
| 请求示例      | 13 |
| 响应示例      | 13 |
| 启用维护模式    | 14 |
| 参数        | 14 |
| 返回值       | 14 |
| 请求示例      | 15 |
| 响应示例      | 16 |
| 自版本以来的新版本 | 16 |
| 查找更多信息    | 16 |
| 启用SSH     | 16 |
| 参数        | 16 |
| 返回值       | 16 |
| 请求示例      | 16 |
| 响应示例      | 17 |
| 获取集群配置    | 17 |
| 参数        | 17 |
| 返回值       | 17 |
| 请求示例      | 17 |
| 响应示例      | 18 |
| 自版本以来的新版本 | 18 |
| 获取集群状态    | 18 |
| 参数        | 19 |
| 返回值       | 19 |
| 请求示例      | 19 |
| 响应示例      | 19 |
| 自版本以来的新版本 | 19 |
| GetConfig | 20 |
| 参数        | 20 |
| 返回值       | 20 |
| 请求示例      | 20 |
| 响应示例      | 20 |
| 自版本以来的新版本 | 20 |
| 查找更多信息    | 20 |
| 获取驱动配置    | 21 |
| 参数        | 21 |
| 返回值       | 21 |
| 请求示例      | 21 |
| 响应示例      | 21 |
| 获取硬件配置    | 23 |

|             |    |
|-------------|----|
| 参数          | 23 |
| 返回值         | 23 |
| 请求示例        | 23 |
| 响应示例        | 23 |
| 自版本以来的新版本   | 25 |
| 获取硬件信息      | 25 |
| 参数          | 25 |
| 返回值         | 25 |
| 请求示例        | 26 |
| 响应示例        | 26 |
| 自版本以来的新版本   | 27 |
| 获取 IPMI 配置  | 27 |
| 参数          | 27 |
| 返回值         | 28 |
| 请求示例        | 28 |
| 响应示例        | 28 |
| 自版本以来的新版本   | 31 |
| 获取 IPMI 信息  | 32 |
| 参数          | 32 |
| 返回值         | 32 |
| 请求示例        | 32 |
| 响应示例        | 32 |
| 自版本以来的新版本   | 35 |
| 获取网络配置      | 35 |
| 参数          | 35 |
| 返回值         | 35 |
| 请求示例        | 36 |
| 响应示例        | 36 |
| 自版本以来的新版本   | 36 |
| 查找更多信息      | 36 |
| 获取网络接口      | 36 |
| 参数          | 36 |
| 返回值         | 37 |
| 请求示例        | 37 |
| 响应示例        | 37 |
| 自版本以来的新版本   | 39 |
| 获取节点活动TLS密码 | 39 |
| 参数          | 39 |
| 返回值         | 39 |
| 请求示例        | 40 |
| 响应示例        | 40 |

|               |    |
|---------------|----|
| 获取节点Fips驱动器报告 | 40 |
| 参数            | 41 |
| 返回值           | 41 |
| 请求示例          | 41 |
| 响应示例          | 41 |
| 自版本以来的新版本     | 42 |
| 获取节点SSL证书     | 42 |
| 参数            | 42 |
| 返回值           | 42 |
| 请求示例          | 42 |
| 响应示例          | 42 |
| 获取节点支持的TLS密码  | 44 |
| 参数            | 44 |
| 返回值           | 44 |
| 请求示例          | 44 |
| 响应示例          | 44 |
| 获取补丁信息        | 45 |
| 参数            | 45 |
| 返回值           | 46 |
| 请求示例          | 46 |
| 响应示例          | 46 |
| 自版本以来的新版本     | 47 |
| 获取待处理操作       | 47 |
| 参数            | 47 |
| 返回值           | 47 |
| 请求示例          | 48 |
| 响应示例          | 48 |
| 自版本以来的新版本     | 48 |
| 获取SSH信息       | 48 |
| 参数            | 49 |
| 返回值           | 49 |
| 请求示例          | 49 |
| 响应示例          | 49 |
| 列表驱动器硬件       | 49 |
| 参数            | 49 |
| 返回值           | 50 |
| 请求示例          | 50 |
| 响应示例          | 50 |
| 自版本以来的新版本     | 51 |
| 查找更多信息        | 52 |
| 列出网络接口        | 52 |

|             |    |
|-------------|----|
| 参数          | 52 |
| 返回值         | 52 |
| 请求示例        | 52 |
| 响应示例        | 53 |
| 自版本以来的新版本   | 54 |
| 列出网络接口统计信息  | 54 |
| 参数          | 54 |
| 返回值         | 54 |
| 请求示例        | 55 |
| 响应示例        | 55 |
| 自版本以来的新版本   | 56 |
| 列表测试        | 56 |
| 参数          | 56 |
| 返回值         | 56 |
| 请求示例        | 56 |
| 响应示例        | 56 |
| 自版本以来的新版本   | 57 |
| 列表实用程序      | 57 |
| 参数          | 57 |
| 返回值         | 57 |
| 请求示例        | 57 |
| 响应示例        | 58 |
| 自版本以来的新版本   | 58 |
| 移除节点SSL证书   | 58 |
| 参数          | 58 |
| 返回值         | 59 |
| 请求示例        | 59 |
| 响应示例        | 59 |
| 重置驱动器       | 59 |
| 参数          | 59 |
| 返回值         | 59 |
| 请求示例        | 60 |
| 响应示例        | 60 |
| 自版本以来的新版本   | 61 |
| 重置节点        | 61 |
| 参数          | 61 |
| 返回值         | 61 |
| 请求示例        | 61 |
| 响应示例        | 62 |
| 自版本以来的新版本   | 63 |
| 重置节点补充TLS密码 | 63 |

|           |    |
|-----------|----|
| 参数        | 63 |
| 返回值       | 63 |
| 请求示例      | 63 |
| 响应示例      | 64 |
| 重启网络      | 64 |
| 参数        | 64 |
| 返回值       | 64 |
| 请求示例      | 64 |
| 响应示例      | 65 |
| 自版本以来的新版本 | 65 |
| 重启服务      | 65 |
| 参数        | 65 |
| 返回值       | 66 |
| 请求示例      | 66 |
| 响应示例      | 66 |
| 自版本以来的新版本 | 66 |
| 设置集群配置    | 67 |
| 参数        | 67 |
| 返回值       | 67 |
| 请求示例      | 67 |
| 响应示例      | 67 |
| 自版本以来的新版本 | 68 |
| 设置配置      | 68 |
| 参数        | 68 |
| 返回值       | 69 |
| 请求示例      | 69 |
| 响应示例      | 70 |
| 自版本以来的新版本 | 70 |
| 查找更多信息    | 70 |
| 设置网络配置    | 70 |
| 参数        | 70 |
| 返回值       | 70 |
| 请求示例      | 70 |
| 响应示例      | 71 |
| 自版本以来的新版本 | 71 |
| 查找更多信息    | 71 |
| 设置节点SSL证书 | 71 |
| 参数        | 71 |
| 返回值       | 72 |
| 请求示例      | 72 |
| 响应示例      | 73 |

|                     |    |
|---------------------|----|
| 设置节点补充TLS密码         | 74 |
| 参数                  | 74 |
| 返回值                 | 74 |
| 请求示例                | 74 |
| 响应示例                | 75 |
| 关闭                  | 75 |
| 参数                  | 75 |
| 返回值                 | 76 |
| 请求示例                | 76 |
| 响应示例                | 76 |
| 自版本以来的新版本           | 77 |
| TestConnectEnsemble | 77 |
| 参数                  | 77 |
| 返回值                 | 77 |
| 请求示例                | 78 |
| 响应示例                | 78 |
| 自版本以来的新版本           | 78 |
| TestConnectMvip     | 78 |
| 参数                  | 78 |
| 返回值                 | 79 |
| 请求示例                | 81 |
| 响应示例                | 81 |
| 自版本以来的新版本           | 82 |
| TestConnectSvip     | 82 |
| 参数                  | 82 |
| 返回值                 | 82 |
| 请求示例                | 85 |
| 响应示例                | 85 |
| 自版本以来的新版本           | 86 |
| 试驾                  | 86 |
| 参数                  | 86 |
| 返回值                 | 87 |
| 请求示例                | 87 |
| 响应示例                | 87 |
| 自版本以来的新版本           | 87 |
| 测试硬件配置              | 87 |
| 参数                  | 87 |
| 返回值                 | 88 |
| 请求示例                | 88 |
| 响应示例                | 88 |
| 自版本以来的新版本           | 88 |



|                 |     |
|-----------------|-----|
| 查找更多信息 .....    | 89  |
| 测试定位集群 .....    | 89  |
| 参数 .....        | 89  |
| 返回值 .....       | 89  |
| 请求示例 .....      | 89  |
| 响应示例 .....      | 89  |
| 自版本以来的新版本 ..... | 90  |
| 测试本地连接 .....    | 90  |
| 参数 .....        | 90  |
| 返回值 .....       | 90  |
| 请求示例 .....      | 91  |
| 响应示例 .....      | 91  |
| 自版本以来的新版本 ..... | 93  |
| 测试网络配置 .....    | 93  |
| 参数 .....        | 93  |
| 返回值 .....       | 93  |
| 请求示例 .....      | 93  |
| 响应示例 1 .....    | 94  |
| 响应示例 2 .....    | 94  |
| 响应示例 3 .....    | 95  |
| 自版本以来的新版本 ..... | 95  |
| 查找更多信息 .....    | 95  |
| 测试Ping .....    | 95  |
| 参数 .....        | 95  |
| 返回值 .....       | 97  |
| 请求示例 .....      | 97  |
| 响应示例 .....      | 97  |
| 自版本以来的新版本 ..... | 98  |
| 测试远程连接 .....    | 98  |
| 参数 .....        | 99  |
| 返回值 .....       | 99  |
| 请求示例 .....      | 99  |
| 响应示例 .....      | 99  |
| 自版本以来的新版本 ..... | 101 |

# Node API 方法

## 检查 VLAN 的 Ping

你可以使用 `CheckPingOnVlan` 在执行部署前网络验证时，测试临时 VLAN 上的网络连接性的方法。`CheckPingOnVlan` 创建一个临时 VLAN 接口，使用该 VLAN 接口向存储集群中的所有节点发送 ICMP 数据包，然后删除该接口。

### 参数

该方法具有以下输入参数：

| 名称       | 描述                                                                                                                                                   | 类型     | 默认值    | 必填项 |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------|-----|
| 尝试       | 指定系统应重复执行测试 ping 的次数。                                                                                                                                | 整数     | 5      | 否   |
| hosts    | 指定要 ping 的设备地址或主机名的逗号分隔列表。                                                                                                                           | string | 集群中的节点 | 否   |
| 界面       | 现有（基础）接口，用于发送 ping 请求。可能值： <ul style="list-style-type: none"><li>• Bond10G：从 Bond10G 接口发送 ping 请求。</li><li>• Bond1G：从 Bond1G 接口发送 ping 请求。</li></ul> | string | 无      | 是   |
| 数据包大小    | 指定发送到每个 IP 地址的 ICMP 数据包中要发送的字节数。字节数必须小于网络配置中指定的最大 MTU。                                                                                               | 整数     | 无      | 否   |
| ping超时毫秒 | 指定等待每个 ping 响应的毫秒数。                                                                                                                                  | 整数     | 500毫秒  | 否   |
| 禁止碎片化    | 启用 ICMP 数据包的 DF（不分片）标志。                                                                                                                              | 布尔值    | false  | 否   |

| 名称              | 描述                                                | 类型     | 默认值 | 必填项 |
|-----------------|---------------------------------------------------|--------|-----|-----|
| sourceAddressV4 | ICMP ping 数据包中使用的源 IPv4 地址。                       | string | 无   | 是   |
| sourceAddressV6 | ICMP ping 数据包中使用的源 IPv6 地址。                       | string | 无   | 是   |
| 总超时时间秒          | 指定 ping 命令在发出下一次 ping 尝试或结束进程之前应等待系统响应的时间（以秒为单位）。 | 整数     | 5   | 否   |
| 虚拟网络标签          | 发送 ping 数据包时要使用的 VLAN ID。                         | 整数     | 无   | 是   |

## 返回值

此方法具有以下返回值：

| 名称 | 描述                               | 类型      |
|----|----------------------------------|---------|
| 结果 | 列出节点能够通信的每个 IP 地址以及 ping 响应统计信息。 | JSON 对象 |

## 请求示例

该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": {
    "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
    }
  }
}

```

自版本以来的新版本

11.1

## 检查拟议节点添加

你可以使用 `CheckProposedNodeAdditions` 测试一组存储节点的方法，以查看是否可以将它们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

该方法具有以下输入参数：

| 名称 | 描述                           | 类型    | 默认值 | 必填项 |
|----|------------------------------|-------|-----|-----|
| 节点 | 存储集群中已准备好添加的存储节点的存储 IP 地址列表。 | 字符串数组 | 无   | 是   |

## 返回值

此方法具有以下返回值：

| 名称       | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 类型    |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 提议的集群有效性 | 指示所提议的存储节点是否可以构成有效的存储集群。可能值： <ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 布尔值   |
| 提议的聚类错误  | 使用建议的存储节点创建存储集群时将会出现的错误。可能的错误代码： <ul style="list-style-type: none"><li>• <code>`nodesNoCapacity`</code> 节点没有任何可用容量。</li><li>• <code>nodesTooLarge</code>：节点在集群容量中所占比例过大，不适合采用主动保护方案。</li><li>• <code>`nodesConnectFailed`</code> 无法连接到节点以查询硬件配置。</li><li>• <code>`nodesQueryFailed`</code> 无法查询节点的硬件配置。</li><li>• <code>`nodesClusterMember`</code> 集群中节点的 IP 地址已被占用。</li><li>• <code>`nonFipsNodeCapable`</code> 启用 FIPS 140-2 驱动器加密功能时，无法将不支持 FIPS 的节点添加到存储集群。</li><li>• <code>`nonFipsDrivesCapable`</code> 启用 FIPS 140-2 驱动器加密功能时，无法将具有非 FIPS 功能驱动器的节点添加到集群中。</li></ul> | 字符串数组 |

## 请求示例

该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": [ ]
  }
}
```

## 自版本以来的新版本

11.0

## 创建集群支持包

你可以使用 `CreateClusterSupportBundle` 在管理节点上收集集群中所有节点的支持包。各个节点支持包被压缩为 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；在存储节点上运行无效。

## 参数



必须对管理节点调用此方法。例如：

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

该方法有以下输入参数：

| 名称       | 描述                                                                               | 类型     | 默认值 | 必填项 |
|----------|----------------------------------------------------------------------------------|--------|-----|-----|
| 允许不完整    | 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。                                                     | 布尔值    | 无   | 否   |
| 捆绑包名称    | 每个创建的支持包都有一个唯一的名称。如果未提供名称，则使用“supportbundle”和节点名称作为文件名。                          | string | 无   | 否   |
| 贵宾       | 集群的MVIP。从集群中的所有节点收集数据包。如果未指定节点参数，则此参数为必填项。                                       | string | 无   | 是   |
| 节点       | 要从中收集数据包的节点的 IP 地址。使用 nodes 或 mvip，但不能同时使用两者，来指定要从中收集捆绑包的节点。如果未指定 mvip，则此参数为必填项。 | 字符串数组  | 无   | 是   |
| password | 集群管理员密码。 * 注意：*此密码输入后会以文本形式显示。                                                   | string | 无   | 是   |
| 用户名      | 集群管理员用户名。                                                                        | string | 无   | 是   |

## 返回值

此方法没有返回值。

## 请求示例

该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
/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"
  }
}
```

## 自版本以来的新版本

9.6

## 创建支持包

您可以使用 `CreateSupportBundle` 在节点目录下创建支持包文件。创建完成后，该软件包将以 tar 文件的形式存储在节点上（可通过 extraArgs 参数选择 gz 压缩选项）。



## 参数

该方法有以下输入参数：

| 名称    | 描述                                              | 类型     | 默认值  | 必填项 |
|-------|-------------------------------------------------|--------|------|-----|
| 捆绑包名称 | 支持包的唯一名称。如果没有提供名称，则使用“supportbundle”和节点名称作为文件名。 | string | 无    | 否   |
| 额外参数  | 使用“--compress gz”将支持包创建为 tar.gz 文件。             | string | 无    | 否   |
| 超时时间秒 | 支持包脚本运行的秒数。                                     | 整数     | 1500 | 否   |

## 返回值

此方法具有以下返回值：

| 名称 | 描述                                                                                                                                                                                                                                                                                                  | 类型      |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 细节 | 支持方案的详细信息。可能值： <ul style="list-style-type: none"><li>• bundleName：在 CreateSupportBundleAPI 方法中指定的名称。如果没有指定名称，则使用“supportbundle”。</li><li>• extraArgs：此方法传递的参数。</li><li>• 文件：系统创建的支持包文件列表。</li><li>• 输出：创建支持包的脚本的命令行输出。</li><li>• timeoutSec：支持包脚本在停止前运行的秒数。</li><li>• url：指向已创建的支持包的 URL。</li></ul> | JSON 对象 |
| 期间 | 用于创建支持包的时间，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

该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"
  }
}
```

## 自版本以来的新版本

9.6

## 删除所有支持包

你可以使用 `DeleteAllSupportBundles` 删除所有使用以下方式生成的支持包的方法  
`CreateSupportBundle` API 方法。

### 参数

此方法没有输入参数。

### 返回值

此方法没有返回值。

### 请求示例

该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" : {}
}
```

### 自版本以来的新版本

9.6

## 禁用维护模式

你可以使用 `DisableMaintenanceMode` 将存储节点从维护模式中移除的方法。只有在维护完成后且节点上线后，才应禁用维护模式。

## 参数

该方法有以下输入参数：

| 名称 | 描述                  | 类型   | 默认值 | 必填项 |
|----|---------------------|------|-----|-----|
| 节点 | 要退出维护模式的存储节点 ID 列表。 | 整数数组 | 无   | 是   |

## 返回值

此方法具有以下返回值：

| 名称   | 描述                                                                                                                                                                                                                              | 类型        |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 异步句柄 | 您可以使用 GetAsyncResult 方法来检索此 asyncHandle，并确定维护模式转换何时完成。                                                                                                                                                                          | 整数        |
| 当前模式 | 节点的当前维护模式状态。可能值： <ul style="list-style-type: none"><li>已禁用：未收到维护请求。</li><li>恢复失败：节点未能从维护模式恢复。</li><li>意外情况：发现该节点已离线，但处于禁用模式。</li><li>正在从维护模式中恢复：节点正在从维护模式中恢复。</li><li>维护准备：正在采取措施准备对节点进行维护。</li><li>准备维护：该节点已准备好执行维护。</li></ul> | 维护模式（字符串） |

|      |                                                                                                                                                                                                                                             |           |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 请求模式 | 请求节点的维护模式状态。可能值： <ul style="list-style-type: none"><li>• 已禁用：未收到维护请求。</li><li>• 恢复失败：节点未能从维护模式恢复。</li><li>• 意外情况：发现该节点已离线，但处于禁用模式。</li><li>• 正在从维护模式中恢复：节点正在从维护模式中恢复。</li><li>• 维护准备：正在采取措施准备对节点进行维护。</li><li>• 准备维护：该节点已准备好执行维护。</li></ul> | 维护模式（字符串） |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|

### 请求示例

该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"
  }
}
```

自版本以来的新版本

12.2

查找更多信息

["NetApp HCI存储维护模式概念"](#)

## 禁用SSH

你可以使用 `DisableSsh` 禁用单个存储节点的 SSH 服务的方法。此方法不会影响集群范围内的 SSH 服务超时时长。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称 | 描述             | 类型  |
|----|----------------|-----|
| 启用 | 此节点的 SSH 服务状态。 | 布尔值 |

### 请求示例

该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}
}
```

## 启用维护模式

您可以使用 `EnableMaintenanceMode` 用于准备存储节点以进行维护的方法。维护场景包括任何需要关闭节点电源或重启节点的任务。

### 参数

该方法有以下输入参数：

| 名称                        | 描述                                                                       | 类型     | 默认值   | 必填项 |
|---------------------------|--------------------------------------------------------------------------|--------|-------|-----|
| forceWithUnresolvedFaults | 即使存在阻塞集群故障，也强制为此节点启用维护模式。                                                | 布尔值    | False | 否   |
| 节点                        | 要置于维护模式的节点 ID 列表。一次只能支持一个节点。                                             | 整数数组   | 无     | 是   |
| 每分钟主交换限额                  | 每分钟要交换的主切片数量。如果未指定，则所有主切片将一次性交换。                                         | 整数     | 无     | 否   |
| timeout                   | 指定维护模式在自动禁用前应保持启用状态的时间。格式为时间字符串（例如，HH:mm:ss）。如果未指定，维护模式将保持启用状态，直到明确禁用为止。 | string | 无     | 否   |

### 返回值

此方法具有以下返回值：

| 名称   | 描述                                                     | 类型 |
|------|--------------------------------------------------------|----|
| 异步句柄 | 您可以使用 GetAsyncResult 方法来检索此 asyncHandle，并确定维护模式转换何时完成。 | 整数 |

|      |                                                                                                                                                                                                            |           |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 当前模式 | 节点的当前维护模式状态。可能值： <ul style="list-style-type: none"><li>• 已禁用：未收到维护请求。</li><li>• 恢复失败：节点未能从维护模式恢复。</li><li>• 正在从维护模式中恢复：节点正在从维护模式中恢复。</li><li>• 维护准备：正在采取措施准备对节点进行维护。</li><li>• 准备维护：该节点已准备好执行维护。</li></ul> | 维护模式（字符串） |
| 请求模式 | 请求节点的维护模式状态。可能值： <ul style="list-style-type: none"><li>• 已禁用：未收到维护请求。</li><li>• 恢复失败：节点未能从维护模式恢复。</li><li>• 正在从维护模式中恢复：节点正在从维护模式中恢复。</li><li>• 维护准备：正在采取措施准备对节点进行维护。</li><li>• 准备维护：该节点已准备好执行维护。</li></ul> | 维护模式（字符串） |

请求示例

该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"
  }
}
```

## 自版本以来的新版本

12.2

## 查找更多信息

["NetApp HCI存储维护模式概念"](#)

## 启用SSH

你可以使用 `EnableSsh` 为单个节点启用安全外壳（SSH）服务的方法。此方法不会影响集群范围内的 SSH 超时持续时间，也不会使节点免受全局 SSH 超时导致的 SSH 禁用。

## 参数

此方法没有输入参数。

## 返回值

该方法返回以下值：

| 名称 | 描述             | 类型  |
|----|----------------|-----|
| 启用 | 此节点的 SSH 服务状态。 | 布尔值 |

## 请求示例

该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}
}
```

## 获取集群配置

你可以使用 `GetClusterConfig` API 方法，用于返回节点与集群通信时使用的集群配置信息。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称 | 描述                | 类型                 |
|----|-------------------|--------------------|
| 集群 | 集群配置信息，节点用于与集群通信。 | <a href="#">集群</a> |

## 请求示例

该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"
    }
  }
}
```

自版本以来的新版本

9.6

## 获取集群状态

你可以使用 `GetClusterState` 用于指示节点是否属于集群的 API 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：

| 名称 | 描述                                                                                                                                       | 类型     |
|----|------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 集群 | 集群名称。                                                                                                                                    | string |
| 状态 | <ul style="list-style-type: none"><li>• 可用：节点尚未配置集群名称。</li><li>• 待处理：该节点已等待加入指定的集群，可以添加。</li><li>• 活动节点：该节点是集群的活跃成员，不能添加到其他集群。</li></ul> | string |

## 请求示例

该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"
}
```

## 自版本以来的新版本

9.6

# GetConfig

你可以使用 `GetConfig` 获取节点所有配置信息的 API 方法。此 API 方法包含与以下两者相同的信息：`GetClusterConfig` 和 `GetNetworkConfig` API 方法。

## 参数

此方法没有输入参数。

## 返回值

该方法返回以下值：

| 名称     | 描述                                                                                                                                                                          | 类型      |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| config | 集群的配置详情。此对象包含： <ul style="list-style-type: none"><li>• <a href="#">集群</a>：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。</li><li>• <a href="#">网络（所有接口）</a>：节点的每个网络接口的网络连接类型和当前设置。</li></ul> | JSON 对象 |

## 请求示例

该方法的请求类似于以下示例：

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

## 响应示例

由于此回复示例篇幅较长，因此已在补充主题中进行了记录。

## 自版本以来的新版本

9.6

## 查找更多信息

- [获取集群配置](#)
- [获取网络配置](#)

## 获取驱动配置

你可以使用 `GetDriveConfig` 获取预期切片和块驱动器数量的驱动器信息，以及当前连接到节点的切片和块驱动器的数量的方法。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称   | 描述           | 类型                    |
|------|--------------|-----------------------|
| 驱动配置 | 连接到节点的驱动器信息。 | <a href="#">drive</a> |

### 请求示例

该方法的请求类似于以下示例：

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

### 响应示例

该方法的响应与以下示例类似。由于长度限制，响应仅包含一个存储节点的一个驱动器的信息。

```

{
  "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
    }
  }
}

```

# 获取硬件配置

你可以使用 `GetHardwareConfig` 获取节点硬件配置信息的方法。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，请使用 `GetHardwareInfo` 改用其他方法。

## 参数

此方法没有输入参数。

## 返回值

该方法返回以下值：

| 名称   | 描述           | 类型      |
|------|--------------|---------|
| 硬件配置 | 硬件信息和当前设置列表。 | JSON 对象 |

## 请求示例

该方法的请求类似于以下示例：

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

## 响应示例

该方法的响应与以下示例类似。

```
{
  "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
        }
    }
}
```

自版本以来的新版本

9.6

## 获取硬件信息

你可以使用 `GetHardwareInfo` 获取单个节点的实时硬件信息和状态的方法。硬件信息通常包括制造商、供应商、版本、驱动器和其他相关标识信息。

### 参数

该方法具有以下输入参数：

| 名称 | 描述                                  | 类型  | 默认值   | 必填项 |
|----|-------------------------------------|-----|-------|-----|
| 力量 | 将此“force”参数设置为 true，即可在集群中的所有节点上运行。 | 布尔值 | false | 否   |

### 返回值

该方法返回以下值：

| 名称   | 描述       | 类型                   |
|------|----------|----------------------|
| 硬件信息 | 节点的硬件信息。 | <a href="#">硬件信息</a> |

## 请求示例

该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"
    },
    ...
]
}
}
}

```

自版本以来的新版本

9.6

## 获取 IPMI 配置

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

### 参数

该方法具有以下输入参数：

| 名称   | 描述                                                                                                                                       | 类型     |
|------|------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 底盘类型 | <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"> <li>全部：返回每种底盘类型的传感器信息。</li> <li>{chassis type}：返回指定底盘类型的传感器信息。</li> </ul> | string |

## 返回值

此方法具有以下返回值：

| 名称      | 描述          | 类型     |
|---------|-------------|--------|
| 传感器名称   | 已找到的传感器的名称。 | string |
| 唯一传感器ID | 传感器的唯一标识符。  | string |

## 请求示例

该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"
        }
    ],
}

```

自版本以来的新版本

9.6



# 获取 IPMI 信息

你可以使用 `GetIpmiInfo` 用于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）的详细报告的方法。

## 参数

此方法没有输入参数。

## 返回值

该方法返回以下值：

| 名称  | 描述             | 类型        |
|-----|----------------|-----------|
| 传感器 | 节点内每个传感器的详细信息。 | JSON 对象数组 |

## 请求示例

该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 方法返回的响应内容较长，因此本文档中已有意省略了部分响应内容。其中包括系统监控的硬件信息部分，以确保节点以最佳性能运行。

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

```

自版本以来的新版本

9.6

## 获取网络配置

你可以使用 `GetNetworkConfig` 显示节点网络配置信息的方法。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称      | 描述                     | 类型                       |
|---------|------------------------|--------------------------|
| network | 节点的每个网络接口的网络连接类型和当前设置。 | <a href="#">网络（所有接口）</a> |

## 请求示例

该方法的请求类似于以下示例：

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

## 响应示例

由于此回复示例篇幅较长，因此已在补充主题中进行了记录。

## 自版本以来的新版本

9.6

## 查找更多信息

[获取网络配置](#)

# 获取网络接口

你可以使用 `GetNetworkInterface` 获取节点上网络接口信息的方法。

## 参数

该方法有以下输入参数：

| 名称 | 描述                                                                                           | 类型     | 默认值 | 必填项 |
|----|----------------------------------------------------------------------------------------------|--------|-----|-----|
| 界面 | 获取每个节点信息的接口名称。可能值： <ul style="list-style-type: none"> <li>Bond1G</li> <li>Bond10G</li> </ul> | string | 无   | 否   |

| 名称 | 描述                          | 类型  | 默认值   | 必填项 |
|----|-----------------------------|-----|-------|-----|
| 力量 | 将此参数设置为 true 可在集群中的所有节点上运行。 | 布尔值 | false | 否   |

## 返回值

该方法返回以下值：

| 名称 | 描述                                                                                                                                                                                 | 类型        |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 节点 | <p>描述存储集群中每个存储节点接口的对象数组。数组中的每个对象都包含以下项：</p> <ul style="list-style-type: none"><li>• nodeID: (整数) 接口信息所适用的存储集群中存储节点的 ID。</li><li>• 结果：（<a href="#">网络接口</a>）此存储节点的接口配置信息。</li></ul> | JSON 对象数组 |

## 请求示例

该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
        }
    }
}
]
}
}

```

自版本以来的新版本

9.6

## 获取节点活动**TLS**密码

您可以使用 `GetNodeActiveTlsCiphers` 在单个节点上获取当前在此节点上接受的 TLS 密码列表的方法。您可以将此方法用于管理节点和存储节点。

### 参数

此方法没有输入参数。

### 返回值

此方法具有以下返回值：



| 名称   | 描述                                 | 类型     |
|------|------------------------------------|--------|
| 强制密码 | 节点必须使用的TLS密码套件列表。这些密码始终在节点上处于活动状态。 | string |
| 补充密码 | 节点的补充 TLS 密码套件列表。                  | string |

## 请求示例

该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"
    ]
  }
}
```

## 获取节点Fips驱动器报告

你可以使用 `GetNodeFipsDrivesReport` 检查存储集群中单个节点的 FIPS 140-2 驱动器加

密功能状态的方法。必须针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：

| 名称         | 描述                                                                                                                                                                                                                      | 类型     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| fipsDrives | 包含此节点对 FIPS 140-2 功能支持状态的 JSON 对象。可能值： <ul style="list-style-type: none"><li>• 无：节点不具备 FIPS 功能。</li><li>• 部分：节点具备 FIPS 功能，但节点中的并非所有驱动器都是 FIPS 驱动器。</li><li>• 就绪：节点具备 FIPS 功能，并且节点中的所有驱动器都是 FIPS 驱动器（或者没有驱动器）。</li></ul> | string |

请求示例

该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"
  }
}
```

自版本以来的新版本

11.5

## 获取节点**SSL**证书

你可以使用 `GetNodeSSLCertificate` 获取管理节点上当前处于活动状态的 SSL 证书的方法。

### 参数



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：

| 名称 | 描述              | 类型      |
|----|-----------------|---------|
| 证书 | 证书的完整 PEM 编码文本。 | string  |
| 细节 | 证书的解码信息。        | JSON 对象 |

### 请求示例

该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\nVQZGEw
JVUzELMAkGA1UECBMCTlYxFTATBgNVBACUUDFZlZ2FzLCBCYXWJ5ITEhMB8G\nA1UEChMYV2hhdc
BIYXBWZW5zIGluIFZlZ2FzLi4uMS0wKwYJKoZIhvcNAQkBFh53\naGF0aGFwcGVuc0B2ZWdhc3
N0YXlzaW4udmVnYXMwHhcNMTcwMzA4MjI1MDI2WhcN\nnMjcwMzA2MjI1MDI2WjCBGzELMAkGA1
UEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYD\nVQZHFAXWZWhcywgQmFieSExITAFBgNVBAoTGF
doYXQgSGFwcGVucyBpbWZWhd\nncy4uLjEtMCSGCSqGSIb3DQEJARYed2hhdGhhcHB1bnNAdm
VnYXNzdGF5c2luLnZl\nZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAA8U+28f
nLKQNWEMMR\nn6akeDKuehSpS79odLGigI18q1CV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FH
P4\nt6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nnmrgp
GZQF3iOSICBhtLKE5186JVT6j5dg6yjUGQO352ylc9HXHcn6lb/jy10DmVNU\nnZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\nn7dg9cyXu4afXcgKy14Ji
NBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzxF+AC\nnq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBrvBRPno5S34zGRhrnDJyTsdnEbTCB\nnuAYDVR0jBIGwMIGtgBRvBRPno5S34zGRhrn
DJyTsdnEbaGBiasBhjCBGzELMAkG\nA1UEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYDVQZHFAXW
ZWhcywgQmFieSExITAF\nnBgNVBAoTGFdoYXQgSGFwcGVucyBpbWZWhdncy4uLjEtMCSGCSqG
SIb3DQEJARYe\nnd2hhdGhhcHB1bnNAdmVnYXNzdGF5c2luLnZlZ2FzgkAzBsiFZjjf/MwDAYD
VR0T\nnBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s7lmQPECwVLfiE/ndtIbnpe\nnMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7Jyfs3714rRolFpX6N\nniebEgAae5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dH0lQ+WYnoO/eIMy0coT26JB15H\nnDEwvdl+DwkxnS1cx1v
ERv51glgua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nnHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\nn7ASOjG+DzUxzwaUI6Jzvs3Xq5Jx8ZAJJDg
l0QoQDWNDoTerBs80nwiouA==\nn-----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"
  }
}

```

# 获取节点支持的TLS密码

您可以使用 `GetNodeSupportedTlsCiphers` 在单个节点上执行此方法，以获取当前该节点支持的 TLS 密码列表。您可以将此方法用于管理节点和存储节点。

## 参数

此方法没有输入参数。

## 返回值

此方法具有以下返回值：

| 名称      | 描述                                                                          | 类型     |
|---------|-----------------------------------------------------------------------------|--------|
| 强制密码    | 节点必须使用的TLS密码套件列表。这些密码始终在节点上处于活动状态。                                          | string |
| 默认补充密码  | 节点的默认补充 TLS 密码套件列表。运行 ResetNodeSupplementalTlsCiphers API 方法时，补充密码将恢复到此列表中。 | string |
| 支持的补充密码 | 您可以使用 SetNodeSupplementalTlsCiphers API 方法配置可用的补充 TLS 密码套件列表。               | string |

## 请求示例

该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"
    ]
  }
}

```

## 获取补丁信息

你可以使用 `GetPatchInfo` 获取有关存储节点上已安装的 Element 软件补丁的信息的方法。

### 参数

该方法有以下输入参数：

| 名称    | 描述                                                                                                                                            | 类型  | 默认值   | 必填项 |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----|-------|-----|
| force | <p>强制该方法在存储集群中的所有节点上运行。只有当向集群 IP 地址而不是单个节点发出 API 请求时才需要这样做。可能值：</p> <ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul> | 布尔值 | false | 否   |

## 返回值

此方法具有以下返回值：

| 名称 | 描述                    | 类型      |
|----|-----------------------|---------|
| 补丁 | 包含有关此节点上已安装补丁程序信息的对象。 | JSON 对象 |

## 请求示例

该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"
        ]
      }
    }
  }
}
```

自版本以来的新版本

12.3

## 获取待处理操作

你可以使用 `GetPendingOperation` 检测节点上当前正在进行的操作的方法。该方法还可以用来报告操作何时完成。

### 参数

此方法没有输入参数。

### 返回值

此方法具有以下返回值：



| 名称        | 描述                                                                                                    | 类型     |
|-----------|-------------------------------------------------------------------------------------------------------|--------|
| pending   | 可能值： <ul style="list-style-type: none"> <li>• 确实：手术仍在进行中。</li> <li>• 错误的： <p>手术已停止进行。</p> </li> </ul> | 布尔值    |
| operation | 正在进行或已完成的操作名称。                                                                                        | string |

## 请求示例

该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",
    }
  }
}
```

## 自版本以来的新版本

9.6

## 获取SSH信息

你可以使用 `GetSshInfo` 查询单个节点上 SSH 服务状态的方法。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称 | 描述             | 类型  |
|----|----------------|-----|
| 结果 | 此节点的 SSH 服务状态。 | 布尔值 |

### 请求示例

该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
  }
}
```

## 列表驱动器硬件

你可以使用 `ListDriveHardware` 列出连接到节点的所有驱动器的方法。当用于单个节点时，此方法返回驱动器硬件信息。在集群主节点 MVIP 上使用时，此方法返回所有节点上所有驱动器的信息。

### 参数



方法响应中的“securitySupported”: true 行并不意味着驱动器能够加密；仅表示可以查询安全状态。如果节点类型型号以“-NE”结尾，则启用这些驱动器的安全功能的命令将失败。

此方法具有以下参数：

| 名称 | 描述                     | 类型  | 默认值 | 必填项 |
|----|------------------------|-----|-----|-----|
| 力量 | 设置为 true 可在所有节点上运行此方法。 | 布尔值 | 无   | 否   |

返回值

该方法返回以下值：

| 名称   | 描述            | 类型        |
|------|---------------|-----------|
| 驱动硬件 | 返回节点的驱动器硬件信息。 | JSON 对象数组 |

请求示例

该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"
      }
    ]
  }
}

```

自版本以来的新版本

9.6

查找更多信息

[启用静态加密](#)

## 列出网络接口

你可以使用 `ListNetworkInterfaces` 列出节点上每个网络接口信息的方法。此 API 方法旨在用于单个节点；访问单个节点需要用户 ID 和密码身份验证。但是，如果在方法调用中将参数 `force` 的值设为 `true`，则可以在集群上使用此方法。在集群上使用该参数时，将列出所有接口。

### 参数

该方法具有以下输入参数：

| 名称 | 描述                                                                                                                             | 类型  | 默认值 | 必填项 |
|----|--------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| 力量 | 可能值： <ul style="list-style-type: none"><li>• <code>true</code>：返回集群中所有网络接口的信息。</li><li>• <code>false</code>：未返回任何信息。</li></ul> | 布尔值 | 无   | 否   |

### 返回值

该方法返回以下值：

| 名称                      | 描述                                                                        | 类型                      |
|-------------------------|---------------------------------------------------------------------------|-------------------------|
| <code>interfaces</code> | 存储节点（或整个存储集群，如果 <code>force =</code> ）的每个网络接口的配置信息列表 <code>true</code> ）。 | <a href="#">网络接口</a> 大批 |

### 请求示例

该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,
```



| 名称       | 描述                                    | 类型                         |
|----------|---------------------------------------|----------------------------|
| 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，例如丢包数和各种类型的网络错误。 | <a href="#">网络接口统计信息大批</a> |

## 请求示例

该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
    }
  ]
}
```



自版本以来的新版本

12.3

## 列表测试

你可以使用 `ListTests` 列出可在节点上运行的测试的方法。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称 | 描述           | 类型    |
|----|--------------|-------|
| 测试 | 可对节点执行的测试列表。 | 字符串数组 |

### 请求示例

该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"
    ]
  }
}
```

自版本以来的新版本

9.6

## 列表实用程序

你可以使用 `ListUtilities` 列出可在节点上运行的操作的方法。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称   | 描述                | 类型    |
|------|-------------------|-------|
| 公用事业 | 当前可在节点上运行的实用程序列表。 | 字符串数组 |

### 请求示例

该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"
    ]
  }
}
```

## 自版本以来的新版本

9.6

## 移除节点**SSL**证书

你可以使用 `RemoveNodeSSLCertificate` 删除管理节点的用户 SSL 证书和私钥的方法。删除证书和私钥后，管理节点将配置为使用默认证书和私钥。

### 参数



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

该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" : {}
}
```

# 重置驱动器

你可以使用 `ResetDrives` 主动初始化驱动器并删除驱动器上当前所有数据的方法。然后，该驱动器可以在现有节点中重复使用，也可以在升级后的节点中使用。

## 参数

该方法有以下输入参数：

| 名称 | 描述                    | 类型     | 默认值 | 必填项 |
|----|-----------------------|--------|-----|-----|
| 驱动 | 要重置的设备名称列表（不是驱动器 ID）。 | string | 无   | 是   |
| 力量 | 设置为 true 可重置驱动器。      | 布尔值    | 无   | 是   |

## 返回值

该方法返回以下值：

| 名称 | 描述             | 类型        |
|----|----------------|-----------|
| 细节 | 正在重置的驱动器的详细信息。 | JSON 对象数组 |

## 请求示例

该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"
  }
}
```

自版本以来的新版本

9.6

## 重置节点

你可以使用 `ResetNode` 将节点重置为出厂设置的方法。调用此方法时，节点中的所有数据、软件包（软件升级等）、配置和日志文件都将被删除。但是，在此操作过程中，节点的网络设置将被保留。加入集群的节点不能重置为出厂设置。

### 参数

ResetNode API 只能用于处于“可用”状态的节点。它不能用于集群中处于“活动”状态或“待定”状态的节点。

小心：

此方法会清除节点上的所有客户数据。

该方法有以下输入参数：

| 名称      | 描述                                    | 类型      | 默认值 | 必填项 |
|---------|---------------------------------------|---------|-----|-----|
| 建造      | 用于指定要将节点重置为的远程 Element 软件映像的 URL。     | URL     | 无   | 否   |
| 力量      | 设置为 true 可重置节点。                       | 布尔值     | 无   | 是   |
| options | 用于输入运行复位操作的规范。如有需要，NetApp技术支持将提供详细信息。 | JSON 对象 | 无   | 否   |

### 返回值

此方法没有返回值。

### 请求示例

该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"
}
}
}

```

自版本以来的新版本

9.6

## 重置节点补充TLS密码

您可以使用 `ResetNodeSupplementalTlsCiphers` 将补充 TLS 密码套件列表恢复为默认值的方法。您可以在管理节点上使用此命令。

参数



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

该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" : {}
}
```

## 重启网络

你可以使用 `RestartNetworking` 重启节点上的网络服务的方法。

小心：

此方法会重启节点上的所有网络服务，导致网络连接暂时中断。

## 参数

该方法具有以下输入参数：

| 名称 | 描述                    | 类型  | 默认值 | 必填项 |
|----|-----------------------|-----|-----|-----|
| 力量 | 设置为 true 可重启节点上的网络服务。 | 布尔值 | 无   | 是   |

## 返回值

此方法没有返回值。

## 请求示例

该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" : {}
}
```

## 自版本以来的新版本

9.6

## 重启服务

你可以使用 `RestartServices` 重启节点上服务的方法。

### 参数

小心：

此方法会导致节点服务暂时中断。

该方法有以下输入参数：

| 名称      | 描述                  | 类型     | 默认值 | 必填项 |
|---------|---------------------|--------|-----|-----|
| 力量      | 设置为 true 可重启节点上的服务。 | 布尔值    | 无   | 是   |
| service | 要重启的服务名称。           | string | 无   | 否   |
| 行动      | 对服务执行的操作（启动、停止、重启）。 | string | 无   | 否   |

## 返回值

此方法具有以下返回值：

| 名称 | 描述                  | 类型      |
|----|---------------------|---------|
| 细节 | 服务重启过程的输出，包括错误（如有）。 | JSON 对象 |
| 期间 | 重启节点服务所需的时间（以秒为单位）。 | string  |
| 结果 | 重启后的结果。             | string  |

## 请求示例

该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"
  }
}
```

## 自版本以来的新版本

9.6

# 设置集群配置

你可以使用 `SetClusterConfig` 用于设置节点与其所属集群通信所用配置的方法。要显示节点的当前集群接口设置，请运行以下命令 `GetClusterConfig` API 方法。

## 参数

该方法具有以下输入参数：

| 名称 | 描述                                            | 类型 | 默认值 | 必填项 |
|----|-----------------------------------------------|----|-----|-----|
| 集群 | 此方法调用期间需要更改的配置属性。只需将您想要更改的字段作为成员添加到此方法的参数中即可。 | 集群 | 无   | 否   |

## 返回值

该方法返回以下值：

| 名称 | 描述              | 类型 |
|----|-----------------|----|
| 集群 | 节点用于与集群通信的配置信息。 | 集群 |

## 请求示例

该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"
    }
  }
}
```

自版本以来的新版本

9.6

## 设置配置

你可以使用 `SetConfig` 设置节点网络和集群信息的方法。此方法在一个 API 方法中包含了相同的设置，这两种方法均可用。`SetClusterConfig` 和 `SetNetworkConfig` 方法。此方法只需包含您想要更改的字段即可。

### 参数

小心：

更改节点上的绑定模式可能会导致网络连接暂时中断。

该方法有以下输入参数：

| 名称      | 描述                            | 类型       | 默认值 | 必填项 |
|---------|-------------------------------|----------|-----|-----|
| 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。 | 集群       | 无   | 否   |
| network | 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

该方法返回以下值：

| 名称     | 描述                                                                                                                                                                                        | 类型      |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| config | <p>节点的新配置和当前配置。此对象包含：</p> <ul style="list-style-type: none"> <li>• <a href="#">集群</a>：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。</li> <li>• <a href="#">网络（所有接口）</a>：节点的每个网络接口的网络连接类型和当前设置。</li> </ul> | JSON 对象 |

## 请求示例

该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 方法的返回值相同。使用 SetConfig 时，可以看到对象显示的所有字段和更新后的值。

## 自版本以来的新版本

9.6

## 查找更多信息

- [设置集群配置](#)
- [设置网络配置](#)
- [GetConfig](#)

## 设置网络配置

你可以使用 `SetNetworkConfig` 设置节点网络配置的方法。要显示节点的当前网络设置，请运行以下命令 `GetNetworkConfig` API 方法。

## 参数

小心：

更改节点上的绑定模式可能会导致网络连接暂时中断。

该方法具有以下输入参数：

| 名称      | 描述                                          | 类型                       | 默认值 | 必填项 |
|---------|---------------------------------------------|--------------------------|-----|-----|
| network | 包含要修改的节点网络设置的对象。您只需将要更改的字段作为属性添加到此方法的参数中即可。 | <a href="#">网络（所有接口）</a> | 无   | 否   |

## 返回值

该方法返回以下值：

| 名称      | 描述         | 类型                       |
|---------|------------|--------------------------|
| network | 节点的最新网络配置。 | <a href="#">网络（所有接口）</a> |

## 请求示例

该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。该方法显示每个对象的所有成员，并包含任何已更改成员的新值。

## 自版本以来的新版本

9.6

## 查找更多信息

- [获取网络配置](#)
- [获取网络配置](#)

## 设置节点SSL证书

你可以使用 `SetNodeSSLCertificate` 为管理节点设置用户 SSL 证书和私钥的方法。



使用 API 后，必须重启管理节点。

## 参数



必须对管理节点调用此方法。例如：



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

该方法有以下输入参数：

| 名称 | 描述                                                                                                                            | 类型     | 默认值 | 必填项 |
|----|-------------------------------------------------------------------------------------------------------------------------------|--------|-----|-----|
| 证书 | 证书的 PEM 编码文本版本。注意：设置节点或集群证书时，证书必须包含 serverAuth 的 extendedKeyUsage 扩展。此扩展程序可使证书在常见的操作系统和浏览器上无错误地使用。如果扩展程序不存在，API 将拒绝该证书，认为其无效。 | string | 无   | 是   |
| 私钥 | 私钥的 PEM 编码文本版本。                                                                                                               | string | 无   | 是   |

## 返回值

此方法没有返回值。

## 请求示例

该方法的请求类似于以下示例：

```
{
  "method" : "SetNodeSSLCertificate",
  "params" : {
    "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEEowIBAAKCAQEA8U+28fnLKQNWEMMR6akeDKuehSpS79odLGigI18qlCV/AUY5\nzLjqsT
jBvTJVrv44yoCTgNrx36U7FHP4t6P/Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP
20HRYpZvY+Uz7LYEFCmrpgGZQF3iOSIcBHtLKE5186JVT6j5dg\n6yjUGQO352ylc9HXHcn6lb
/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\njaCJC1CWau3sEHCckLOhWeY4HaPS
oWq0GKLMkKdChB4nWUYg3gSWQkCgYEA9zuN\nHW8GPS+yjixeKXmkK00x/vvxzR+J5HH5znaI
Hss48THyhzXpLr+v30Hy2h0yAlBS\nny5Ja6wsomb0mVe4NxVtVawg2E9vVvTa1UC+TNmFBBuL
RPfjcnjDerrSuQ5lYY+M\nC9MJtXGfhp//G0bzwsRzZxOBsUJb15tpaZIs9MCgYAJricpkKjM
0x1Z1jdVXsos\nPilnbho4qLngrzuUuxKXEPEnzBxUOqCpwQgdzZLYYw788TCVVIVXLEYem2s0
7dDA\nDTo+WrzQNkvC6IgqtXH1RgqegIoG1VbgQsbsYmDhdaQ+os4+AoeQXw3vgAhJ/qNJ\nnjQ
4Ttw3ylt7FYkRH26ACWQKBgQC74Zmf4JuRLAo5WSZFxpcmMvtnlvdutqUH4kXA\nnzPssy6t+QE
La1fFbAXkZ5PglITK752aiaX6KQNG6qRsA3VS1J6drD9/2AofOQU17\n+n+jOkGzmmoXf49Zj3iS
akwg0ZbQNGXNxEsCAUr0BYAobPp9/fB4PbtUs99fvtocFr\nnjS562QKBgCb+JMDP5q7jpUuspj
0obd/ZS+MsomE+gFAMBJ71KFQ7KuoNezNFO+ZE\nn3rnR8AqAm4VMzqRaHS2PWNe2H14J4hKu96
qNPnHbsW1NjXdAL9P7oqQIrhGLVdhX\nInDXvTgXmDMoet4BKnfteLrXFKHgGqXJoczq4JWzGS
IHNgvkrH60\n-----END RSA PRIVATE KEY-----\n",
    "certificate": "-----BEGIN CERTIFICATE-----
\nMIIEEdzCCA1+gAwIBAgIJAMwbIhWY43/zMA0GCSqGSIb3DQEBAQUAMIGDMQswCQYD\nJVUzELMAkGA1UECBMCTlYxFTATBgNVBACUUDFZlZ2FzLCBCYXWJ5ITEhMB8G\nBIYXBWZW5zIGluIFZlZ2FzLi4uMS0wKWyJKoZIhvcNAQkBFh53\nN0YXlzaW4udmVnYXMwHhcNMTCwMzA2MjI1MDI2WhcN\nUEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYD\nVnYXNzdGF5c2luLnZl\nnZ2FzMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAE8U+28f
nLKQNWEMR\nn6akeDKuehSpS79odLGigI18qlCV/AUY5ZLjqsTjBvTJVRv44yoCTgNrx36U7FH
P4\nt6P/Si0aYr4ovx15wDpEM3Qyy5JPB7JelOB6AD7fmiTweP20HRYpZvY+Uz7LYEFC\nnmrgp
GZQF3iOSiCBHtLKE5186JVT6j5dg6yjUGQ0352ylc9HXHcn6lb/jy10DmVNU\nnZ0caQwAmIS3J
moyx+zj/Ya4WKq+2SqTAX7bX0F3wHHfXnZlHnM8fET5N/9A+K6lS\nn7dg9cyXu4afXcgKy14Ji
NBvqbBjhgJtE76yAy6rTHu0xM3jjdkcb9Y8miNzx+AC\nnq+itawIDAQABo4HrMIHoMB0GA1Ud
DgQWBBrvBRPno5S34zGRhrnDJyTsdnEbTCB\nnuAYDVR0jBIGwMIGtgBRvBRPno5S34zGRhrn
DJyTsdnEbaGBiasBhjCBGzELMAkG\nnA1UEBhMCVVMxCzAJBgNVBAGTAk5WMRUwEwYD\nVnYXNzdGF5c2luLnZl\nnZ2FzggkAzBsiFZjjf/MwDAYD
VR0T\nnBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAhVND5s71mQPECwVLfiE/ndtIbnpe\nnMq
o5geQHCHnNlu5RV9j8aYHp9kW2qCDJ5vueZtZ2L1tC4D7Jyfs3714rRolFpX6N\nniebEgAae5e
WvB6zgiAcMRIKqu3DmJ7y3CFGk9dHolQ+WYnoO/eIMy0coT26JB15H\nnDEwvdl+DwkxnS1cx1v
ERv51g1gua6AE3tBrlov8q1G4zMJboo3YEwMFwxLkxAFXR\nnHgMoPDym099kvc84B1k7HkDGHp
r4tLfVelDJy2zCWIQ5ddbVpyPW2xuE4p4BGx2B\nn7ASojG+DzUxzwaUI6Jzvs3Xq5Jx8ZAjJDg
l0QoQDWNDoTerBs80nwioA==\n-----END CERTIFICATE-----\n"
  },
  "id" : 2
}

```

## 响应示例

此方法返回类似于以下示例的响应：

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

## 设置节点补充TLS密码

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

### 参数



必须对管理节点调用此方法。例如：

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

该方法具有以下输入参数：

| 名称   | 描述                                    | 类型     | 默认值 | 必填项 |
|------|---------------------------------------|--------|-----|-----|
| 补充密码 | 补充密码套件名称采用 OpenSSL 命名方案。密码套件名称不区分大小写。 | string | 无   | 是   |

### 返回值

此方法具有以下返回值：

| 名称   | 描述                                 | 类型     |
|------|------------------------------------|--------|
| 强制密码 | 节点必须使用的TLS密码套件列表。这些密码始终在节点上处于活动状态。 | string |
| 补充密码 | 节点的补充 TLS 密码套件列表。                  | string |

### 请求示例

该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` 重启或关闭集群中节点的方法。您可以使用此方法关闭集群中的单个节点、多个节点或所有节点。

## 参数

该方法有以下输入参数：

| 名称     | 描述                                                                                                  | 类型     | 默认值 | 必填项 |
|--------|-----------------------------------------------------------------------------------------------------|--------|-----|-----|
| 节点     | 要重启或关闭的节点的 NodeID 列表。                                                                               | 整数数组   | 无   | 是   |
| option | 针对该集群应采取的行动。可能值： <ul style="list-style-type: none"> <li>• 重启：重启集群。</li> <li>• 停止：执行完全断电。</li> </ul> | string | 重启  | 否   |

## 返回值

此方法没有返回值。

## 请求示例

该方法的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
    ]
  }
}
```

自版本以来的新版本

9.6

## TestConnectEnsemble

你可以使用 `TestConnectEnsemble` 验证与指定数据库集合的连接性的方法。默认情况下，它使用节点所属集群的集成模型。或者，您可以提供不同的组件来测试连接性。

### 参数

该方法具有以下输入参数：

| 名称 | 描述                           | 类型     | 默认值 | 必填项 |
|----|------------------------------|--------|-----|-----|
| 合奏 | 用于连接性测试的集群节点 IP 地址列表（以逗号分隔）。 | string | 无   | 否   |

### 返回值

该方法返回以下值：

| 名称 | 描述                                                                                                                                                                                                    | 类型      |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 细节 | <p>返回的对象：</p> <ul style="list-style-type: none"><li>• <code>nodes</code>：（对象）测试中每个集成节点的列表以及测试结果。</li><li>• <code>duration</code>（字符串）运行测试所需的时间。</li><li>• <code>result</code>：（字符串）整个测试的结果。</li></ul> | JSON 对象 |

## 请求示例

该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"
  }
}
```

自版本以来的新版本

9.6

## TestConnectMvip

你可以使用 `TestConnectMvip` 测试与存储集群的管理连接的方法。该测试会向 MVIP 发送 ping 请求，并执行一个简单的 API 方法来验证连接性。

### 参数

该方法具有以下输入参数：

| 名称 | 描述                                           | 类型     | 默认值 | 必填项 |
|----|----------------------------------------------|--------|-----|-----|
| 贵宾 | 您可以传递此值来测试不同 MVIP 的管理连接。测试与目标集群的连接时，不需要使用此值。 | string | 无   | 否   |

## 返回值

该方法返回以下值：



| 名称 | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 类型      |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| 细节 | <p>测试操作信息（JSON 对象）：</p> <ul style="list-style-type: none"> <li>• <code>connected</code>: 指示测试是否可以连接到MVIP（布尔值）</li> <li>• <code>mvip</code>: 针对（字符串）测试的MVIP</li> <li>• <code>pingBytes</code>: 56 字节和 1500 字节（对象）的 ping 测试详情 <ul style="list-style-type: none"> <li>◦ <code>`56`56</code> 字节 ping 测试结果（JSON 对象）： <ul style="list-style-type: none"> <li>▪ <code>individualResponseTimes</code>: 每个集成节点的响应时间列表（字符串数组）</li> <li>▪ <code>individualStatus</code>: 来自每个集群节点的 ping 状态列表（布尔数组）</li> <li>▪ <code>`responseTime`</code> 平均 ping 响应时间（字符串）</li> <li>▪ <code>successful</code>: 指示 ping 测试是否成功（布尔值）</li> </ul> </li> <li>◦ <code>`1500`1500</code> 字节 ping 测试结果（JSON 对象）： <ul style="list-style-type: none"> <li>▪ <code>individualResponseTimes</code>: 每个集成节点的响应时间列表（字符串数组）</li> <li>▪ <code>individualStatus</code>: 来自每个集群节点的 ping 状态列表（布尔数组）</li> <li>▪ <code>`responseTime`</code> 平均 ping 响应时间（字符串）</li> <li>▪ <code>`successful`</code> ping 测试是否成功（布尔值）</li> </ul> </li> <li>◦ <code>duration</code>: 运行测试所需的时间（字符串）</li> <li>◦ <code>`result`</code> 测试的整体结果（字符串）</li> </ul> </li> </ul> | JSON 对象 |

## 请求示例

该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"
}
}

```

自版本以来的新版本

9.6

## TestConnectSvip

您可以使用 `TestConnectSvip` 测试存储与存储集群连接的方法。该测试使用 ICMP 数据包 ping SVIP，如果成功，则作为 iSCSI 发起程序进行连接。

### 参数

该方法具有以下输入参数：

| 名称   | 描述                                           | 类型     | 默认值 | 必填项 |
|------|----------------------------------------------|--------|-----|-----|
| svip | 您可以传递此值来测试不同 SVIP 的管理连接。测试与目标集群的连接时，不需要使用此值。 | string | 无   | 否   |

### 返回值

该方法返回以下值：



| 名称 | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 类型     |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 细节 | <p>测试操作信息（JSON 对象）：</p> <ul style="list-style-type: none"> <li>• connected: 指示测试是否可以连接到 SVIP（布尔值）</li> <li>• svip: 针对（字符串）测试的 SVIP</li> <li>• pingBytes: 56 字节和 9000 字节（对象）的 ping 测试详情 <ul style="list-style-type: none"> <li>◦ `56`56 字节 ping 测试结果（JSON 对象）： <ul style="list-style-type: none"> <li>▪ individualResponseTimes: 每个集成节点的响应时间列表（字符串数组）</li> <li>▪ individualStatus: 来自每个集群节点的 ping 状态列表（布尔数组）</li> <li>▪ `responseTime` 平均 ping 响应时间（字符串）</li> <li>▪ successful: 指示 ping 测试是否成功（布尔值）</li> </ul> </li> <li>◦ `9000`9000 字节 ping 测试结果（JSON 对象）： <ul style="list-style-type: none"> <li>▪ individualResponseTimes: 每个集成节点的响应时间列表（字符串数组）</li> <li>▪ individualStatus: 来自每个集群节点的 ping 状态列表（布尔数组）</li> <li>▪ `responseTime` 平均 ping 响应时间（字符串）</li> <li>▪ successful: 指示 ping 测试是否成功（布尔值）</li> </ul> </li> <li>◦ duration: 运行测试所需的时间（字符串）</li> <li>◦ `result` 测试的整体结果（字符串）</li> </ul> </li> </ul> | string |

## 请求示例

该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"
}
}

```

自版本以来的新版本

9.6

## 试驾

你可以使用 `TestDrives` 对节点上所有驱动器运行硬件验证的方法。该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。

### 参数

你只能使用 `TestDrives` 对集群中非“活动”节点执行此方法。



这项测试大约需要10分钟。

该方法有以下输入参数：

| 名称 | 描述                   | 类型  | 默认值 | 必填项 |
|----|----------------------|-----|-----|-----|
| 力量 | 设置为 true 以测试节点上的驱动器。 | 布尔值 | 无   | 是   |
| 分钟 | 指定测试运行的分钟数。          | 整数  | 10  | 否   |

## 返回值

该方法返回以下值：

| 名称 | 描述              | 类型      |
|----|-----------------|---------|
| 细节 | 有关测试操作成功或失败的信息。 | JSON 对象 |

## 请求示例

该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

此方法返回一个表格，其中包含节点中每个驱动器的测试结果。

## 自版本以来的新版本

9.6

## 测试硬件配置

你可以使用 `TestHardwareConfig` 对节点执行硬件测试的方法。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在。

## 参数



这些测试并非旨在检测硬件故障。

该方法有以下输入参数：



| 名称    | 描述                                                                                                                                  | 类型  | 默认值   | 必填项 |
|-------|-------------------------------------------------------------------------------------------------------------------------------------|-----|-------|-----|
| clean | 启动硬件配置测试，并清除缓存。可能值：<br><br><ul style="list-style-type: none"> <li>• true：删除缓存的测试结果文件并重新运行测试。</li> <li>• false：检索缓存的测试结果。</li> </ul> | 布尔值 | false | 否   |
| 力量    | 要成功重置节点，此方法中必须包含力参数。                                                                                                                | 布尔值 | 无     | 是   |

## 返回值

该方法返回以下值：

| 名称 | 描述      | 类型      |
|----|---------|---------|
| 细节 | 硬件配置详情。 | JSON 对象 |

## 请求示例

该方法的请求类似于以下示例：

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

## 响应示例

由于此回复示例篇幅较长，因此已在补充主题中进行了记录。

## 自版本以来的新版本

9.6

查找更多信息

[测试硬件配置](#)

## 测试定位集群

你可以使用 `TestLocateCluster` 验证节点是否能够找到集群配置中指定的集群的方法。输出结果验证集群已创建，并列出集群集合中的节点。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称 | 描述              | 类型      |
|----|-----------------|---------|
| 细节 | 有关测试操作成功或失败的信息。 | JSON 对象 |

### 请求示例

该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"
  }
}
```

自版本以来的新版本

9.6

## 测试本地连接

你可以使用 `TestLocalConnectivity` 用于 ping 活动集群中每个节点的集群 IP (CIP) 的方法。

### 参数

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称 | 描述                        | 类型      |
|----|---------------------------|---------|
| 细节 | 本地活动集群中每个节点的单独 ping 响应时间。 | JSON 对象 |

## 请求示例

该方法的请求类似于以下示例：

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

## 响应示例

该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"
}
}
```

自版本以来的新版本

9.6

## 测试网络配置

您可以使用 `TestNetworkConfig` 测试配置的网络设置是否与系统上使用的网络设置相匹配的方法。

### 参数

当您在 UI 或 TUI 中使用 SetNetworkConfig 方法配置节点时，配置将被验证并存储。TestNetworkConfig API 测试使用存储的配置进行验证后逻辑。例如，在发生断电或网络故障时，您可以使用此 API 方法确保节点以最新存储的网络配置运行。这验证了配置中没有错误，并且当前配置正在使用中。

此测试旨在仅显示响应输出中的错误。如果没有错误，此测试不会返回任何输出。请参阅以下回复示例。

此方法没有输入参数。

### 返回值

该方法返回以下值：

| 名称 | 描述                                | 类型      |
|----|-----------------------------------|---------|
| 细节 | 包含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

该方法的请求类似于以下示例：

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

## 响应示例 1

如果没有检测到错误，则不返回任何响应。

```
{
  "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"
  }
}
```

### 自版本以来的新版本

9.6

### 查找更多信息

[设置网络配置](#)

## 测试Ping

你可以使用 `TestPing` 使用 ICMP 数据包测试集群中所有节点在 1G 和 10G 接口上的网络连接的方法。该测试根据网络配置中的 MTU 设置，为每个数据包使用合适的 MTU 大小。`TestPing` 不会创建临时 VLAN 接口。

### 参数

该方法具有以下输入参数：

| 名称 | 描述                    | 类型 | 默认值 | 必填项 |
|----|-----------------------|----|-----|-----|
| 尝试 | 指定系统应重复执行测试 ping 的次数。 | 整数 | 5   | 否   |



| 名称              | 描述                                                                                                                                                             | 类型     | 默认值   | 必填项 |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------|-----|
| hosts           | 指定要 ping 的设备地址或主机名的逗号分隔列表。如果没有指定主机，则该方法会 ping 存储集群中的主机。                                                                                                        | string | 无     | 否   |
| 界面              | <p>现有（基础）接口，用于发送 ping 请求。可能值：</p> <ul style="list-style-type: none"> <li>• Bond10G：从 Bond10G 接口发送 ping 请求。</li> <li>• Bond1G：从 Bond1G 接口发送 ping 请求。</li> </ul> | string | 无     | 否   |
| 数据包大小           | 指定发送到每个 IP 地址的 ICMP 数据包中要发送的字节数。字节数必须小于网络配置中指定的最大 MTU。                                                                                                         | 整数     | 无     | 否   |
| ping超时毫秒        | 指定等待每个 ping 响应的毫秒数。                                                                                                                                            | 整数     | 500毫秒 | 否   |
| 禁止碎片化           | 启用 ICMP 数据包的 DF（不分片）标志。                                                                                                                                        | 布尔值    | false | 否   |
| sourceAddressV4 | ICMP ping 数据包中使用的源 IPv4 地址。                                                                                                                                    | string | 无     | 否   |
| sourceAddressV6 | ICMP ping 数据包中使用的源 IPv6 地址。                                                                                                                                    | string | 无     | 否   |
| 总超时时间秒          | 指定 ping 命令在发出下一次 ping 尝试或结束进程之前应等待系统响应的时间（以秒为单位）。                                                                                                              | 整数     | 5     | 否   |

| 名称     | 描述                        | 类型 | 默认值 | 必填项 |
|--------|---------------------------|----|-----|-----|
| 虚拟网络标签 | 发送 ping 数据包时要使用的 VLAN ID。 | 整数 | 无   | 否   |

## 返回值

该方法返回以下值：

| 名称 | 描述                               | 类型      |
|----|----------------------------------|---------|
| 细节 | 列出节点能够通信的每个 IP 地址以及 ping 响应统计信息。 | JSON 对象 |

## 请求示例

该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"
  }
}

```

自版本以来的新版本

5.0

## 测试远程连接

您可以使用 `TestRemoteConnectivity` ping 远程集群的每个节点并检查远程集群数据库连

接的方法。必须将聚类结果成对才能使用此方法获得有用的结果。如果远程数据库连接失败，系统响应会列出异常情况。

参数

此方法没有输入参数。

返回值

该方法返回以下值：

| 名称 | 描述                 | 类型      |
|----|--------------------|---------|
| 细节 | 每个节点的单独 ping 响应时间。 | JSON 对象 |

请求示例

该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"
}
```

自版本以来的新版本

9.6

## 版权信息

版权所有 © 2025 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。