集群 API 方法

Element Software

NetApp
November 19, 2025

This PDF was generated from https://docs.netapp.com/zh-cn/element-software-123/api/reference_element_api_addnodes.html on November 19, 2025. Always check docs.netapp.com for the latest.

目录

集群 API 方法 1
- 了解更多信息 2
- 添加节点 2
  - 参数 2
  - 返回值 3
  - 请求示例 3
  - 响应示例 3
  - 自版本以来的新增功能 4
  - 了解更多信息 4
- ClearClusterFaults 4
  - 参数 4
  - 返回值 5
  - 请求示例 5
  - 响应示例 5
  - 自版本以来的新增功能 5
- CreateClusterInterfacePreference 6
  - Parameters 6
  - 返回值 6
  - 请求示例 6
  - 响应示例 6
  - 自版本以来的新增功能 7
- DeleteClusterInterfacePreference 7
  - Parameters 7
  - 返回值 7
  - 请求示例 7
  - 响应示例 7
  - 自版本以来的新增功能 8
- EnableFeature 8
  - 参数 8
  - 返回值 8
  - 请求示例 8
  - 响应示例 9
  - 自版本以来的新增功能 9
- GetClusterCapacity 9
  - 效率计算 9
  - 整体效率比率计算 9
  - Parameters 9
  - 返回值 10
  - 请求示例 10

响应示例 10
自版本以来的新增功能 11
GetClusterFullThreshold 11
Parameters 12
返回值 12
请求示例 14
响应示例 15
自版本以来的新增功能 15
了解更多信息 15
GetClusterHardwareInfo 15
参数 15
返回值 16
请求示例 16
响应示例 16
自版本以来的新增功能 17
了解更多信息 17
GetClusterInfo 17
Parameters 17
返回值 17
请求示例 17
响应示例 17
自版本以来的新增功能 18
GetClusterInterfacePreference 18
Parameters 19
返回值 19
请求示例 19
响应示例 19
自版本以来的新增功能 20
GetClusterMasterNodeID 20
Parameters 20
返回值 20
请求示例 20
响应示例 20
自版本以来的新增功能 21
GetClusterStats 21
Parameters 21
返回值 21
请求示例 21
响应示例 21
自版本以来的新增功能 22
GetClusterVersionInfo 22
cluster version info 对象成员 22

Parameters 23
返回值 23
请求示例 24
响应示例 24
自版本以来的新增功能 26
GetFeatureStatus 26
参数 26
返回值 26
请求示例 27
响应示例 27
自版本以来的新增功能 27
GetLoginSessionInfo 28
Parameters 28
返回值 28
请求示例 28
响应示例 28
自版本以来的新增功能 29
GetNodeHardwareInfo 29
参数 29
返回值 29
请求示例 29
响应示例 30
自版本以来的新增功能 30
了解更多信息 30
GetNodeStats 30
参数 30
返回值 30
请求示例 31
响应示例 31
自版本以来的新增功能 31
ListActiveNodes 31
Parameters 32
返回值 32
请求示例 32
响应示例 32
自版本以来的新增功能 32
了解更多信息 32
ListAllNodes 32
Parameters 32
返回值 32
请求示例 33
响应示例 33

自版本以来的新增功能 34
ListClusterFaults 34
Parameters 35
返回值 35
请求示例 35
响应示例 36
自版本以来的新增功能 38
ListClusterInterfacePreferences 38
Parameters 38
返回值 38
请求示例 38
响应示例 38
自版本以来的新增功能 39
ListEvents 39
Parameters 39
返回值 40
请求示例 40
响应示例 40
自版本以来的新增功能 42
ListNodeStats 42
Parameters 42
返回值 42
请求示例 42
响应示例 42
自版本以来的新增功能 43
ListISCSISessions 43
Parameters 43
返回值 43
请求示例 44
响应示例 44
自版本以来的新增功能 45
ListServices 45
Parameters 45
返回值 46
请求示例 46
响应示例 46
自版本以来的新增功能 48
ListPendingNodes 48
IPv4 和 IPv6 管理地址 48
Parameters 48
返回值 48
请求示例 48

响应示例 49
自版本以来的新增功能 49
了解更多信息 50
ListPendingActiveNodes 50
Parameters 50
返回值 50
请求示例 50
响应示例 50
自版本以来的新增功能 51
ModifyClusterFullThreshold 51
Parameters 51
返回值 52
请求示例 55
响应示例 56
自版本以来的新增功能 56
ModifyClusterInterfacePreference 56
Parameters 56
返回值 57
请求示例 57
响应示例 57
自版本以来的新增功能 57
RemoveNodes 57
删除集群主节点 57
参数 58
返回值 58
请求示例 58
响应示例 58
自版本以来的新增功能 59
SetLoginSessionInfo 59
参数 59
返回值 59
请求示例 59
响应示例 60
自版本以来的新增功能 60
关闭 60
Parameters 60
返回值 61
响应示例 61
自版本以来的新增功能 61

集群 API 方法

您可以通过 Element 软件集群 API 方法管理存储集群以及属于存储集群的节点的配置和拓扑。

某些集群 API 方法可在属于集群一部分或已配置为加入集群的节点上运行。您可以向新集群或现有集群添加节点。已准备好添加到集群的节点处于 " 待定 " 状态，这意味着它们已配置，但尚未添加到集群。

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

- 关闭

了解更多信息

- "SolidFire 和 Element 软件文档"
- "早期版本的 NetApp SolidFire 和 Element 产品的文档"

添加节点

您可以使用 `addnodes` 方法向集群添加一个或多个新节点。

当未配置的节点首次启动时，系统会提示您配置此节点。配置节点后，它将注册为集群中的 " 待定节点 " 。运行 Element 软件的存储集群会根据集群上的版本自动创建节点映像。添加待定节点时，方法响应会包含 asyncHandle 值，您可以将该值与 `GetAsyncResult` 方法结合使用来查询自动映像过程的状态。

添加光纤通道节点的过程与向集群添加 Element iSCSI 存储节点的过程相同。光纤通道节点会使用节点 ID 在系统中注册。当它们变为可访问状态时，它们将处于 " 待定节点 " 状态。`ListAllNodes` 方法将返回 iSCSI 节点以及可添加到集群中的任何光纤通道节点的 pendingNodeID 。

将节点添加到已配置用于虚拟网络的集群时，系统需要足够数量的虚拟存储 IP 地址来为新节点分配虚拟 IP 。如果没有可用于新节点的虚拟 IP 地址，则 `addNode` 操作将失败。使用 `ModfyVirtualNetwork` 方法向虚拟网络添加更多存储 IP 地址。

添加节点后，该节点上的所有驱动器将变为可用，您可以使用 `AddDrives` 方法添加这些驱动器以增加集群的存储容量。

添加新节点后，可能需要几秒钟的时间，才能启动并将其驱动器注册为可用。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
自动安装	如果为 true ，则在添加节点时将对该节点执行恢复出厂映像（ RTFI ）。默认行为是执行 RTFI 。如果 `cEnableAutoInstall` cluster 常量为 false ，则优先于此参数。如果正在进行升级，则无论此参数的值如何， RTFI 过程都不会发生。	boolean	无	否

Name	Description	Type	默认值	Required
待定节点	要添加的节点的待定节点 ID 。您可以使用 ListPendingNodes 方法列出所有待定节点。	整型数组	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
自动安装	添加的节点是否返回到出厂映像。	boolean
nodes	将先前的 "pendingNodeID" 映射到 "nodeID" 的对象数组。在添加运行不兼容软件版本的待定节点时，此阵列包含一个 asyncHandle 值，您可以将该值与 GetAsyncResult 方法结合使用来查询自动映像过程的状态。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

了解更多信息

- 添加驱动器
- GetAsyncResult
- ListAllNodes
- ModifyVirtualNetwork

ClearClusterFaults

您可以使用 `ClearClusterFaults` 方法清除有关当前和先前检测到的故障的信息。可以清除已解决和未解决的故障。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
故障类型	确定要清除的故障类型。可能值： • current 当前检测到但尚未解决的故障。 • resolved： 以前检测到并已解决的故障。 • all： 当前故障和已解决故障。故障状态可通过故障对象的 `re`已解决 字段确定。	string	已解决	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

CreateClusterInterfacePreference

使用 `CreateClusterInterfacePreference` 方法，与运行 Element 软件的存储集群集成的系统可以在存储集群上创建和存储任意信息。此方法供内部使用。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
name	集群接口首选项的名称。	string	无	是的。
value	集群接口首选项的值。	string	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

11.0

DeleteClusterInterfacePreference

使用 `DDeleteClusterInterfacePreference` 方法，与运行 Element 软件的存储集群集成的系统可以删除现有集群接口首选项。此方法供内部使用。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
name	要删除的集群接口首选项的名称。	string	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

11.0

EnableFeature

您可以使用 `EnableFeature` 方法启用默认禁用的集群功能，例如 VVol 。

参数

此方法具有以下输入参数。

> 对于运行 Element 软件 11.x 的系统，在设置保护域监控之前或之后启用虚拟卷会导致集群保护域功能仅在节点级别起作用。

Name	Description	Type	默认值	Required
feature	启用集群功能。可能值： • `FIPS` ：为 HTTPS 通信启用 FIPS 140-2 认证加密。 • `FipsDrives` ：为存储集群启用 FIPS 140-2 驱动器支持。 • `snapMirror` ：启用 SnapMirror 复制集群功能。 • `VVol` ：启用 Element 软件 VVol 集群功能。	string	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

GetClusterCapacity

您可以使用 `GetClusterCapacity` 返回整个存储集群的高级别容量测量结果。此方法将返回可用于计算 Element Web UI 中显示的效率速率的字段。您可以在脚本中使用效率计算来返回精简配置，重复数据删除，数据压缩和整体效率的效率比率。

效率计算

使用以下公式计算精简配置，重复数据删除和数据压缩。这些公式适用于 Element 8.2 及更高版本。

- thinProvisioningFactor = （非零锁定 + 零锁定） / 非零锁定
- deDuplicationFactor = （非 ZeroBlocks + snapshotNonZeroBlocks ） / uniqueBlocks
- compressionFactor = （ uniqueBlocks * 4096 ） / （ uniqueBlocksUsedSpace * 0.93 ）

整体效率比率计算

使用以下等式通过精简配置，重复数据删除和数据压缩效率计算的结果计算整体集群效率。

- efficiencyFactor = thinProvisioningFactor * deDuplicationFactor * compressionFactor

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
集群容量	存储集群的容量测量结果。	集群容量

请求示例

此方法的请求类似于以下示例：

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


自版本以来的新增功能

9.6

GetClusterFullThreshold

您可以使用 `GetClusterFullThreshold` 方法查看为集群填充度级别设置的阶段。此方法将返回集群的所有填充度指标。

> 当集群达到块集群填充度的 " 错误 " 阶段时，随着集群接近 " 关键 " 阶段，所有卷上的最大 IOPS 都会线性降低到卷的最小 IOPS 。这有助于防止集群达到块集群填充度的关键阶段。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
块性	当前计算得出的集群块填充度级别。 • stage1Happy ： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage2Aware ： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage3Low ： 系统无法针对两个非同步节点故障提供冗余数据保护。对应于 Web UI 中的 * 警告 * 状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量比错误状态低 3% 时触发此警报）。 • stage4Critical ： 系统无法针对单节点故障提供冗余数据保护。无法创建新卷或克隆。对应于 Element UI 中的 * 错误 * 状态。 • stage5CompletelyConsumed ： 已完全使用。此集群为只读，并保持 iSCSI 连接，但所有写入操作均会暂停。对应于 Element UI 中的 * 严重 * 状态。	string
全满	反映 "blockFullness" 和 "metadataFullness" 之间的最高填充度级别。	string
maxMetadataOverProvisionFactor	一个值，表示元数据空间相对于可用空间量可以过度配置的次数。例如，如果有足够的元数据空间来存储 100 TiB 的卷，并且此数字设置为 5 ，则可以创建 500 TiB 的卷。	整型

Name	Description	Type
元数据全满	当前计算得出的集群元数据填充度级别。 • stage1Happy ： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage2Aware ： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage3Low ： 系统无法针对两个非同步节点故障提供冗余数据保护。对应于 Web UI 中的 * 警告 * 状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量比错误状态低 3% 时触发此警报）。 • stage4Critical ： 系统无法针对单节点故障提供冗余数据保护。无法创建新卷或克隆。对应于 Element UI 中的 * 错误 * 状态。 • stage5CompletelyConsumed ： 已完全使用。此集群为只读，并保持 iSCSI 连接，但所有写入操作均会暂停。对应于 Element UI 中的 * 严重 * 状态。	string
sliceReserveUsedThresholdPct	错误情况。如果预留的分区利用率大于此值，则会触发系统警报。	整型
stage2AwareThreshold	感知条件。为阶段 2 集群阈值级别设置的值。	整型
stage2BlockThresholdBytes	存在第 2 阶段条件的集群所使用的字节数。	整型
stage2MetadataThresholdBytes	存在第 2 阶段填充度条件的集群所使用的元数据字节数。	
stage3BlockThresholdBytes	存在第 3 阶段填充度条件的集群所使用的存储字节数。	整型
stage3BlockThresholdPercent	为阶段 3 设置的百分比值。达到此百分比时，警报日志中会发布一条警告。	整型

Name	Description	Type
stage3LowThreshold	错误情况。由于集群容量不足而创建系统警报的阈值。	整型
stage3MetadataThresholdBytes	存在第 3 阶段填充度条件的集群使用的元数据字节数。	整型
stage3MetadataThresholdPercent	为 stage3 设置的元数据填充度百分比值。达到此百分比时，警报日志中将发布警告。	整型
stage4BlockThresholdBytes	存在第 4 阶段填充度条件的集群所使用的存储字节数。	整型
stage4CriticalThreshold	错误情况。创建系统警报以警告集群容量严重不足的阈值。	整型
stage4MetadataThresholdBytes	存在第 4 阶段填充度条件的集群使用的元数据字节数。	整型
stage5BlockThresholdBytes	存在第 5 阶段填充度条件的集群使用的存储字节数。	整型
stage5MetadataThresholdBytes	存在第 5 阶段填充度条件的集群使用的元数据字节数。	整型
总和集群字节	集群的物理容量，以字节为单位。	整型
sumTotalMetadataClusterBytes	可用于存储元数据的总空间量。	整型
sumUsedClusterBytes	集群上使用的存储字节数。	整型
sumUsedMetadataClusterBytes	卷驱动器上用于存储元数据的空间量。	整型

请求示例

此方法的请求类似于以下示例：

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

自版本以来的新增功能

9.6

了解更多信息

ModifyClusterFullThreshold

GetClusterHardwareInfo

您可以使用 `GetClusterHardwareInfo` 方法检索集群中所有光纤通道节点， iSCSI 节点和驱动器的硬件状态和信息。这通常包括制造商，供应商，版本以及其他相关硬件标识信息。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
type	在响应中仅包含以下类型的硬件信息之一。可能值： • 驱动器： 仅在响应中列出驱动器信息。 • nodes ： 仅在响应中列出节点信息。 • all ： 在响应中同时包含驱动器和节点信息。 如果省略此参数，则假定类型为 all 。	string	全部	否

返回值

此方法具有以下返回值：

Name	Description	Type
clusterHardwareInfo	集群中所有节点和驱动器的硬件信息。此输出中的每个对象都标有给定节点的 nodeID 。	hardwareInfo

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例的长度较长，因此它会记录在一个补充主题中。

自版本以来的新增功能

9.6

了解更多信息

GetClusterHardwareInfo

GetClusterInfo

您可以使用 `GetClusterInfo` 方法返回有关集群的配置信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
ClusterInfo	集群信息	ClusterInfo

请求示例

此方法的请求类似于以下示例：

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


自版本以来的新增功能

9.6

GetClusterInterfacePreference

使用 `GetClusterInterfacePreference` 方法，与运行 Element 软件的存储集群集成的系统可以获取有关现有集群接口首选项的信息。此方法供内部使用。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
name	集群接口首选项的名称。	string	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
首选项	请求的集群接口首选项的名称和值。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

11.0

GetClusterMasterNodeID

您可以使用 `GetClusterMasterNodeID` 方法检索执行集群范围管理任务并持有存储虚拟 IP 地址（ SVIP ）和管理虚拟 IP 地址（ MVIP ）的节点的 ID 。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
节点 ID	主节点的 ID 。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetClusterMasterNodeID",
  "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": {
    "nodeID": 1
   }
}
```

自版本以来的新增功能

9.6

GetClusterStats

您可以使用 `GetClusterStats` 方法检索集群的高级别活动测量结果。返回的值是创建集群后累积的值。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
clusterStats	集群活动信息。	clusterStats

请求示例

此方法的请求类似于以下示例：

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


自版本以来的新增功能

9.6

GetClusterVersionInfo

您可以使用 `GetClusterVersionInfo` 方法检索有关集群中每个节点上运行的 Element 软件版本的信息。此方法还会返回有关当前正在升级软件的节点的信息。

cluster version info 对象成员

此方法具有以下对象成员：

Name	Description	Type
节点 ID	节点的 ID 。	整型
nodeInternalRevision	节点的内部软件版本。	string
NodeVersion	节点的软件版本。	string

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
clusterAPIX 版本	集群上的当前 API 版本。	string
clusterVersion	集群上当前运行的 Element 软件版本。	string
clusterVersionInfo	集群中节点的列表以及每个节点的版本信息。	JSON 对象数组
pendingClusterVersion	如果存在，则表示集群软件当前正在升级或还原到的版本。	string
softwareVersionInfo	升级的状态。对象成员： • 当前版本： 节点上的当前软件版本。 • nodeID ： 要从 CurrentVersion 升级到 pendingVersion 的节点的 ID 。如果没有正在进行的升级，则此字段为 0 （零）。 • packageName ： 要安装的软件包的名称。 • pendingVersion ： 要安装的软件版本。 • StartTime ： 开始安装的日期和时间，格式为 UTC+0 。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

自版本以来的新增功能

9.6

GetFeatureStatus

您可以使用 `GetFeatureStatus` 方法检索集群功能的状态。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
feature	集群功能的状态。如果未提供任何值，则系统将返回所有功能的状态。可能值： • vVol ： 检索 VVol 集群功能的状态。 • SnapMirror ： 检索 SnapMirror 复制集群功能的状态。 • FIPS ： 检索用于 HTTPS 通信的 FIPS 140-2 加密功能的状态。 • fipsDrives ： 检索 FIPS 140-2 驱动器加密功能的状态。	string	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
features	指示功能名称及其状态的功能对象数组。对象成员： • feature ： （字符串）功能的名称。 • Enabled ： （布尔值）功能是否已启用。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetFeatureStatus",
    "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": {
        "features": [
            {
                "enabled": true,
                "feature": "Vvols"
            },
            {
                "enabled": true,
                "feature": "SnapMirror"
            },
            {
                "enabled": true,
                "feature": "Fips"
            },
            {
                "enabled": true,
                "feature": "FipsDrives"
            }
        ]
    }
}
```

自版本以来的新增功能

9.6

GetLoginSessionInfo

您可以使用 `GetLoginSessionInfo` 方法返回登录身份验证会话对登录 Shell 和 TUI 均有效的时间段。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
loginSessionInfo	包含身份验证到期期限的对象。返回的可能对象： • 超时 此会话超时并到期的时间（以分钟为单位）。格式化为 H：mm： ss例如： 1： 30： 00 ， 20： 00 ， 5： 00 。无论输入超时的格式如何，系统都会删除所有前导零和冒号。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetLoginSessionInfo",
    "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" : {
    "loginSessionInfo" : {
      "timeout" : "30:00"
    }
  }
}
```

自版本以来的新增功能

9.6

GetNodeHardwareInfo

您可以使用 `GetNodeHardwareInfo` 方法返回指定节点的所有硬件信息和状态。这通常包括制造商，供应商，版本以及其他相关硬件标识信息。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
节点 ID	要请求其硬件信息的节点的 ID 。如果指定了光纤通道节点，则会返回有关光纤通道节点的信息。	整型	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
nodeHardwareInfo	指定节点 ID 的硬件信息。此输出中的每个对象都标有给定节点的 nodeID 。	hardwareInfo

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例的长度较长，因此它会记录在一个补充主题中。

自版本以来的新增功能

9.6

了解更多信息

GetNodeHardwareInfo （光纤通道节点的输出）

GetNodeHardwareInfo （ iSCSI 的输出）

GetNodeStats

您可以使用 `GetNodeStats` 方法检索单个节点的高级别活动测量结果。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
节点 ID	指定要返回其统计信息的节点的 ID 。	整型	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
NodeStats	节点活动信息。	NodeStats

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

ListActiveNodes

您可以使用 `ListActiveNodes` 方法返回集群中当前活动节点的列表。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
nodes	集群中的活动节点列表。	node 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例的长度较长，因此它会记录在一个补充主题中。

自版本以来的新增功能

9.6

了解更多信息

ListActiveNodes

ListAllNodes

您可以使用 `ListAllNodes` 方法列出集群中的活动节点和待定节点。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
nodes	描述集群中活动节点的对象列表。	node
待定 ActiveNodes	描述集群中待定活动节点的对象列表。	pendingActiveNode 数组
待定节点	描述集群中待定节点的对象列表。	pendingNode 数组

请求示例

此方法的请求类似于以下示例：

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


自版本以来的新增功能

9.6

ListClusterFaults

您可以使用 `ListClusterFaults` 方法列出有关在集群上检测到的任何故障的信息。使用此方法，您可以列出当前故障以及已解决的故障。系统每 30 秒缓存一次故障。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
最佳实践	包括由次优系统配置触发的故障。可能值： • true • false	boolean	无	否
故障类型	确定返回的故障类型。可能值： • Current ： 列出未解决的活动故障。 • resolved ： 列出先前检测到并已解决的故障。 • all ： 列出当前故障和已解决的故障。您可以在故障对象的 " `re`已解决 " 成员中查看故障状态。	string	全部	否

返回值

此方法具有以下返回值：

Name	Description	Type
故障	描述请求的集群故障的对象。	故障

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListClusterFaults",
    "params": {
      "faultTypes": "current",
      "bestPractices": 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": {
    "faults": [
      {
        "clusterFaultID": 1,
        "code": "notUsingLACPBondMode",
        "data": null,
        "date": "2016-04-26T14:57:04.275286Z",
        "details": "Bond1G interfaces found not using LACP bond mode.
Nodes not set to LACP bond mode: {1,2,4,5}",
        "driveID": 0,
        "driveIDs": [],
        "nodeHardwareFaultID": 0,
        "nodeID": 0,
        "resolved": false,
        "resolvedDate": "",
        "serviceID": 0,
        "severity": "bestPractice",
        "type": "cluster"
      },
      {
        "clusterFaultID": 9,
        "code": "disconnectedClusterPair",
        "data": null,
        "date": "2016-04-26T20:40:08.736597Z",
        "details": "One of the clusters in a pair may have become
misconfigured or disconnected.  Remove the local pairing and retry pairing
the clusters. Disconnected Cluster Pairs: []. Misconfigured Cluster Pairs:
[3]",
        "driveID": 0,
        "driveIDs": [],
        "nodeHardwareFaultID": 0,
        "nodeID": 0,
        "resolved": false,
        "resolvedDate": "",
        "serviceID": 0,
        "severity": "warning",
        "type": "cluster"
      }
    ]
  }
}
```

自版本以来的新增功能

9.6

ListClusterInterfacePreferences

使用 `ListClusterInterfacePreference` 方法，与运行 Element 软件的存储集群集成的系统可以列出存储在系统上的现有集群接口首选项。此方法供内部使用。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
首选项	当前存储在存储集群上的集群接口对象列表，每个对象都包含首选项的名称和值。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

```
{
     "method": "ListClusterInterfacePreferences",
     "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": {
        "preferences": [
            {
                "name": "prefname",
                "value": "testvalue"
            }
        ]
    }
}
```

自版本以来的新增功能

11.0

ListEvents

您可以使用 `ListEvents` 方法列出在集群上检测到的事件，按从旧到最新的顺序进行排序。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
驱动器 ID	指定仅返回具有此驱动器 ID 的事件。	整型	0	否
endEventID	标识要返回的事件 ID 范围的结尾。	整型	无限制	否
endPublicTime	指定仅返回早于此时间发布的事件。	string	0	否
endReportTime	指定仅返回早于此时间报告的事件。	string	0	否
EventType	指定要返回的事件类型。请参见 事件 可能的事件类型。	string	0	否
最大通风数	指定要返回的最大事件数。	整型	无限制	否

Name	Description	Type	默认值	Required
节点 ID	指定仅返回具有此节点 ID 的事件。	整型		
服务 ID	指定仅返回具有此服务 ID 的事件。			
startEventID	标识要返回的事件范围的起始位置。	整型	0	否
startPublishTime	指定仅返回在此时间之后发布的事件。	string	0	否
startReportTime	指定仅返回在此时间之后报告的事件。	string	0	否

返回值

此方法具有以下返回值：

Name	Description	Type
事件	事件列表。	事件 数组

请求示例

此方法的请求类似于以下示例：

```
{
   "method": "ListEvents",
   "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":{
    "events":[
```

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

"eligibleBS":[5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,24,25,26,27,28,29,30
,31,40,41,42,43,44,45,46,47,52,53,54,55,56,57,58,59,60],
                  "generation":1431550800,
                  "participatingSS":[23,35,39,51]
               },
          "driveID":0,
          "eventID":2130,
          "eventInfoType":"gcEvent",
          "message":"GCStarted",
          "nodeID":0,
          "serviceID":2,
          "severity":0,
          "timeOfPublish":"2015-05-13T21:00:02.354128Z",
          "timeOfReport":"2015-05-13T21:00:02.353894Z"
       },{
          "details":"",
          "driveID":0,
          "eventID":2129,
          "eventInfoType":"tSEvent",
          "message":"return code:2 t:41286 tt:41286 qcc:1 qd:1 qc:1 vrc:1
tt:2 ct:Write etl:524288",
          "nodeID":0,
          "serviceID":0,
          "severity":0,
          "timeOfPublish":"2015-05-13T20:45:21.586483Z",
          "timeOfReport":"2015-05-13T20:45:21.586311Z"
       }
```

```
      ]
    }
}
```

自版本以来的新增功能

9.6

ListNodeStats

您可以使用 `ListNodeStats` 方法查看存储集群中所有存储节点的高级别活动测量结果。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
NodeStats	存储节点活动信息。	NodeStats

请求示例

此方法的请求类似于以下示例：

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


自版本以来的新增功能

9.6

ListISCSISessions

您可以使用 `ListISCSISessions` 方法列出集群中卷的 iSCSI 连接信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
会话	有关每个 iSCSI 会话的信息。	会话

请求示例

此方法的请求类似于以下示例：

```
{
   "method": "ListISCSISessions",
   "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": {
    "sessions": [
      {
        "accountID": 1,
        "accountName": "account1",
        "createTime": "2017-04-11T03:39:32.030291Z",
        "driveID": 23,
        "driveIDs": [23],
        "initiator": null,
        "initiatorIP": "10.1.1.1:37138",
        "initiatorName": "iqn.2010-01.net.solidfire.eng:c",
        "initiatorPortName": "iqn.2010-
01.net.solidfire.eng:c,i,0x23d860000",
        "initiatorSessionID": 9622126592,
        "msSinceLastIscsiPDU": 243,
        "msSinceLastScsiCommand": 141535021,
        "nodeID": 3,
        "serviceID": 6,
        "sessionID": 25769804943,
        "targetIP": "10.1.1.2:3260",
        "targetName": "iqn.2010-01.com.solidfire:a7sd.3",
        "targetPortName": "iqn.2010-01.com.solidfire:a7sd.3,t,0x1",
        "virtualNetworkID": 0,
        "volumeID": 3,
        "volumeInstance": 140327214758656
      }
     ...
    ]
  }
}
```

自版本以来的新增功能

9.6

ListServices

您可以使用 `ListServices` 方法列出集群上运行的节点，驱动器，当前软件和其他服务的服务信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
服务	在驱动器和节点上运行的服务。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
   "method": "ListServices",
   "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": {
    "services": [
        {
            "drive": {
                "assignedService": 22,
                "asyncResultIDs": [],
                "attributes": {},
                "capacity": 300069052416,
                "customerSliceFileCapacity": 0,
                "driveID": 5,
                "driveStatus": "assigned",
                "driveType": "block",
                "failCount": 0,
                "nodeID": 4,
                "reservedSliceFileCapacity": 0,
                "serial": "scsi-SATA_INTEL_SSDSC2",
                "slot": 3
            },
            "drives": [
                {
                    "assignedService": 22,
                    "asyncResultIDs": [],
```

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

```
                "driveIDs": [
                    5
                ],
                "firstTimeStartup": true,
                "ipcPort": 4008,
                "iscsiPort": 0,
                "nodeID": 4,
                "serviceID": 22,
                "serviceType": "block",
                "startedDriveIDs": [],
                "status": "healthy"
            }
        }
    ]
}
```

自版本以来的新增功能

9.6

ListPendingNodes

您可以使用 `ListPendingNodes` 方法列出系统中的待定存储节点。待定节点是指正在运行并配置为加入存储集群但尚未使用 Addnodes API 方法添加的存储节点。

IPv4 和 IPv6 管理地址

请注意， `ListPendingNodes` 不会列出管理 IP 地址（ MIP ）和管理虚拟 IP 地址（ MVIP ）的地址类型不同的待定节点。例如，如果待定节点具有 IPv6 MVIP 和 IPv4 MIP ，则 `ListPendingNodes` 将不会在结果中包含此节点。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
待定节点	集群中待定节点的列表。	pendingNode 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

了解更多信息

添加节点

ListPendingActiveNodes

您可以使用 `ListPendingActiveNodes` 方法列出集群中处于 PendingActive 状态，处于待定状态和活动状态的节点。处于此状态的节点将返回到出厂映像。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
待定 ActiveNodes	详细介绍系统中所有 PendingActive 节点信息的对象列表。	pendingActiveNode 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

ModifyClusterFullThreshold

您可以使用 `ModfyClusterFullThreshold` 方法更改存储集群接近特定容量利用率时系统生成事件的级别。在系统生成警告之前，您可以使用阈值设置指示可接受的已利用块存储量。

例如，如果您希望在系统达到 "Error" 级别块存储利用率以下 3% 时收到警报，请为 stage3BlockThresholdPercent 参数输入值 "3" 。如果达到此级别，系统会向集群管理控制台中的事件日志发送警报。

Parameters

此方法具有以下输入参数：

ⓘ	必须至少选择一个参数。

Name	Description	Type	默认值	Required
maxMetadataOverProvisionFactor	一个值，表示元数据空间相对于可用空间量可以过度配置的次数。例如，如果有足够的元数据空间来存储 100 TiB 的卷，并且此数字设置为 5 ，则可以创建 500 TiB 的卷。	整型	5.	否
stage2AwareThreshold	在系统触发容量通知之前集群中剩余容量的节点数。	整型	无	否
stage3BlockThresholdPercent	块存储利用率低于 " 错误 " 阈值的百分比，此阈值会导致系统触发集群 " 警告 " 警报。	整型	无	否
stage3MetadataThresholdPercent	元数据存储利用率低于 "`Error` " 阈值的百分比，此阈值会导致系统触发集群 "`Warning` " 警报。	整型	无	否

返回值

此方法具有以下返回值：

Name	Description	Type

块性	当前计算得出的集群块填充度级别。 • stage1Happy ： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage2Aware ： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage3Low ： 系统无法针对两个非同步节点故障提供冗余数据保护。对应于 Web UI 中的 * 警告 * 状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量比错误状态低 3% 时触发此警报）。 • stage4Critical ： 系统无法针对单节点故障提供冗余数据保护。无法创建新卷或克隆。对应于 Element UI 中的 * 错误 * 状态。 • stage5CompletelyConsumed ： 已完全使用。此集群为只读，并保持 iSCSI 连接，但所有写入操作均会暂停。对应于 Element UI 中的 * 严重 * 状态。	string
全满	反映 "blockFullness" 和 "metadataFullness" 之间的最高填充度级别。	string
maxMetadataOverProvisionFactor	一个值，表示元数据空间相对于可用空间量可以过度配置的次数。例如，如果有足够的元数据空间来存储 100 TiB 的卷，并且此数字设置为 5 ，则可以创建 500 TiB 的卷。	整型

元数据全满	当前计算得出的集群元数据填充度级别。 • stage1Happy： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage2Aware： 无警报或错误情况。对应于 Web UI 中的 * 运行状况 * 。 • stage3Low： 系统无法针对两个非同步节点故障提供冗余数据保护。对应于 Web UI 中的 * 警告 * 状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量比错误状态低 3% 时触发此警报）。 • stage4Critical： 系统无法针对单节点故障提供冗余数据保护。无法创建新卷或克隆。对应于 Element UI 中的 * 错误 * 状态。 • stage5CompletelyConsumed： 已完全使用。此集群为只读，并保持 iSCSI 连接，但所有写入操作均会暂停。对应于 Element UI 中的 * 严重 * 状态。	string
sliceReserveUsedThresholdPct	错误情况。如果预留的分区利用率大于返回的 sliceReserveUsedThresholdPct 值，则会触发系统警报。	整型
stage2AwareThreshold	感知条件。为 " 第 2 阶段 " 集群阈值级别设置的值。	整型
stage2BlockThresholdBytes	存在第 2 阶段填充度条件的集群所使用的字节数。	整型
stage2MetadataThresholdBytes	存在第 2 阶段填充度条件的集群所使用的元数据字节数。	
stage3BlockThresholdBytes	存在第 3 阶段填充度条件的集群所使用的存储字节数。	整型
stage3BlockThresholdPercent	为阶段 3 设置的百分比值。达到此百分比时，警报日志中会发布一条警告。	整型

stage3LowThreshold	错误情况。由于集群容量不足而创建系统警报的阈值。	整型
stage3MetadataThresholdBytes	存在第 3 阶段填充度条件的集群使用的元数据字节数。	
stage4BlockThresholdBytes	存在第 4 阶段填充度条件的集群所使用的存储字节数。	整型
stage4CriticalThreshold	错误情况。创建系统警报以警告集群容量严重不足的阈值。	整型
stage4MetadataThresholdBytes	存在第 4 阶段填充度条件的集群使用的元数据字节数。	
stage5BlockThresholdBytes	存在第 5 阶段填充度条件的集群使用的存储字节数。	整型
stage5MetadataThresholdBytes	存在第 5 阶段填充度条件的集群使用的元数据字节数。	
总和集群字节	集群的物理容量，以字节为单位。	整型
sumTotalMetadataClusterBytes	可用于存储元数据的总空间量。	整型
sumUsedClusterBytes	集群上使用的存储字节数。	整型
sumUsedMetadataClusterBytes	卷驱动器上用于存储元数据的空间量。	整型

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

ModifyClusterInterfacePreference

使用 `ModfyClusterInterfacePreference` 方法，与运行 Element 软件的存储集群集成的系统可以更改现有集群接口首选项。此方法供内部使用。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
name	要修改的集群接口首选项的名称。	string	无	是的。

value	集群接口首选项的新值。	string	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

11.0

RemoveNodes

您可以使用 `RemoveNodes` 删除不应再加入集群的一个或多个节点。

在删除节点之前，必须使用 `RemoveDrives` 方法删除此节点包含的所有驱动器。在 `RemoveDrives` 进程完成且所有数据均已迁移出节点之前，无法删除节点。删除节点后，它会将自身注册为待定节点。您可以重新添加节点或将其关闭（关闭节点会将其从待定节点列表中删除）。

删除集群主节点

如果使用 `RemoveNodes` 删除集群主节点，则该方法可能会在返回响应之前超时。如果方法调用无法删除节点

，请重新执行方法调用。请注意，如果要删除集群主节点以及其他节点，则应使用单独的调用来删除集群主节点。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
ignoreEnsembleTholanceChange	删除节点时，忽略对集合节点容错的更改。 如果存储集群使用的数据保护方案允许发生多个节点故障，并且删除节点会降低集合的节点容错能力，则节点删除通常会失败并显示错误。您可以将此参数设置为 true ，以禁用集合容错检查，以便成功删除节点。	boolean	false	否
nodes	要删除的节点的节点 ID 列表。	整型数组	无	是的。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

SetLoginSessionInfo

您可以使用 `SetLoginSessionInfo` 方法设置会话登录身份验证的有效期。在登录期限过后，系统上没有任何活动，身份验证将过期。登录期限过后，要继续访问集群，需要新的登录凭据。

参数

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
超时	集群身份验证的有效期。格式化为 HH：mm： ss例如： 01： 30： 00 ， 00： 90： 00 和 00： 00： 5400 均可用于等于 90 分钟超时期限。最小超时值为 1 分钟。如果未提供值或将值设置为零，则登录会话没有超时值。	string	30 分钟	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

关闭

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

Parameters

此方法具有以下输入参数：

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

返回值

此方法没有返回值。=此方法的请求示例请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

自版本以来的新增功能

9.6

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。