



NetApp Element 12.0

API 参考指南

2020 年 4 月 | 215-14525_2020-04_zh-cn
ng-gpso-china-documents@netapp.com

内容

关于 Element 软件 API.....	12
请求对象成员.....	12
响应对象成员.....	12
请求端点.....	13
API 身份验证.....	13
异步方法.....	13
Attributes.....	14
通用对象.....	15
account.....	15
AuthSessionInfo.....	15
bulkVolumeJob.....	16
binding（虚拟卷）.....	17
certificateDetails.....	18
集群.....	18
clusterAdmin.....	19
clusterCapacity.....	20
clusterconfig.....	21
clusterInfo.....	22
clusterPair.....	23
clusterStats.....	23
clusterStructure.....	25
drive.....	26
driveStats.....	27
error.....	28
event.....	28
fault.....	29
fibreChannelPort.....	31
fipsErrorNodeReport.....	32
fipsNodeReport.....	32
fipsReport.....	32
groupSnapshot.....	33
hardwareInfo.....	33
host（虚拟卷）.....	34
IDPConfigInfo.....	35
initiator.....	35
keyProviderKmp.....	36
keyServerKmp.....	36
ldapConfiguration.....	37
loggingServer.....	38
network（绑定接口）.....	38
network（所有接口）.....	41
network（以太网接口）.....	41
network（本地接口）.....	42
network（SNMP）.....	43
networkInterface.....	44

node.....	44
节点保护域.....	46
nodeStats.....	46
ontapVersionInfo.....	47
pendingActiveNode.....	47
pendingNode.....	48
保护域.....	49
protectionDomainLevel.....	50
protectionDomainResiliency.....	50
protectionDomainTolerance.....	51
protectionSchemeResiliency.....	51
protectionSchemeTolerance.....	51
protocolEndpoint.....	52
QoS.....	52
QoSPolicy.....	53
remoteClusterSnapshotStatus.....	53
schedule.....	54
session（光纤通道）.....	55
session（iSCSI）.....	56
snapMirrorAggregate.....	57
snapMirrorClusterIdentity.....	57
snapMirrorEndpoint.....	57
snapMirrorJobScheduleCronInfo.....	58
snapMirrorLunInfo.....	58
snapMirrorNetworkInterface.....	59
snapMirrorNode.....	60
snapMirrorPolicy.....	60
snapMirrorPolicyRule.....	61
snapMirrorRelationship.....	61
snapMirrorVolume.....	63
snapMirrorVolumeInfo.....	63
snapMirrorVserver.....	64
snapMirrorVserverAggregateInfo.....	65
snapshot.....	65
snmpTrapRecipient.....	66
storageContainer.....	67
syncJob.....	67
task（虚拟卷）.....	69
usmUser.....	70
virtualNetwork.....	70
virtualVolume.....	71
volume.....	72
volumeAccessGroup.....	74
volumePair.....	75
volumeStats.....	75
通用方法.....	78
GetAPI.....	78
GetAsyncResult.....	84
GetCompleteStats.....	86
GetLimits.....	87
GetOrigin.....	88
GetRawStats.....	89

ListAsyncResults.....	89
帐户方法.....	92
AddAccount.....	92
GetAccountByID.....	93
GetAccountByName.....	94
GetAccountEfficiency.....	95
ListAccounts.....	96
ModifyAccount.....	98
RemoveAccount.....	99
管理员 API 方法.....	101
AddClusterAdmin.....	101
GetCurrentClusterAdmin.....	102
GetLoginBanner.....	103
ListClusterAdmins.....	104
ModifyClusterAdmin.....	105
RemoveClusterAdmin.....	106
SetLoginBanner.....	107
集群 API 方法.....	109
AddNodes.....	109
ClearClusterFaults.....	111
CreateClusterInterfacePreference.....	111
DeleteClusterInterfacePreference.....	112
EnableFeature.....	113
GetClusterCapacity.....	114
GetClusterFullThreshold.....	115
GetClusterHardwareInfo.....	118
GetClusterInfo.....	119
GetClusterInterfacePreference.....	120
GetClusterMasterNodeID.....	121
GetClusterStats.....	121
GetClusterVersionInfo.....	123
GetFeatureStatus.....	124
GetLoginSessionInfo.....	126
GetNodeHardwareInfo.....	127
GetNodeStats.....	128
ListActiveNodes.....	129
ListAllNodes.....	129
ListClusterFaults.....	131
ListClusterInterfacePreferences.....	132
ListEvents.....	133
ListNodeStats.....	135
ListISCSISessions.....	136
ListServices.....	137
ListPendingNodes.....	139
ListPendingActiveNodes.....	141
ModifyClusterFullThreshold.....	142
ModifyClusterInterfacePreference.....	145
RemoveNodes.....	146

SetLoginSessionInfo.....	147
Shutdown.....	148
集群创建 API 方法.....	149
CheckProposedCluster.....	149
CreateCluster.....	150
GetBootstrapConfig.....	151
驱动器 API 方法.....	154
AddDrives.....	154
GetDriveHardwareInfo.....	155
GetDriveStats.....	157
ListDrives.....	158
ListDriveStats.....	159
RemoveDrives.....	161
SecureEraseDrives.....	162
光纤通道 API 方法.....	164
GetVolumeAccessGroupLunAssignments.....	164
ListFibreChannelPortInfo.....	165
ListFibreChannelSessions.....	167
ListNodeFibreChannelPortInfo.....	168
ModifyVolumeAccessGroupLunAssignments.....	170
LDAP API 方法.....	172
AddLdapClusterAdmin.....	172
EnableLdapAuthentication.....	173
DisableLdapAuthentication.....	175
GetLdapConfiguration.....	176
TestLdapAuthentication.....	177
多因素身份验证 API 方法.....	179
AddDdpClusterAdmin.....	179
CreateIDPDCConfiguration.....	180
DeleteAuthSession.....	181
DeleteAuthSessionSBYClusterAdmin.....	182
DeleteAuthSessionSBYUserName.....	183
DeleteIDPDCConfiguration.....	185
DisableIDDPAuthentication.....	186
启用 IDPAuthentication.....	186
GetIDPAuthenticationState.....	187
ListActiveAuthSessions.....	188
LISTIDPDCConfigurations.....	188
更新 IDEDPCConfiguration.....	190
会话身份验证 API 方法.....	192
ListAuthSessionsByClusterAdmin.....	192
ListAuthSessionSBYUserName.....	193

节点 API 方法	195
CheckPingOnVlan.....	195
CheckProposedNodeAdditions.....	197
CreateClusterSupportBundle.....	199
CreateSupportBundle.....	200
DeleteAllSupportBundles.....	202
DisableSsh.....	202
EnableSsh.....	203
GetClusterConfig.....	203
GetClusterState.....	204
GetConfig.....	205
GetDriveConfig.....	206
GetHardwareConfig.....	207
GetHardwareInfo.....	209
GetIpmiConfig.....	211
GetIpmiInfo.....	214
GetNetworkConfig.....	217
GetNetworkInterface.....	217
GetNodeActiveTlsCiphers.....	220
GetNodeFipsDrivesReport.....	221
GetNodeSSLCertificate.....	221
GetNodeSupportedTlsCiphers.....	223
GetPendingOperation.....	224
GetSshInfo.....	225
ListNetworkInterfaces.....	225
ListDriveHardware.....	227
ListTests.....	229
ListUtilities.....	230
RemoveNodeSSLCertificate.....	230
ResetDrives.....	231
ResetNode.....	232
ResetNodeSupplementalTlsCiphers.....	234
RestartNetworking.....	234
RestartServices.....	235
SetClusterConfig.....	236
SetConfig.....	237
SetNetworkConfig.....	239
SetNodeSSLCertificate.....	240
SetNodeSupplementalTlsCiphers.....	241
Shutdown.....	243
TestConnectEnsemble.....	244
TestConnectMvip.....	245
TestConnectSvip.....	247
TestDrives.....	249
TestHardwareConfig.....	249
TestLocateCluster.....	250
TestLocalConnectivity.....	251
TestNetworkConfig.....	253
TestPing.....	255
TestRemoteConnectivity.....	257

复制 API 方法	260
集群配对操作顺序.....	260
卷配对操作顺序.....	260
配对集群支持的复制模式.....	261
CompleteClusterPairing.....	261
CompleteVolumePairing.....	262
ListClusterPairs.....	263
ListActivePairedVolumes.....	264
ModifyVolumePair.....	266
RemoveClusterPair.....	267
RemoveVolumePair.....	267
StartClusterPairing.....	268
StartVolumePairing.....	269
安全 API 方法	271
AddKeyServerToProviderKmp.....	271
CreateKeyProviderKmp.....	272
CreateKeyServerKmp.....	273
CreatePublicPrivateKeyPair.....	274
DeleteKeyProviderKmp.....	276
DeleteKeyServerKmp.....	276
DisableEncryptionAtRest.....	277
EnableEncryptionAtRest.....	278
GetClientCertificateSignRequest.....	279
GetKeyProviderKmp.....	280
GetKeyServerKmp.....	281
ListKeyProvidersKmp.....	282
ListKeyServersKmp.....	284
ModifyKeyServerKmp.....	286
RemoveKeyServerFromProviderKmp.....	288
TestKeyProviderKmp.....	289
TestKeyServerKmp.....	290
SnapMirror API 方法	291
AbortSnapMirrorRelationship.....	291
BreakSnapMirrorRelationship.....	291
BreakSnapMirrorVolume.....	292
CreateSnapMirrorEndpoint.....	293
CreateSnapMirrorEndpointUnmanaged.....	293
CreateSnapMirrorRelationship.....	294
CreateSnapMirrorVolume.....	295
DeleteSnapMirrorEndpoints.....	296
DeleteSnapMirrorRelationships.....	296
GetOntapVersionInfo.....	296
GetSnapMirrorClusterIdentity.....	297
InitializeSnapMirrorRelationship.....	298
ListSnapMirrorAggregates.....	298
ListSnapMirrorEndpoints.....	299
ListSnapMirrorLuns.....	299
ListSnapMirrorNetworkInterfaces.....	300

ListSnapMirrorNodes.....	300
ListSnapMirrorPolicies.....	301
ListSnapMirrorSchedules.....	301
ListSnapMirrorRelationships.....	302
ListSnapMirrorVolumes.....	302
ListSnapMirrorVservers.....	303
ModifySnapMirrorEndpoint.....	304
ModifySnapMirrorEndpoint (非受管)	305
ModifySnapMirrorRelationship.....	305
UpdateSnapMirrorRelationship.....	306
QuiesceSnapMirrorRelationship.....	307
ResumeSnapMirrorRelationship.....	307
ResyncSnapMirrorRelationship.....	308

系统配置 API 方法.....309

DisableBMCColdReset.....	309
DisableClusterSsh.....	309
DisableSnmp.....	310
启用 BMCColdReset.....	311
EnableClusterSsh.....	312
EnableSnmp.....	313
GetBinAssignmentProperties.....	314
GetClusterSshInfo.....	316
GetClusterStructure.....	317
GetFipsReport.....	317
GetLldpConfig.....	318
GetLldpInfo.....	319
GetNodeFipsDrivesReport.....	320
GetNtpInfo.....	320
GetNvramInfo.....	321
GetProtectionDomainLayout.....	322
GetRemoteLoggingHosts.....	324
GetSnmpACL.....	325
GetSnmpInfo.....	326
GetSnmpState.....	327
GetSnmpTrapInfo.....	328
GetSSLCertificate.....	329
ListProtectionDomainLevels.....	331
RemoveSSLCertificate.....	332
ResetNetworkConfig.....	333
ResetSupplementalTlsCiphers.....	333
SetClusterStructure.....	334
SetLldpConfig.....	335
SetNtpInfo.....	336
SetProtectionDomainLayout.....	336
SetRemoteLoggingHosts.....	339
SetSnmpACL.....	340
SetSnmpInfo.....	341
SetSnmpTrapInfo.....	343
SetSSLCertificate.....	344
SnmpSendTestTraps.....	346
TestAddressAvailability.....	346

多租户网络 API 方法	348
虚拟网络命名约定.....	348
AddVirtualNetwork.....	348
ModifyVirtualNetwork.....	350
ListVirtualNetworks.....	352
RemoveVirtualNetwork.....	354
卷 API 方法	355
AddInitiatorsToVolumeAccessGroup.....	355
AddVolumesToVolumeAccessGroup.....	356
CancelClone.....	358
CancelGroupClone.....	358
CloneMultipleVolumes.....	359
CloneVolume.....	361
CopyVolume.....	364
CreateQoSPolicy.....	365
CreateVolume.....	367
CreateVolumeAccessGroup.....	370
CreateBackupTarget.....	372
CreateInitiators.....	373
DeleteInitiators.....	375
DeleteQoSPolicy.....	376
DeleteVolume.....	376
DeleteVolumes.....	378
DeleteVolumeAccessGroup.....	380
GetBackupTarget.....	381
GetVolumeStats.....	382
GetDefaultQoS.....	383
GetQoSPolicy.....	384
GetVolumeCount.....	386
GetVolumeAccessGroupEfficiency.....	386
GetVolumeEfficiency.....	387
ListActiveVolumes.....	389
ListBackupTargets.....	390
ListBulkVolumeJobs.....	390
ListDeletedVolumes.....	392
ListInitiators.....	393
ListQoSPolicies.....	394
ListSyncJobs.....	396
ListVolumeQoSHistograms.....	397
ListVolumes.....	399
ListVolumeStats.....	401
ListVolumeAccessGroups.....	403
ListVolumesForAccount.....	404
ListVolumeStatsByAccount.....	405
ListVolumeStatsByVirtualVolume.....	407
ListVolumeStatsByVolume.....	408
ListVolumeStatsByVolumeAccessGroup.....	410
ModifyBackupTarget.....	411
ModifyInitiators.....	412
ModifyQoSPolicy.....	415

ModifyVolume.....	416
ModifyVolumes.....	420
ModifyVolumeAccessGroup.....	424
PurgeDeletedVolume.....	426
PurgeDeletedVolumes.....	427
RemoveBackupTarget.....	428
RemoveInitiatorsFromVolumeAccessGroup.....	429
RemoveVolumesFromVolumeAccessGroup.....	430
RestoreDeletedVolume.....	431
SetDefaultQoS.....	432
StartBulkVolumeRead.....	433
StartBulkVolumeWrite.....	435
UpdateBulkVolumeStatus.....	436
卷快照 API 方法.....	439
快照概述.....	439
CreateGroupSnapshot.....	439
CreateSchedule.....	442
CreateSnapshot.....	448
DeleteGroupSnapshot.....	450
DeleteSnapshot.....	451
GetSchedule.....	452
ListGroupSnapshots.....	453
ListSchedules.....	455
ListSnapshots.....	456
ModifyGroupSnapshot.....	457
ModifySchedule.....	459
ModifySnapshot.....	462
RollbackToGroupSnapshot.....	464
RollbackToSnapshot.....	466
虚拟卷 API 方法.....	469
CreateStorageContainer.....	469
DeleteStorageContainers.....	470
GetStorageContainerEfficiency.....	471
GetVirtualVolumeCount.....	472
ListProtocolEndpoints.....	472
ListStorageContainers.....	474
ListVirtualVolumeBindings.....	475
ListVirtualVolumeHosts.....	476
ListVirtualVolumes.....	477
ListVirtualVolumeTasks.....	479
ModifyStorageContainer.....	480
集群故障代码.....	482
访问控制.....	493
响应示例.....	497

GetConfig.....	497
GetClusterHardwareInfo.....	499
GetLldpInfo.....	509
GetNetworkConfig.....	532
GetNodeHardwareInfo (iSCSI 的输出)	535
GetNodeHardwareInfo (光纤通道节点的输出)	536
GetNvramInfo.....	542
ListActiveNodes.....	549
ListActiveVolumes.....	551
TestHardwareConfig.....	558
联系 NetApp 支持部门.....	563
版权.....	564
商标.....	565
意见反馈及更新通知.....	566

关于 Element 软件 API

Element API 基于 HTTPS 上的 JSON-RPC 协议。JSON-RPC 是一种基于轻型 JSON 数据交换格式的简单文本 RPC 协议。客户端库可用于所有主要的编程语言。

可以通过 HTTPS POST 请求向 API 端点发出 API 请求。POST 请求的正文是一个 JSON-RPC 请求对象。API 当前不支持批处理请求（即在一个 POST 中包含多个请求对象）。提交 API 请求时，可以使用“application/json-rpc”作为请求的内容类型，并确保正文未采用表单编码格式。

注：Element Web UI 将使用本文中所述的 API 方法。可以通过启用 API 日志来监控 UI 中的 API 操作；这样您可以查看对系统发出的方法。您可以同时启用请求和响应，查看系统对所发出的方法的回复。

除非另有规定，否则 API 响应中的所有日期字符串均为 UTC+0 格式。

注：如果存储集群负载过重，或者您提交了大量连续 API 请求而中间没有任何延迟，则某个方法可能会失败并返回错误“xDBVersionMismatch”。如果发生这种情况，请重试此方法调用。

请求对象成员

每个 Element 软件 API 请求都具有以下基本部分：

名称	说明	类型	默认值	必需
method	要调用的方法的名称。	字符串	无	是
参数	包含正在调用的方法的参数的对象。需要已命名参数。不允许使用定位参数（作为数组传递）。	JSON 对象	{}	否
id	用于将请求与结果中返回的响应匹配的标识符。	字符串或整型	{}	否

响应对象成员

每个 Element 软件 API 响应正文都具有以下基本部分：

名称	说明	类型
result	此方法返回的对象。系统将返回一个对象，其中包含与该方法的已记录返回值对应的指定成员。如果已发生错误，则此成员不存在。	JSON 对象
error	发生错误时返回的对象。此成员仅在已发生错误时存在。	对象
id	用于将请求与请求中提供的响应匹配的标识符。	字符串或整型
unusedParameters	至少有一个不正确参数已传递到 API 方法且尚未使用的警告消息。	对象

请求端点

API 中使用三种类型的请求端点（存储集群、存储集群创建和每节点）。您应始终使用您的 Element 软件版本支持的最新端点。

API 中的三个请求端点通过以下方式指定：

集群 API 方法

用于存储集群范围 API 请求的 HTTPS 端点是 `https://<mvip>/json-rpc/<api-version>`，其中：

- `<mvip>` 是存储集群的管理虚拟 IP 地址。
- `<api-version>` 是所使用的 API 版本。

集群创建和启动 API 方法

用于创建存储集群和访问启动 API 请求的 HTTPS 端点是 `https://<nodeIP>/json-rpc/<api-version>`，其中：

- `<nodeIP>` 是要添加到集群中的节点的 IP 地址。
- `<api-version>` 是所使用的 API 版本。

每个节点 API 方法

用于单个存储节点 API 请求的 HTTPS 端点是 `https://<nodeIP>:442/json-rpc/<api-version>`，其中：

- `<nodeIP>` 是存储节点的管理 IP 地址；442 是运行 HTTPS 服务器的端口。
- `<api-version>` 是所使用的 API 版本。

API 身份验证

可以使用 API 向系统进行身份验证，方法是所有 API 请求包括一个 HTTP 基本身份验证标头。如果您忽略身份验证信息，系统则会拒绝未经身份验证的请求并返回 HTTP 401 响应。系统支持通过 TLS 进行 HTTP 基本身份验证。

使用集群管理员帐户进行 API 身份验证。

异步方法

某些 API 方法为异步方法，这意味着，此类方法返回时，其执行的操作可能没有完成。异步方法将返回一个句柄，您可查询该句柄以查看操作状态。某些操作的状态信息可能包括完成的百分比。

查询异步操作时，其结果可为以下类型之一：

- `DriveAdd`：系统正在将驱动器添加到集群中。
- `BulkVolume`：系统正在卷之间执行复制操作，例如备份或还原。
- `Clone`：系统正在克隆卷。
- `DriveRemoval`：系统正在从驱动器中复制数据，准备将其从集群中删除。
- `RtftiPendingNode`：系统正在节点上安装兼容软件，以便稍后将其添加到集群中。

使用异步方法或获取正在运行的异步操作的状态时，请注意以下几点：

- 各个方法文档中有对异步方法的说明。
- 异步方法会返回“`asyncHandle`”，此句柄可由发出此句柄的 API 方法识别。您可以使用此句柄轮询异步操作的状态或结果。

- 您可以使用 `GetAsyncResult` 方法获取各个异步方法的结果。使用 `GetAsyncResult` 查询已完成的操作时，系统将返回此结果并自动从系统中清除此结果。使用 `GetAsyncResult` 查询未完成的操作时，系统将返回此结果，但不会将其清除。
- 您可以使用 `ListAsyncResults` 方法获取所有正在运行或已完成的异步方法的状态和结果。在这种情况下，系统不会清除已完成操作的结果。

Attributes

许多 API 请求和响应使用对象以及简单类型。对象是键-值对的集合，其中值可以是简单类型，也可以是其他对象。属性是自定义的名称-值对（可由用户在 JSON 对象中设置）。使用某些方法可以在创建或修改对象时添加属性。

已编码属性对象的大小限制为 1000 字节。

对象成员

此对象包含以下成员：

名称	说明	类型
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象

请求示例

以下请求示例使用 `AddClusterAdmin` 方法：

```
{
  "method": "AddClusterAdmin",
  "params": {
    "username": "joeadmin",
    "password": "68!5Aru268)$",
    "access": [
      "volume",
      "reporting"
    ],
    "attributes": {
      "name1": "value1",
      "name2": "value2",
      "name3": "value3"
    }
  }
}
```

通用对象

Element 软件 API 采用 JSON 对象来表示有组织的数据。其中许多 API 方法都会使用这些对象进行数据输入和输出。本节记录了这些常用对象；仅在一种方法中使用的对象会与此方法记录在一起，不会记录在本节中。

account

account 对象包含有关帐户的信息。此对象仅包含有关帐户的“已配置”信息，而不包含任何运行时信息或使用情况信息。

对象成员

此对象包含以下成员：

名称	说明	类型
accountID	帐户的唯一帐户 ID。	整型
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
initiatorSecret	启动程序 CHAP 密钥。	字符串
status	帐户的当前状态。可能值： <ul style="list-style-type: none">active: 该帐户处于活动状态。locked: 该帐户已锁定。removed: 该帐户已删除并清除。	字符串
storageContainerID	与此帐户关联的虚拟卷存储容器的唯一 ID。	UUID
targetSecret	目标 CHAP 密钥。	字符串
username	帐户的用户名。	字符串
volumes	该帐户拥有的卷的卷 ID 列表。	整型数组

相关参考

[AddAccount](#) on page 92

您可以使用 AddAccount 向系统添加新帐户。创建新帐户后，您还可以使用此方法在该帐户下创建新卷。为帐户指定的 CHAP 设置将应用于此帐户拥有的所有卷。

[GetAccountByID](#) on page 93

您可以使用 GetAccountByID 获取有关具有给定帐户 ID 的帐户的详细信息。

[GetAccountByName](#) on page 94

您可以使用 GetAccountByName 获取有关具有给定用户名的特定帐户的详细信息。

[ListAccounts](#) on page 96

您可以使用 ListAccounts 获取存储租户帐户的完整列表，并提供可选的寻呼支持。元素帐户允许访问卷。

AuthSessionInfo

authSessionInfo 该对象包含有关验证会话的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
AccessGroupList	用户的访问组列表。	字符串数组
authMethod	集群管理员用户拥有的授权类型。可能值： <ul style="list-style-type: none"> • 通过 LDAP 进行 LDAP 身份验证。 • 集群—通过存储在集群数据库中的用户名和密码进行身份验证。 • IDP —通过第三方身份认证提供者进行验证。 	字符串
集群管理	与此会话关联的集群管理员列表。对于与 LDAP 或第三方身份认证提供者（IDP）相关的会话、这将是与此会话关联的匹配的群集管理单元的聚合列表。	整型数组
FinalTimeout	会话失效的时间。此时间在创建会话时设置，不能更改。	字符串
IDPConfigVersion	创建会话时的 IDP 配置版本。	整型
上次访问超时	会话因非活动状态而失效的时间。访问会话以供使用时、直到到达终止超时会话失效的时间为止、该会话设置为新值。	字符串
会话创建时间	创建会话的时间。	字符串
sessionID	此会话的 UUID。	UUID
username	与此会话关联的用户名。对于与 LDAP 相关的会话、这将是用户的 LDAP DN。对于与第三方 IDP 相关的会话、这将是一个任意名称值对、用于在会话中审核操作。它不一定与集群上的集群管理员名称匹配。例如、SAML 主题 NameID、但这将由 IDP 配置和 SAML 声明的结果内容来决定。	字符串

bulkVolumeJob

bulkVolumeJob 对象包含批量卷读取或写入操作（例如克隆或快照创建）的相关信息。

对象成员

此对象包含以下成员：

名称	说明	类型
attributes	批量卷作业的 JSON 属性。	JSON 对象
bulkVolumeID	内部批量卷作业 ID。	整型
createTime	为批量卷作业创建的时间戳（UTC+0 格式）。	ISO 8601 日期字符串
elapsedTime	自开始作业以来经过的秒数。	字符串
format	批量卷操作的格式。可能值： <ul style="list-style-type: none"> • compressed • native 	字符串

名称	说明	类型
key	由批量卷会话创建的唯一密钥。	字符串
percentComplete	由操作报告的完成百分比。	整型
remainingTime	估计剩余时间，以秒为单位。	整型
srcVolumeID	源卷 ID。	整型
status	操作的状态。可能值： <ul style="list-style-type: none"> preparing active done failed 	字符串
script	脚本的名称（如果已提供）。	字符串
snapshotID	快照的 ID（如果该快照位于批量卷作业的源中）。	整型
type	批量操作的类型。可能值： <ul style="list-style-type: none"> read write 	字符串

binding（虚拟卷）

binding 对象包含有关虚拟卷绑定的信息。可以使用 `ListVirtualVolumeBindings` API 方法为所有虚拟卷检索此信息的列表。

对象成员

此对象包含以下成员：

名称	说明	类型
protocolEndpointID	协议端点的唯一 ID。	UUID
protocolEndpointInBandID	协议端点的 scsiNAADeviceID。	字符串
protocolEndpointType	协议端点的类型。SCSI 是为协议端点类型返回的唯一值。	字符串
virtualVolumeBindingID	虚拟卷绑定对象的唯一 ID。	整型
virtualVolumeHostID	虚拟卷主机的唯一 ID。	UUID
virtualVolumeID	虚拟卷的唯一 ID。	UUID
virtualVolumeSecondaryID	虚拟卷的辅助 ID。	字符串

相关参考

[ListVirtualVolumeBindings](#) on page 475

您可以使用 `ListVirtualVolumeBindings` 方法获取集群中绑定到协议端点的所有虚拟卷的列表。

[protocolEndpoint](#) on page 52

`protocolEndpoint` 对象包含协议端点的属性。可以通过使用 `API` 方法 `ListProtocolEndpoints` 为集群中的所有协议端点检索此信息。

certificateDetails

`certificateDetails` 对象包含有关安全证书的解码信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>issuer</code>	颁发者的名称。	字符串
<code>modulus</code>	公共密钥的模量。	字符串
<code>notAfter</code>	证书的到期日期。	ISO 8601 字符串
<code>notBefore</code>	证书的开始日期。	ISO 8601 字符串
<code>serial</code>	证书序列号。	字符串
<code>sha1Fingerprint</code>	经 DER 编码的证书的摘要。	字符串
<code>subject</code>	主题名称。	字符串

集群

`cluster` 对象包含节点用于与集群通信的信息。可以使用 `GetClusterConfig` `API` 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>cipi</code>	用于集群通信的网络接口。	字符串
集群	唯一集群名称。	字符串
<code>encryptionCapable</code>	指示节点是否支持驱动器加密。	布尔型
<code>ensemble</code>	加入集群的节点。	字符串数组
<code>fipsDriveConfiguration</code>	指示节点是否支持经 FIPS 140-2 认证的驱动器。	布尔型
<code>mipi</code>	用于进行节点管理的网络接口。	字符串
<code>name</code>	集群名称。	字符串
<code>nodeID</code>	集群中的节点的节点 ID。	字符串
<code>pendingNodeID</code>	集群中的待定节点的 ID。	整型
<code>role</code>	标识节点的角色。	整型
<code>sipi</code>	用于存储流量的网络接口。	字符串

名称	说明	类型
state	节点的当前状态。可能值： <ul style="list-style-type: none"> • Available: 节点尚未配置集群名称。 • Pending: 节点正在等待添加到已命名的特定集群，可以进行添加。 • Active: 节点是集群的活动成员，不能添加到其他集群。 • PendingActive: 节点当前正还原出厂软件映像，尚未成为集群的活动成员。完成后，将过渡到 Active 状态。 	字符串
version	在节点上运行的软件版本。	字符串

成员可修改性和节点状态

下表说明了在每种可能的节点状态下是否可修改相应的对象参数。

参数名称	可用状态	待定状态	活动状态
cipi	否	否	否
cluster	是	是	否
encryptionCapable	否	否	否
ensemble	否	否	否
mipi	是	是	否
name	是	是	是
nodeID	否	否	否
pendingNodeID	否	否	否
role	否	否	否
sipi	否	否	否
state	否	否	否
version	否	否	否

相关参考

[GetClusterConfig](#) on page 203

您可以使用 `GetClusterConfig` API 方法返回有关节点与其集群通信所使用的集群配置的信息。

clusterAdmin

`clusterAdmin` 对象包含有关当前集群管理员用户的信息。可以使用 API 方法 `GetCurrentClusterAdmin` 检索管理员用户信息。

对象成员

此对象包含以下成员：

名称	说明	类型
访问	该集群管理员可使用的方法。	字符串数组

名称	说明	类型
authMethod	集群管理员用户拥有的授权类型。可能值： <ul style="list-style-type: none"> LDAP 集群 Local 	字符串
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
clusterAdminID	该集群管理员用户的集群管理员 ID。	整型
username	该集群管理员的用户名。	字符串

相关参考

[GetCurrentClusterAdmin](#) on page 102

您可以使用 `GetCurrentClusterAdmin` 方法返回当前主集群管理员的信息。主集群管理员是在创建集群之时创建的。

clusterCapacity

`clusterCapacity` 对象包含集群的容量测量结果概要。可以使用 API 方法 `GetClusterCapacity` 获取集群容量信息。对象成员中的空间测量结果是以字节计算的。

对象成员

此对象包含以下成员：

名称	说明	类型
activeBlockSpace	块驱动器上的空间量。此成员包含其他信息，例如可清理的元数据条目和空间。	整型
activeSessions	正在与集群通信的活动 iSCSI 会话的数量。	整型
averageIOPS	自午夜（协调世界时 [Coordinated Universal Time, UTC]）以来集群的平均 IOPS。	整型
clusterRecentIOSize	集群中所有卷的 IOPS 的平均大小。	整型
currentIOPS	过去 5 秒内集群中所有卷的平均 IOPS。	整型
maxIOPS	当前集群的估计的最大 IOPS 容量。	整型
maxOverProvisionableSpace	最大的可配置空间量。这是一个计算所得值。如果当前已配置空间与新卷大小之和将超过此数字，则无法创建新卷。该值的计算方法如下： $\text{maxOverProvisionableSpace} = \text{maxProvisionedSpace} * \text{maxMetadataOverProvisionFactor}$	整型
maxProvisionedSpace	所有卷均已全满（没有精简配置元数据）时总的可配置空间量。	整型
maxUsedMetadataSpace	卷驱动器上用于存储元数据的字节数。	整型
maxUsedSpace	所有活动的块驱动器上总的空间量。	整型
nonZeroBlock	上次垃圾回收操作完成之后包含数据的 4KiB 块的总数。	整型
peakActiveSessions	自午夜 (UTC) 以来 iSCSI 连接的高峰数量。	整型

名称	说明	类型
peakIOPS	自午夜 (UTC) 以来 currentIOPS 的最高值。	整型
provisionedSpace	集群中所有卷上的已配置空间总量。	整型
timestamp	采集此集群容量样本时的日期和时间 (UTC+0 格式)。	ISO 8601 字符串
totalOps	在集群的整个生命周期执行的 I/O 操作的总数。	整型
uniqueBlocks	存储在块驱动器上的块的总数。值包括复制的块。	整型
uniqueBlocksUsedSpace	uniqueBlocks 在块驱动器上占用的总数据量。有关此数字与 uniqueBlocks 值之间关系的信息, 请参见 GetclusterCapacity 方法。	整型
usedMetadataSpace	卷驱动器上用于存储元数据的总字节数。	整型
usedMetadataSpaceInSnapshots	卷驱动器上用于存储快照中的唯一数据的字节数。此数字提供可通过删除系统上的所有快照重新获取的元数据空间估计值。	整型
usedSpace	由系统中所有块驱动器使用的总空间量。	整型
zeroBlocks	上次垃圾回收操作完成之后不包含数据的空 4KiB 块的总数。	整型

相关参考

[GetClusterCapacity](#) on page 114

可以使用 GetClusterCapacity 返回整个存储集群的容量测量结果概要。此方法将返回一些字段, 可用于计算 Element Web UI 中显示的效率等级。可以在脚本中使用效率计算, 以返回精简配置、重复数据删除、数据压缩和整体效率的效率等级。

clusterconfig

clusterConfig 该对象返回节点用于与群集通信的信息。

对象成员

此对象包含以下成员:

名称	说明	类型
cipi	用于集群通信的网络接口。	字符串
集群	集群的唯一名称。	字符串
encryptionCapable	指定节点是否支持加密。	布尔型
ensemble	参与集群的节点。	字符串数组
fipsDriveConfiguration	指定节点是否支持 FIPS 140-2 认证驱动器。	布尔型
HasLocalAdmin	指定集群是否具有本地管理员。	布尔型
mipi	用于进行节点管理的网络接口。	字符串
name	集群的唯一标识符。	字符串
nodeID	节点的唯一标识符。	整型
pendingNodeID	暂挂节点的唯一标识符。	整型
role	标识节点的角色。	字符串

名称	说明	类型
sipi	用于存储的网络接口。	字符串
state	指示节点的状态。	字符串
version	指示节点的版本。	字符串

clusterInfo

clusterInfo 对象包含节点用于与集群通信的信息。可以使用 GetClusterInfo API 方法获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
defaultProtectionScheme	新卷默认使用的保护方案，除非在 Create Volume 方法调用中提供了保护方案。	字符串
enabledProtectionSchemes	此存储集群上已启用的所有保护方案的列表。	字符串数组
encryptionAtRestState	空闲时加密功能的状态。可能值： <ul style="list-style-type: none"> Enabling: 即将启用空闲时加密。 Enabled: 已启用空闲时加密。 Disabling: 即将禁用空闲时加密。 Disabled: 已禁用空闲时加密。 	字符串
ensemble	加入集群的节点。	字符串数组
mvip	集群在管理网络上的浮动（虚拟）IP 地址。	字符串
mvipInterface	与 MVIP 地址关联的物理接口。	字符串
mvipNodeID	持有主 MVIP 地址的节点。	字符串
mvipVlanTag	MVIP 地址的 VLAN 标识符。	字符串
name	唯一集群名称。	字符串
repCount	要存储在集群中的每个数据段的副本数。有效值为“2”。	整型
supportedProtectionSchemes	此存储集群上支持的所有保护方案的列表。	字符串数组
svip	集群在存储 (iSCSI) 网络上的浮动（虚拟）IP 地址。	字符串
svipInterface	与主 SVIP 地址关联的物理接口。	字符串
svipNodeID	持有主 SVIP 地址的节点。	字符串
svipVlanTag	主 SVIP 地址的 VLAN 标识符。	字符串
uniqueID	集群的唯一 ID。	字符串
uuid	集群的唯一标识符。	字符串

相关参考

[GetClusterInfo](#) on page 119

您可以使用 `GetClusterInfo` 方法返回有关集群的配置信息。

clusterPair

`clusterPair` 对象包含与本地集群配对的集群的相关信息。您可以使用 `ListClusterPairs` 方法检索本地集群的 `clusterPair` 对象的列表。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>clusterName</code>	对中另一个集群的名称。	字符串
<code>clusterPairID</code>	为对中每个集群指定的唯一 ID。	整型
<code>clusterPairUUID</code>	集群对的通用唯一标识符。	字符串
<code>UUID</code>	集群对中远程集群的唯一标识符。	整型
延迟	两个集群之间的延迟（以毫秒为单位）。	整型
<code>mvip</code>	已配对集群的管理连接的 IP 地址。	字符串
<code>status</code>	两个配对集群之间的连接状态。可能值： <ul style="list-style-type: none"> Unconfigured Connected Misconfigured Disconnected 	字符串
<code>version</code>	集群对中另一个集群的 Element 版本。	字符串

相关参考

[ListClusterPairs](#) on page 263

您可以使用 `ListClusterPairs` 方法列出与当前集群配对的所有集群。此方法将返回有关活动和待定集群配对的信息，例如有关当前配对以及集群配对连接性和延迟性（以毫秒为单位）的统计信息。

clusterStats

`clusterStats` 对象包含集群的统计数据。该对象中包含的许多与卷相关的统计信息都是针对集群中的所有卷计算得出的平均值。您可以使用 `GetClusterStats` 方法为集群检索此信息。

对象成员

此对象包含以下成员：

名称	说明	计算方式	类型
<code>actualIOPS</code>	整个集群在过去 500 毫秒的当前实际 IOPS。	时间点	整型
<code>averageIOPSize</code>	集群在过去 500 毫秒的最新 I/O 的平均大小（以字节为单位）。	时间点	整型

名称	说明	计算方式	类型
clientQueueDepth	对卷执行的未完成的读取和写入操作的数量。	不适用	整型
clusterUtilization	正在利用的集群容量。	不适用	浮点型
latencyUsec	集群在过去 500 毫秒完成操作的平均时间（以微秒为单位）。	时间点	整型
normalizedIOPS	整个集群在过去 500 毫秒的平均 IOPS 数。	时间点	整型
readBytes	自集群创建以来从集群读取的总累积字节数。	单调递增	整型
readBytesLastSample	在最后一个采样周期内从集群读取的总字节数。	时间点	整型
readLatencyUsec	集群在过去 500 毫秒完成读取操作的平均时间（以微秒为单位）。	时间点	整型
readLatencyUsecTotal	自集群创建以来执行读取操作所用的总时间。	单调递增	整型
readOps	自集群创建以来集群的总累积读取操作数。	单调递增	整型
readOpsLastSample	最后一个采样周期内读取操作的总数。	时间点	整型
samplePeriodMsec	采样周期的长度（以毫秒为单位）。	不适用	整型
servicesCount	在集群上运行的服务数量。如果等于 servicesTotal，这表示从所有节点收集了有效的统计信息。	时间点	整型
servicesTotal	在集群上运行的预期服务总数。	不适用	整型
timestamp	当前时间（UTC+0 格式）。	不适用	ISO 8601 日期字符串
unalignedReads	自集群创建以来集群的总累积未对齐读取操作数。	单调递增	整型
unalignedWrites	自集群创建以来集群的总累积未对齐写入操作数。	单调递增	整型
writeBytes	自集群创建以来写入集群的总累积字节数。	单调递增	整型
writeBytesLastSample	在最后一个采样周期内写入集群的总字节数。	单调递增	整型
writeLatencyUsec	集群在过去 500 毫秒完成写入操作的平均时间（以微秒为单位）。	时间点	整型

名称	说明	计算方式	类型
writeLatencyUsecTotal	自集群创建以来执行写入操作所用的总时间。	单调递增	整型
writeOps	自集群创建以来集群的总累积写入操作数。	单调递增	整型
writeOpsLastSample	最后一个采样周期内写入操作的总数。	时间点	整型

相关参考

[GetClusterStats](#) on page 121

您可以使用 `GetClusterStats` 方法检索集群的活动测量结果概要。返回值从创建集群之后开始累计。

clusterStructure

`clusterStructure` 对象用于保存由 `GetClusterStructure` 方法创建的集群配置备份信息。您可以使用 `SetClusterStructure` 方法将此信息还原到要重建的存储集群中。

对象成员

此对象包含以下方法的返回信息组合：

- [GetClusterInfo](#)
- [ListAccounts](#)
- [ListInitiators](#)
- [ListVolumes](#) (includeVirtualVolumes=false)
- [ListVolumeAccessGroups](#)
- [ListStorageContainers](#)
- [ListQoS Policies](#)
- [GetSnmppInfo](#)
- [GetNtpInfo](#)
- [ListVirtualNetworks](#)
- [ListClusterAdmins](#)
- [ListSchedules](#)
- [ListSnapMirrorEndpoints](#)
- [GetFeatureStatus](#)
- [GetLdapConfiguration](#)
- [GetRemoteLoggingHosts](#)
- [GetDefaultQoS](#)
- [GetVolumeAccessGroupLunAssignments](#)

相关参考

[GetClusterStructure](#) on page 317

您可以使用 `GetClusterStructure` 方法备份当前存储集群配置信息。如果在运行此方法时更改了存储集群配置，则此配置备份的内容将不可预测。您可以将此数据保存到一个文本文件中，然后在发生灾难时，在其他集群上或同一集群上还原此数据。

[SetClusterStructure](#) on page 334

您可以使用 `SetClusterStructure` 方法从备份中还原存储集群配置信息。调用此方法时，您可以通过 `params` 参数传递包含要还原的配置信息的 `clusterStructure` 对象。

drive

`drive` 对象包含有关集群的活动节点中各个驱动器的信息。此对象包含已作为卷元数据或块驱动器添加的驱动器以及尚未添加并且可用的驱动器的相关详细信息。可以使用 `ListDrives` API 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>attributes</code>	名称-值对的列表，采用 JSON 对象格式。此对象始终为空，并且无法修改。	JSON 对象
容量	驱动器的总容量，以字节为单位。	整型
<code>chassisSlot</code>	对于 HCI 平台，此值为该驱动器所位于的服务器机箱中的节点号和插槽编号。对于存储平台，插槽编号为“slot”整数的字符串表示形式。	字符串
驱动程序详细信息	如果驱动器的状态为“故障”，则此字段将提供有关驱动器标记为故障的详细信息。	字符串
<code>driveID</code>	此驱动器的 ID。	整型
<code>DriveSecurityFault FaultStatut罪</code>	如果启用或禁用驱动器安全保护失败，则说明其失败的原因。如果该值为“无”，则不会出现故障。	字符串
密钥 ID	密钥提供程序用来获取验证密钥以解锁此驱动器的密钥 ID。	UUID
<code>keyProviderID</code>	标识用于解锁此驱动器的验证密钥的提供者。	整型
<code>nodeID</code>	包含此驱动器的节点的 ID。	整型
<code>segmentFileSize</code>	驱动器的分段文件大小（以字节为单位）。	整型
<code>serial</code>	驱动器序列号。	字符串
<code>slot</code>	该驱动器所位于的服务器机箱的插槽编号，或者为 -1（如果 SATA DIMM 设备用于内部元数据驱动器）。	整型
<code>status</code>	驱动器的状态。可能值： <ul style="list-style-type: none"> <code>available</code>：可用驱动器。 <code>active</code>：活动驱动器。 <code>erasing</code>：正在安全擦除驱动器。将永久删除该驱动器上的任何数据。 <code>failed</code>：驱动器已发生故障。先前存在于该驱动器上的任何数据已迁移到集群中的其他驱动器。 <code>removing</code>：正在删除驱动器。先前存在于该驱动器上的任何数据将迁移到集群中的其他驱动器。 	字符串
<code>type</code>	驱动器的类型。可能值： <ul style="list-style-type: none"> <code>volume</code>：存储卷元数据。 <code>block</code>：存储块数据。 <code>unknown</code>：驱动器尚未处于活动状态，驱动器类型有待确定。 	字符串

名称	说明	类型
usableCapacity	驱动器的可用容量（以字节为单位）。	整型

相关参考

[ListDrives](#) on page 158

您可以使用 `ListDrives` 方法列出集群的活动节点中的驱动器。此方法返回已作为卷元数据或块驱动器添加的驱动器，以及尚未添加但可用的驱动器。

driveStats

`driveStats` 对象包含单个驱动器的活动测量结果概要。可以使用 API 方法 `GetDriveStats` 检索测量结果信息。

对象成员

此对象包含以下成员：

名称	说明	类型
activeSessions	当前正在使用该驱动器的 iSCSI 会话的数量（仅对元数据驱动器存在）。	整型
driveID	集群中的驱动器的唯一 ID。	整型
failedDieCount	已发生故障的驱动器硬件元件的数量。	整型
lifeRemainingPercent	驱动器介质损耗指标。	整型
lifetimeReadBytes	在驱动器的生命周期内从该驱动器读取的总字节数。	整型
lifetimeWriteBytes	在驱动器的生命周期内写入到该驱动器的总字节数。	整型
powerOnHours	该驱动器已打开电源的小时数。	整型
readBytes	由于客户端操作而从该驱动器读取的总字节数。	整型
readOps	由于客户端操作而在该驱动器上执行的读取操作的总数。	整型
reallocatedSectors	此驱动器中已替换的坏扇区的数量。	整型
reserveCapacityPercent	驱动器的可用预留容量。	整型
timestamp	当前时间（UTC+0 格式）。	ISO 8601 日期字符串
totalCapacity	驱动器的总容量，以字节为单位。	整型
uncorrectableErrors	来自驱动器中的自我监控、分析和报告技术 (Self-Monitoring, Analysis and Reporting Technology, SMART) 监控系统的“报告无法纠正的错误”值。	整型
usedCapacity	驱动器的已用容量，以字节为单位。	整型
usedMemory	托管此驱动器的节点当前使用的内存量。	整型
writeBytes	由于客户端活动而写入到该驱动器的总字节数。	整型
writeOps	由于客户端活动而对该驱动器执行的写入操作的总数。	整型

相关参考

[GetDriveStats](#) on page 157

您可以使用 `GetDriveStats` 方法获取单个驱动器的活动测量结果概要。值从驱动器添加到集群中之后开始累计。部分值特定于块驱动器。运行此方法时，将返回块或元数据驱动器类型的统计数据。

error

`error` 对象包含错误代码和消息（如果运行方法调用期间发生错误）。系统生成的所有错误的错误代码均为 500。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>code</code>	用于标识错误的数字代码。系统生成的所有错误均会返回代码 500。	整型
<code>name</code>	发生的特定错误的唯一标识符。虽然每个方法会返回一组已记录的错误，但您还应准备好处理无法识别的错误。	字符串
<code>message</code>	错误说明以及可能的其他详细信息。	字符串

event

`event` 对象包含 API 方法调用期间或系统执行操作期间发生的事件的详细信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>details</code>	有关事件的附加信息。	JSON 对象
<code>driveID</code>	报告故障的驱动器的驱动器 ID。如果不适用，则为 0。	整型
<code>driveID</code>	报告故障的驱动器的驱动器 ID 列表。如果不适用，则为一个空列表。	整型数组
<code>eventID</code>	与每个事件关联的唯一 ID。	整型
<code>eventInfoType</code>	故障的类型。	字符串
<code>message</code>	已发生的事件的字符串说明。	字符串
<code>nodeID</code>	报告故障的节点的节点 ID。如果不适用，则为 0。	整型
<code>serviceID</code>	报告故障的服务的服务 ID。如果不适用，则为 0。	整型
<code>severity</code>	事件报告的严重性。	整型
<code>timeOfPublish</code>	集群的事件日志收到该事件的时间（UTC+0 格式）。	ISO 8601 日期字符串
<code>timeOfReport</code>	集群中发生该事件的时间（UTC+0 格式）。	ISO 8601 日期字符串

注：如果发生了事件，而此事件无法立即发布，则 `timeOfReport` 和 `timeOfPublish` 可能稍有差别。

事件类型

以下列表介绍了 `eventInfoType` 成员可能包含的事件类型：

- `apiEvent`：通过 API 或 Web UI 启动的事件，用于修改设置。
- `binAssignmentsEvent`：与向内部容器分配数据相关的事件。

- `binSyncEvent`: 与在块服务之间重新分配数据相关的事件。
- `bsCheckEvent`: 与块服务检查相关的事件。
- `bsKillEvent`: 与块服务终止相关的事件。
- `bulkOpEvent`: 适用于整个卷的事件，例如卷备份、还原、快照或克隆。
- `cloneEvent`: 与卷克隆相关的事件。
- `clusterMasterEvent`: 集群配置更改事件，例如添加或删除节点。
- `dataEvent`: 与读写数据相关的事件。
- `dbEvent`: 与集合节点数据库相关的事件。
- `driveEvent`: 与驱动器操作相关的事件。
- `encryptionAtRestEvent`: 与存储数据加密相关的事件。
- `ensembleEvent`: 与集合大小增加或减少相关的事件。
- `fibreChannelEvent`: 与光纤通道节点配置或连接相关的事件。
- `gcEvent`: 与垃圾收集相关的事件。此类进程每 60 分钟运行一次以回收块驱动器上的存储。
- `ieEvent`: 与内部系统错误相关的事件。
- `installEvent`: 与在待定存储节点上自动安装软件相关的事件。
- `iSCSIEvent`: 与 iSCSI 连接或配置问题相关的事件。
- `limitEvent`: 与即将达到所允许上限数的帐户或集群中的卷或虚拟卷数量相关的事件。
- `networkEvent`: 与虚拟网络相关的事件。
- `platformHardwareEvent`: 与在硬件设备上检测到的问题相关的事件。
- `remoteClusterEvent`: 与远程集群配对相关的事件。
- `schedulerEvent`: 与计划快照相关的事件。
- `serviceEvent`: 与系统服务状态相关的事件。
- `statEvent`: 与系统统计信息相关的事件。
- `sliceEvent`: 与元数据存储相关的事件。
- `snmpTrapEvent`: 与 SNMP 陷阱相关的事件。
- `tsEvent`: 系统传输服务事件。
- `unexpectedException`: 与意外错误相关的事件。
- `vasaProviderEvent`: 与 VMware VASA Provider 相关的事件。

相关参考

[ListEvents](#) on page 133

您可以使用 `ListEvents` 方法列出在集群上检测到的事件，按从旧到新的顺序排列。

fault

`fault` 对象包含有关在集群中检测到的故障的信息。`ListClusterFaults` 方法将返回集群故障信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>clusterFaultID</code>	与每个集群故障关联的唯一 ID。	整型
<code>code</code>	检测到的特定故障的故障代码。有关更多详细信息，请参见“集群故障代码”。	字符串
<code>data</code>	其他特定于故障的信息。	JSON 对象
<code>date</code>	当前时间（UTC+0 格式）。	ISO 8601 字符串

名称	说明	类型
details	故障说明以及其他详细信息。	字符串
driveID	驱动器 ID 列表中的第一个驱动器 ID。如果驱动器 ID 列表为空（表示未返回任何涉及驱动器的故障），则此值为 0。	整型
driveID	与此故障关联的驱动器的驱动器 ID 值列表。对于涉及驱动器的故障，将返回此成员。如果没有任何故障，则为空数组。	整型数组
nodeHardwareFaultID	分配给集群中的硬件故障的标识符。	整型
nodeID	与此故障关联的节点的节点 ID。对于 node 和 drive 故障，将返回此成员；对于其他故障，将设置为 0。	整型
resolved	故障的解决状态。可能值： <ul style="list-style-type: none"> • true: 不再检测到此故障。 • false: 此故障仍然存在。 	布尔型
resolvedDate	解决故障的日期和时间。	ISO 8601 字符串
serviceID	与故障关联的服务。如果故障不与任何服务关联，则此值为“0”（零）。	整型
severity	故障的严重性。可能值： <ul style="list-style-type: none"> • warning: 次要问题。在此严重性级别，集群正常运行且允许升级。 • error: 一般不会影响服务的故障（但可能会降低性能或丢失 HA）。某些功能可能已禁用。 • critical: 影响服务的严重故障系统无法处理 API 请求或客户端 I/O，可能会丢失数据。 • bestPractice: 由非最佳系统配置触发的故障。 	字符串
type	故障的类型。可能值： <ul style="list-style-type: none"> • node: 影响整个节点的故障。 • drive: 影响单个驱动器的故障。 • cluster: 影响整个集群的故障。 • service: 影响集群上某个服务的故障。 • volume: 影响单个卷的故障。 	字符串

相关参考

[ListClusterFaults](#) on page 131

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。

[集群故障代码](#) on page 482

如果存储集群出现可能需要引起管理员注意的错误或状况，它会生成集群故障。您可以使用 `ListClusterFaults` 方法检索存储集群中当前已解决和未解决的故障列表。

fibresChannelPort

`fibresChannelPort` 对象包含某个节点上或集群的整个节点中各个端口的信息。可以使用 `ListNodeFibreChannelPortInfo` 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
固件	光纤通道端口上安装的固件版本。	整型
hbaPort	各个主机总线适配器 (HBA) 端口的 ID。	整型
model	端口上的 HBA 型号。	字符串
nPortID	唯一的端口节点 ID。	字符串
pciSlot	包含光纤通道节点机箱中的 PCI 卡的插槽。	整型
serial	光纤通道端口上的序列号。	字符串
speed	端口上的 HBA 的速度。	字符串
state	可能值： <ul style="list-style-type: none"> 未知 NotPresent 联机 脱机 Blocked Bypassed Diagnostics Linkdown 错误 Loopback Deleted 	字符串
switchWwn	光纤通道交换机端口的全球通用名称。	字符串
wwnn	HBA 节点的全局通用节点名称。	字符串
wwpn	分配给 HBA 的物理端口的全球通用端口名称。	字符串

相关参考

[ListNodeFibreChannelPortInfo](#) on page 168

您可以使用 `ListNodeFibreChannelPortInfo` 方法列出有关节点上的光纤通道端口的信息。

fipsErrorNodeReport

`fipsErrorNodeReport` 对象包含在您使用 `GetFipsReport` 方法查询有关 FIPS 140-2 支持的信息时未使用此信息进行响应的每个节点的错误信息。

对象成员

此对象包含以下成员：

名称	说明	类型
nodeID	未响应的节点的 ID。	整型
error	一个 JSON 对象，其中包含错误信息。	JSON 对象

fipsNodeReport

`fipsNodeReport` 对象包含有关存储集群中某个节点的 FIPS 140-2 支持的信息。可以使用 `GetFipsReport` 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
nodeID	报告此信息的节点的 ID。	整型
fipsDrives	是否已为此节点启用 FIPS 140-2 驱动器加密。可能值： <ul style="list-style-type: none"> None：节点不支持 FIPS 驱动器加密。 Partial：节点支持 FIPS 驱动器加密，但并非所有现有驱动器都为支持 FIPS 的驱动器。 Ready：节点支持 FIPS 驱动器加密，并且所有现有驱动器均为支持 FIPS 的驱动器，或者不存在任何驱动器。 	FipsDrivesStatusType
httpsEnabled	是否为此节点启用 FIPS 140-2 HTTPS 加密。可能值： <ul style="list-style-type: none"> true：启用 false：禁用 	布尔型

fipsReport

`fipsReport` 对象包含有关存储集群中所有节点的 FIPS 140-2 支持的信息。可以使用 `GetFipsReport` 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
nodes	报告存储集群中每个节点的 FIPS 140-2 支持状态。	fipsNodeReport
errorNodes	未使用 FIPS 140-2 支持状态进行响应的每个节点的错误信息。	fipsErrorNodeReport

groupSnapshot

groupSnapshot 对象包含有关一组卷的快照的信息。您可以使用 ListGroupSnapshots API 方法检索组快照信息。

对象成员

此对象包含以下成员：

名称	说明	类型
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
createTime	组快照的创建日期和时间（UTC+0 格式）。	ISO 8601 日期字符串
enableRemoteReplication	确定是否为此快照启用远程复制。	布尔型
groupSnapshotID	组快照的唯一 ID。	整型
groupSnapshotUUID	组快照的 UUID。	字符串
members	一个对象数组，包含有关组快照中每个成员的信息。	<i>snapshot</i> 数组
name	组快照的名称；或者，如果未指定，则会使用快照的创建日期和时间（UTC 格式）。	字符串或 ISO 8601 日期字符串
remoteStatuses	一个数组，其中包含目标集群上每个远程快照的通用标识符和复制状态（与从源集群看到的信息一致）。	<i>remoteClusterSnapshotStatus</i> 数组
status	快照的当前状态。可能值： <ul style="list-style-type: none"> Unknown：获取此快照的状态时出错。 Preparing：正在准备此快照以供使用，此快照目前不可写入。 RemoteSyncing：正在从远程集群复制此快照。 Done：此快照已完成准备或复制，现可供使用。 Active：此快照为活动分支。 Cloning：此快照参与 CopyVolume 操作。 	字符串

相关参考

[ListGroupSnapshots](#) on page 453

您可以使用 ListGroupSnapshots 方法返回已创建的所有组快照的相关信息。

hardwareInfo

hardwareInfo 对象包含集群中每个节点的硬件和状态的详细信息。可以使用 GetHardwareInfo API 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
boardSerial	DMI 板序列号。	字符串
bus	主板介质总线信息。	JSON 对象

名称	说明	类型
chassisSerial	机箱的序列号。	字符串
driveHardware	节点中每个驱动器的信息的列表。	JSON 对象数组
fibreChannelPorts	节点上的光纤通道端口的列表。	整型数组
hardwareConfig	主板外设配置信息。	JSON 对象
kernelCrashDumpState	操作系统内核的崩溃转储配置。	字符串
memory	固件和系统内存硬件信息。	JSON 对象
network	每个节点的网络接口的硬件的描述。	JSON 对象
networkInterfaces	节点的网络接口的状态。	JSON 对象
nodeSlot	对于 HCI 平台，与此节点的机箱插槽对应的字母为“A”、“B”、“C”或“D”。对于存储平台，此值为 null。	字符串
nvram	节点的 NVRAM 统计信息。	JSON 对象
origin	主板的供应商。	字符串
platform	机箱平台的描述。	JSON 对象
serial	产品的序列号。	字符串
存储	存储控制器信息。	JSON 对象
systemMemory	操作系统内存使用量和性能信息。	JSON 对象
system	节点机箱的类型。	JSON 对象
uuid	节点的唯一 ID。	UUID

相关参考

[GetHardwareInfo](#) on page 209

您可以使用 `GetHardwareInfo` 方法获取单个节点的硬件信息和状态。硬件信息通常包含制造商、供应商、版本、驱动器和其他相关的标识信息。

host (虚拟卷)

`host` 对象包含有关虚拟卷主机的信息。您可以使用 `ListVirtualVolumeHosts` 方法为所有虚拟卷主机获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
bindings	描述虚拟卷主机绑定的对象列表。	整型数组
clusterID	该主机关联的集群的唯一 ID。	UUID
hostAddress	虚拟卷主机的 IP 地址或 DNS 名称。	字符串
initiatorNames	虚拟卷主机的启动程序 IQN 的列表。	字符串数组
virtualVolumeHostID	此虚拟卷主机的唯一 ID。	UUID

名称	说明	类型
visibleProtocolEndpointIDs	此主机可识别的协议端点 ID 的列表。	UUID 数组

相关参考

[ListVirtualVolumeHosts](#) on page 476

您可以使用 `ListVirtualVolumeHosts` 方法获取集群已知的所有虚拟卷主机的列表。虚拟卷主机是通过 VASA API 提供程序启动会话的 VMware ESX 主机。

IDPConfigInfo

`idpConfigInfo` 该对象包含有关第三方身份认证提供者（IDP）的配置和集成详细信息。

对象成员

此对象包含以下成员：

名称	说明	类型
enabled	指定是否启用此第三方 IDPConfiguration。	布尔型
IDPConfigurationID	第三方 IDP 配置的 UUID。	UUID
IDPMetadata	用于配置和集成 SAML 2.0 单点登录的详细信息 的元数据。	字符串
IDPName	用于检索 SAML 2.0 单点登录的 IDP 提供者的 名称。	字符串
ServiceProviderCertificate	用于与此 IDP 通信的 PEM 格式 Base64 编码的 PKCS#10X.509 证书。	字符串
Spmetadataurl 的文件	用于从集群中检索服务提供商（SP）元数据 的 URL，以提供给 IDP 以建立信任关系。	字符串

initiator

`initiator` 对象包含有关 iSCSI 或光纤通道启动程序的信息。`initiator` 对象可以包含 IQN 或 WWPN 标识符。可以使用 `ListInitiators` 方法来获取系统上所有已知启动程序的列表。

对象成员

此对象包含以下成员：

名称	说明	类型
alias	分配给该启动程序的友好名称（如 果有）。	字符串
attributes	分配给该启动程序的一组 JSON 属 性。如果未分配任何属性，则为 空。	JSON 对象
chapUsername	此启动程序的唯一 CHAP 用户名。	字符串
initiatorID	此启动程序的数字标识符。	整型
initiatorName	启动程序名称（采用 IQN 或 WWPN 格式）。	字符串

名称	说明	类型
initiatorSecret	用于对此启动程序进行身份验证的 CHAP 密钥。	字符串
requireChap	如果需要对此启动程序使用 CHAP，则为 True。	布尔型
targetSecret	用于对目标进行身份验证的 CHAP 密钥（使用相互 CHAP 身份验证时）。	字符串
volumeAccessGroups	此启动程序所属的卷访问组 ID 列表。	整型数组

相关参考

[ListInitiators](#) on page 393

您可以使用 `ListInitiators` 方法获取启动程序 IQN 或全球通用端口名称 (World Wide Port Name, WWPN) 的列表。

keyProviderKmpip

`keyProviderKmpip` 对象用于描述密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥提供程序。密钥提供程序提供了一种用于检索集群功能（例如空闲加密）所使用的身份验证密钥的机制，可从此密钥提供程序检索此密钥。

对象成员

此对象包含以下成员：

名称	说明	类型
keyProviderID	KMIP 密钥提供程序的 ID。此 ID 是在创建此密钥提供程序期间由集群分配的唯一值，无法更改。	整型
keyProviderIsActive	如果此 KMIP 密钥提供程序处于活动状态，则为 True。如果存在已创建但尚未删除的待处理密钥，并因此而假定此密钥正在使用中，则会将此提供程序视为处于活动状态。	布尔型
keyProviderName	KMIP 密钥提供程序的名称。	字符串
keyServerIDs	与此提供程序关联的密钥服务器 ID。必须先添加此服务器，然后此提供程序才能处于活动状态。在此提供程序处于活动状态期间，无法删除此服务器。每个提供程序仅支持一个服务器 ID。	整型数组
kmpipCapabilities	此 KMIP 密钥提供程序的功能，包括底层库、FIPS 合规性、SSL 提供程序等详细信息。	字符串

keyServerKmpip

`keyServerKmpip` 对象用于描述密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器，可从此服务器检索用于集群功能（例如空闲加密）的身份验证密钥。

对象成员

此对象包含以下成员：

名称	说明	类型
keyProviderID	如果已将此 KMIP 密钥服务器分配给一个提供程序，则此成员包含将其分配到的 KMIP 密钥提供程序的 ID。否则，此成员为 null。	整型
keyServerID	KMIP 密钥服务器的 ID。此 ID 是在创建此密钥服务器期间由集群分配的唯一值。无法更改此值。	整型
kmipAssignedProviderIsActive	如果已将此 KMIP 密钥服务器分配给一个提供程序（keyProviderID 不为空），则此成员可指示此提供程序是否处于活动状态（提供当前正在使用的密钥）。否则，此成员为 null。	布尔型
kmipCaCertificate	外部密钥服务器根 CA 的公共密钥证书。在 TLS 通信中会使用此证书来验证外部密钥服务器提供的证书。对于各个服务器使用不同 CA 的密钥服务器集群，此成员包含一个串联字符串，其中会列出所有这些 CA 的根证书。	字符串
kmipClientCertificate	Element 存储 KMIP 客户端使用的 PEM 格式 Base64 编码 PKCS#10 X.509 证书。	字符串
kmipKeyServerHostnames	与此 KMIP 密钥服务器关联的主机名或 IP 地址。	字符串数组
kmipKeyServerName	此 KMIP 密钥服务器的名称。此名称仅用于显示，不需要唯一。	字符串
kmipKeyServerPort	与此 KMIP 密钥服务器关联的端口号（通常为 5696）。	整型

ldapConfiguration

ldapConfiguration 对象包含存储系统上 LDAP 配置的相关信息。可以使用 API 方法 GetLdapConfiguration 检索 LDAP 信息。

对象成员

此对象包含以下成员：

名称	说明	类型
authType	标识要使用的用户身份验证方法。可能值： <ul style="list-style-type: none"> DirectBind SearchAndBind 	字符串
enabled	标识是否针对 LDAP 配置了系统。可能值： <ul style="list-style-type: none"> true false 	布尔型
groupSearchBaseDN	树中用作组搜索起点的基础 DN（系统将从此处开始执行子树搜索）。	字符串
groupSearchCustomFilter	所使用的自定义搜索筛选器。	字符串

名称	说明	类型
groupSearchType	控制所使用的组搜索筛选器。可能值： <ul style="list-style-type: none"> NoGroups: 无组支持。 ActiveDirectory: 用户的所有 AD 组的嵌套成员资格。 MemberDN: MemberDN 样式的组（单层）。 	字符串
searchBindDN	为了对用户执行 LDAP 搜索而登录时所使用的完全限定 DN（需要对 LDAP 目录具有读取访问权限）。	字符串
serverURIs	LDAP 服务器 URI 列表（例如 ldap://1.2.3.4 和 ldaps://1.2.3.4:123），以英文逗号分隔。	字符串
userDNTemplate	一个字符串，用于形成完全限定用户 DN。	字符串
userSearchBaseDN	树中用作搜索起点的基础 DN（将从此处执行子树搜索）。	字符串
userSearchFilter	所使用的 LDAP 筛选器。	字符串

相关参考

[GetLdapConfiguration](#) on page 176

您可以使用 GetLdapConfiguration 方法获取集群上当前处于活动状态的 LDAP 配置。

loggingServer

loggingServer 对象包含有关为存储集群配置的任何日志记录主机的信息。您可以使用 GetRemoteLoggingHosts 确定当前的日志记录主机，然后使用 SetRemoteLoggingHosts 设置所需的当前和新日志记录主机列表。

对象成员

此对象包含以下成员：

名称	说明	类型
host	日志服务器的 IP 地址。	字符串
port	用于与日志服务器通信的端口号。	整型

network（绑定接口）

network（绑定接口）对象包含存储节点上绑定网络接口的配置信息。您可以使用 GetConfig 和 GetNetworkConfig 方法为存储节点获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
address	节点上分配给此接口的 IPv4 地址。	字符串
addressV6	分配给节点上绑定 1G 接口的 IPv6 管理地址。	字符串
bond-downdelay	检测到链路故障后，在禁用从属节点之前等待的时间。	字符串
bond-fail_over_mac	网络接口的 MAC 地址的配置。	字符串
bond-miimon	检查 MII 链路状态是否存在链路故障的频率。	整型

名称	说明	类型
bond-mode	绑定模式。可能值： <ul style="list-style-type: none"> ActivePassive（默认） ALB LACP（建议） 	字符串
bond-primary_reselect	指定何时选择主绑定从属接口作为活动从属接口。可能值： <ul style="list-style-type: none"> 始终 Better Failure 	字符串
bond-slaves	绑定从属接口的列表。	字符串
bond-lacp_rate	如果绑定模式为 LACP，则此速率可能会更改为以下项之一： <ul style="list-style-type: none"> LACP Fast（默认） LACP Slow 	整型
bond-updelay	检测到链路后，在启用从属接口之前等待的时间。	整型
dns-nameservers	用于域名服务的地址列表（以英文逗号或空格分隔）。	字符串
dns-search	DNS 搜索域列表（以空格或英文逗号分隔）。	字符串
family	接口经过配置后可使用的地址系列。对于 IPv4，目前支持“inet”。	字符串
gateway	用于从本地网络发送流量的 IPv4 路由器网络地址。	字符串
gatewayV6	用于从本地绑定 1G 网络发送流量的 IPv6 路由器网络地址。	字符串
ipV6PrefixLength	绑定 1G 网络上 IPv6 流量中类型为“net”的静态路由的子网前缀长度。	字符串
macAddress	分配给该接口并通过网络检测到的实际 MAC 地址。	字符串
macAddressPermanent	制造商分配给该接口的不可变 MAC 地址。	字符串
method	用于配置接口的方法。可能值： <ul style="list-style-type: none"> Loopback：用于定义 IPv4 环回接口。 manual：用于定义未自动配置的接口。 dhcp：可用于通过 DHCP 获取 IP 地址。 static：用于使用静态分配的 IPv4 地址定义以太网接口。 	字符串
mtu	接口可以传输的最大数据包大小（以字节为单位）。必须大于或等于 1500；最多支持 9000。	字符串
网络掩码	用于为接口指定子网的位掩码。	字符串
network	基于网络掩码指示 IP 地址范围的开始位置。	字符串
路由	要应用于路由表的路由字符串数组，以逗号分隔。	字符串数组

名称	说明	类型
status	接口状态。可能值： <ul style="list-style-type: none"> Down: 接口未处于活动状态。 Up: 接口已准备就绪，但无链路。 UpAndRunning: 接口已准备就绪，且已建立链路。 	字符串
symmetricRouteRules	节点上配置的对称路由规则。	字符串数组
upAndRunning	指示接口是否已准备就绪且有链路。	布尔型
virtualNetworkTag	接口的虚拟网络标识符（VLAN 标记）。	字符串

成员可修改性和节点状态

下表说明了在每种可能的节点状态下是否可修改相应的对象参数。

成员名称	可用状态	待定状态	活动状态
address	是	是	否
addressV6	是	是	否
bond-downdelay	由系统配置	不适用	不适用
bond-fail_over_mac	由系统配置	不适用	不适用
bond-miimon	由系统配置	不适用	不适用
bond-mode	是	是	是
bond-primary_reselect	由系统配置	不适用	不适用
bond-slaves	由系统配置	不适用	不适用
bond-lacp_rate	是	是	是
bond-updelay	由系统配置	不适用	不适用
dns-nameservers	是	是	是
dns-search	是	是	是
family	否	否	否
gateway	是	是	是
gatewayV6	是	是	是
ipV6PrefixLength	是	是	是
macAddress	由系统配置	不适用	不适用
macAddressPermanent	由系统配置	不适用	不适用
method	否	否	否
mtu	是	是	是
netmask	是	是	是
network	否	否	否
routes	是	是	是
status	是	是	是

成员名称	可用状态	待定状态	活动状态
symmetricRouteRules	由系统配置	不适用	不适用
upAndRunning	由系统配置	不适用	不适用
virtualNetworkTag	是	是	是

相关参考

[GetConfig](#) on page 205

您可以使用 `GetConfig` API 方法获取节点的所有配置信息。此 API 方法包含可通过 `GetClusterConfig` 和 `GetNetworkConfig` API 方法获得的相同信息。

[GetNetworkConfig](#) on page 217

您可以使用 `GetNetworkConfig` 方法显示节点的网络配置信息。

network（所有接口）

`network`（所有接口）对象用于收集有关存储节点的网络接口配置的信息。您可以使用 `GetConfig` 和 `GetNetworkConfig` 方法为存储节点获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
Bond10G	绑定 10G 绑定接口的配置信息。	network （绑定接口）
Bond1G	绑定 1G 绑定接口的配置信息。	network （绑定接口）
eth0-5	存储节点中的每个以太网接口对应一个对象，用于说明此接口的配置信息。这些对象的编号为 0 到 5，与接口名称匹配。	network （以太网接口）
lo	环回接口的配置信息。	network （本地接口）

相关参考

[GetConfig](#) on page 205

您可以使用 `GetConfig` API 方法获取节点的所有配置信息。此 API 方法包含可通过 `GetClusterConfig` 和 `GetNetworkConfig` API 方法获得的相同信息。

[GetNetworkConfig](#) on page 217

您可以使用 `GetNetworkConfig` 方法显示节点的网络配置信息。

network（以太网接口）

`network`（以太网接口）对象包含单个以太网接口的配置信息。您可以使用 `GetConfig` 和 `GetNetworkConfig` 方法为存储节点获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
bond-master	指定此物理接口作为绑定从属接口加入的绑定接口。	字符串
family	接口经过配置后可使用的地址系列。对于 IPv4，目前支持“inet”。	字符串

名称	说明	类型
macAddress	分配给该接口并通过网络检测到的实际 MAC 地址。	字符串
macAddressPermanent	制造商分配给该接口的不可变 MAC 地址。	字符串
method	用于配置接口的方法。可能值： <ul style="list-style-type: none"> • loopback: 用于定义 IPv4 环回接口。 • manual: 用于定义未自动配置的接口。 • dhcp: 可用于通过 DHCP 获取 IP 地址。 • static: 用于使用静态分配的 IPv4 地址定义以太网接口。 	字符串
status	接口状态。可能值： <ul style="list-style-type: none"> • Down: 接口未处于活动状态。 • Up: 接口已准备就绪，但无链路。 • UpAndRunning: 接口已准备就绪，且已建立链路。 	字符串
upAndRunning	(布尔值) 指示接口是否已准备就绪且具有链路。	布尔型

成员可修改性和节点状态

下表说明了在每种可能的节点状态下是否可修改相应的对象参数。

参数名称	可用状态	待定状态	活动状态
bond-master	否	否	否
family	否	否	否
macAddress	由系统配置	不适用	不适用
macAddressPermanent	由系统配置	不适用	不适用
method	否	否	否
status	是	是	是
upAndRunning	由系统配置	不适用	不适用

相关参考

[GetConfig](#) on page 205

您可以使用 `GetConfig` API 方法获取节点的所有配置信息。此 API 方法包含可通过 `GetClusterConfig` 和 `GetNetworkConfig` API 方法获得的信息。

[GetNetworkConfig](#) on page 217

您可以使用 `GetNetworkConfig` 方法显示节点的网络配置信息。

network (本地接口)

`network` (本地接口) 对象包含存储节点上的本地网络接口 (例如环回接口) 的配置信息。您可以使用 `GetConfig` 和 `GetNetworkConfig` 方法为存储节点获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
family	接口经过配置后可使用的地址系列。对于 IPv4，目前支持“inet”。	字符串
macAddress	分配给该接口并通过网络检测到的实际 MAC 地址。	字符串
macAddressPermanent	制造商分配给该接口的不可变 MAC 地址。	字符串
method	用于配置接口的方法。可能值： <ul style="list-style-type: none"> loopback：用于定义 IPv4 环回接口。 manual：用于定义未自动配置的接口。 dhcp：可用于通过 DHCP 获取 IP 地址。 static：用于使用静态分配的 IPv4 地址定义以太网接口。 	字符串
status	接口状态。可能值： <ul style="list-style-type: none"> Down：接口未处于活动状态。 Up：接口已准备就绪，但无链路。 UpAndRunning：接口已准备就绪，且已建立链路。 	字符串
upAndRunning	指示接口是否已准备就绪且有链路。	布尔型

成员可修改性和节点状态

下表说明了在每种可能的节点状态下是否可修改相应的对象参数。

参数名称	可用状态	待定状态	活动状态
family	否	否	否
macAddress	由系统配置	不适用	不适用
macAddressPermanent	由系统配置	不适用	不适用
method	否	否	否
status	是	是	是
upAndRunning	由系统配置	不适用	不适用

相关参考

[GetConfig](#) on page 205

您可以使用 `GetConfig` API 方法获取节点的所有配置信息。此 API 方法包含可通过 `GetClusterConfig` 和 `GetNetworkConfig` API 方法获得的信息。

[GetNetworkConfig](#) on page 217

您可以使用 `GetNetworkConfig` 方法显示节点的网络配置信息。

network (SNMP)

SNMP network 对象包含集群节点的 SNMP v3 配置的相关信息。

对象成员

此对象包含以下成员：

名称	说明	类型
访问	SNMP 信息请求允许的访问类型。可能值： <ul style="list-style-type: none"> ro: 只读访问权限。 rw: 读写访问权限。 rosys: 对有限的系统信息集的只读访问权限。 	字符串
cidr	CIDR 网络掩码。此网络掩码必须是一个大于或等于 0 且小于或等于 32 的整数。此网络掩码还不能等于 31。	整型
community	SNMP 社区字符串。	字符串
network	此成员以及 cidr 成员控制访问权限和社区字符串适用的网络。特殊值 “default” 用于指定将应用到所有网络的条目。如果此成员为主机名或 “default”，将忽略 CIDR 掩码。	字符串

相关参考

[GetSnmplibInfo](#) on page 326

您可以使用 `GetSnmplibInfo` 方法获取当前简单网络管理协议 (Simple Network Management Protocol, SNMP) 配置信息。

networkInterface

`networkInterface` 对象包含存储节点上单个网络接口的配置信息。

对象成员

此对象包含以下成员：

名称	说明	类型
address	接口的 IPv4 管理地址。	字符串
addressV6	接口的 IPv6 管理地址。	字符串
broadcast	接口的广播地址。	字符串
macAddress	接口的 MAC 地址。	字符串
mtu	接口的最大传输单元（以字节为单位）。	整型
name	接口的名称。	字符串
命名空间	是否为此接口分配了虚拟网络命名空间。	布尔型
网络掩码	接口的子网掩码。	字符串
status	接口的运行状态。	字符串
type	接口的类型（绑定主接口、绑定从属接口等）。	字符串
virtualNetworkTag	分配给虚拟网络上的接口的 VLAN ID。	整型

node

`node` 对象包含集群中每个节点的相关信息。可以使用 `ListActiveNodes` 和 `ListAllNodes` 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
associatedFServiceID	节点的光纤通道服务 ID。如果节点不是光纤通道节点，则为“0”。	整型
associatedMasterServiceID	节点的主服务 ID。	整型
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
chassisName	用于唯一标识机箱；对于一个机箱中的所有节点，此值均相同。	字符串
cip	分配给节点的集群 IP 地址。	字符串
cipi	用于集群通信的网络接口。	字符串
CustomProtectionDomainName	唯一标识自定义保护域。对于给定自定义保护域中所有机箱中的所有存储节点，此名称相同。	字符串
fibreChannelTargetPortGroup	与该节点关联的目标组。如果节点不是光纤通道节点，则为“null”。	整型
mip	用于进行节点管理的 IP 地址。	字符串
mipi	用于进行节点管理的网络接口。	字符串
name	节点的主机名。	字符串
nodeID	该节点的节点 ID。	整型
nodeSlot	对于 HCI 平台，与此节点的机箱插槽对应的字母为“A”、“B”、“C”或“D”。对于存储平台，此值为 null。	字符串
platformInfo	节点的硬件信息。成员： <ul style="list-style-type: none"> chassisType: 节点的硬件平台。 cpuModel: 硬件平台的 CPU 型号。 nodeMemoryGB: 安装在物理平台中的内存量，以 GB 为单位。 nodeType: 节点型号名称。 platformConfigVersion: 为此节点硬件配置的软件版本。 	JSON 对象
role	节点在集群中的角色。可能值： <ul style="list-style-type: none"> 管理 存储 计算 见证 	
sip	分配给节点的存储 IP 地址。	字符串
sipi	用于存储流量的网络接口。	字符串
softwareVersion	返回节点上当前运行的 Element 软件版本。	字符串
uuid	与该节点关联的通用唯一标识符。	字符串
virtualNetworks	包含虚拟网络 IP 地址和 ID 的对象。	<i>virtualNetwork</i> 数组

相关参考

[ListActiveNodes](#) on page 129

您可以使用 `ListActiveNodes` 方法返回集群中当前处于活动状态的节点的列表。

[ListAllNodes](#) on page 129

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

节点保护域

`nodeProtectionDomains` 该对象包含有关节点标识以及与该节点关联的保护域的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
nodeID	节点的唯一标识符。	整型
域保护	节点所属的保护域列表。	保护域

nodeStats

`nodeStats` 对象包含节点的活动测量结果概要。您可以使用 API 方法 `getNodeStats` 和 `ListNodeStats` 获取部分或所有 `nodeStats` 对象。

对象成员

此对象包含以下成员：

名称	说明	类型
count	<code>nodeStats</code> 对象中的样本总数。	整型
cpu	CPU 使用率 (%)。	整型
cpuTotal	CPU 利用率的单调递增值。	整型
cBytesIn	通过集群接口传入的字节。	整型
cBytesOut	通过集群接口传出的字节。	整型
sBytesIn	通过存储接口传入的字节。	整型
sBytesOut	通过存储接口传出的字节。	整型
mBytesIn	通过管理接口传入的字节。	整型
mBytesOut	通过管理接口传出的字节。	整型
networkUtilizationCluster	集群网络接口的网络接口利用率（以百分比表示）。	整型
networkUtilizationStorage	存储网络接口的网络接口利用率（以百分比表示）。	整型
readLatencyUsecTotal	对节点执行读取操作所用的总时间的单调递增值。	整型
readOps	对节点执行的总读取操作的单调递增值。	整型
ssLoadHistogram	用于展示一段时间内的分区服务负载的直方图数据。	JSON 对象
timestamp	当前时间（UTC+0 格式）。	ISO 8601 日期字符串
usedMemory	内存使用总量（以字节为单位）。	整型
writeLatencyUsecTotal	对节点执行写入操作所用的总时间的单调递增值。	整型

名称	说明	类型
writeOps	对节点执行的总写入操作的单调递增值。	整型

相关参考

[GetNodeStats](#) on page 128

您可以使用 `GetNodeStats` 方法检索单个节点的活动测量结果概要。

[ListNodeStats](#) on page 135

您可以使用 `ListNodeStats` 方法查看存储集群中所有存储节点的活动测量结果概要。

ontapVersionInfo

`ontapVersionInfo` 对象包含有关 `SnapMirror` 关系中 ONTAP 集群的 API 版本的信息。Element Web 界面可使用 `GetOntapVersionInfo` API 方法获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
clientAPIMajorVesion	Element API 客户端正在使用的 ONTAP API 主要版本。	字符串
clientAPIMinorVesion	Element API 客户端正在使用的 ONTAP API 次要版本。	字符串
ontapAPIMajorVersion	ONTAP 系统支持的 API 当前主要版本。	字符串
ontapAPIMinorVesion	ONTAP 系统支持的 API 当前次要版本。	字符串
ontapVersion	ONTAP 集群上运行的当前软件版本。	字符串

pendingActiveNode

`pendingActiveNode` 对象包含当前处于“`pendingActive`”状态（介于待定和活动状态之间）的节点的相关信息。这些节点为当前正还原出厂软件映像的节点。使用 `ListPendingActiveNodes` API 方法可为所有 `pendingActive` 节点返回此信息的列表。

对象成员

此对象包含以下成员：

名称	说明	类型
activeNodeKey	一个允许节点在成功安装软件之后自动加入集群的唯一密钥。	字符串
assignedNodeID	节点的分配节点 ID。	字符串
asyncHandle	可用于查询操作状态的异步方法句柄。	整型
cip	分配给节点的集群 IP 地址。	字符串
mip	分配给节点的管理 IP 地址。	字符串
nodeSlot	对于 HCI 平台，与此节点的机箱插槽对应的字母为“A”、“B”、“C”或“D”。对于存储平台，此值为 <code>null</code> 。	字符串

名称	说明	类型
pendingActiveNodeID	节点的待定节点 ID。	整型
platformInfo	节点的硬件信息。成员： <ul style="list-style-type: none"> chassisType: 节点的硬件平台。 cpuModel: 硬件平台的 CPU 型号。 nodeMemoryGB: 安装在物理平台中的内存量，以 GB 为单位。 nodeType: 节点型号名称。 platformConfigVersion: 为此节点硬件配置的软件版本。 	JSON 对象
role	节点在集群中的角色。可能值： <ul style="list-style-type: none"> 管理 存储 计算 见证 	
sip	分配给节点的存储 (iSCSI) IP 地址。	字符串
softwareVersion	节点上当前运行的 Element 软件版本。	字符串

相关参考

[ListPendingActiveNodes](#) on page 141

您可以使用 `ListPendingActiveNodes` 方法列出集群中处于“PendingActive”状态（介于待定和活动状态之间）的节点。处于此状态的节点正还原出厂映像。

pendingNode

`pendingNode` 对象包含可添加到集群中的节点的相关信息。使用 `ListPendingNodes` API 方法可为所有 `pending` 节点返回此信息的列表。可以使用 `AddNodes` API 方法将列出的任何节点添加到集群中。

对象成员

此对象包含以下成员：

名称	说明	类型
cipi	分配给节点的集群 IP 地址。	字符串
activeNodeKey	一个允许节点在成功安装软件之后自动加入集群的唯一密钥。	字符串
assignedNodeID	节点的分配节点 ID。	字符串
asyncHandle	可用于查询操作状态的异步方法句柄。	整型
chassisName	用于唯一标识机箱；对于一个机箱中的所有节点，此值均相同。	字符串
cip	分配给节点的集群 IP 地址。	字符串
mip	分配给节点的管理 IP 地址。	字符串

名称	说明	类型
nodeSlot	对于 HCI 平台，与此节点的机箱插槽对应的字母为“A”、“B”、“C”或“D”。对于存储平台，此值为 null。	字符串
pendingActiveNodeID	节点的待定节点 ID。	整型
platformInfo	节点的硬件信息。成员： <ul style="list-style-type: none"> chassisType: 节点的硬件平台。 cpuModel: 硬件平台的 CPU 型号。 nodeMemoryGB: 安装在物理平台中的内存量，以 GB 为单位。 nodeType: 节点型号名称。 platformConfigVersion: 为此节点硬件配置的软件版本。 	JSON 对象
role	节点在集群中的角色。可能值： <ul style="list-style-type: none"> 管理 存储 计算 见证 	
sip	分配给节点的存储 (iSCSI) IP 地址。	字符串
softwareVersion	节点上当前运行的 Element 软件版本。	字符串

相关参考

[AddNodes](#) on page 109

您可以使用 AddNodes 方法将一个或多个新节点添加到集群中。

[ListPendingNodes](#) on page 139

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且已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。

保护域

protectionDomain 该对象包含保护域的名称和类型详细信息。

对象成员

此对象包含以下成员：

名称	说明	类型
保护域名	保护域的名称。	字符串
protectionDomainType	保护域的类型。可能值： <ul style="list-style-type: none"> 机箱：单个机箱中的所有存储节点。 自定义：单个客户定义的保护域中的所有存储节点。 	字符串

protectionDomainLevel

protectionDomainLevel 对象包含有关存储集群的当前容错和故障恢复能力级别的信息。容错级别表示集群在发生故障时可以继续读取和写入数据的能力，而故障恢复能力级别则表示集群在一个相关类型的保护域发生一个或多个故障时自动自行修复的能力。

对象成员

此对象包含以下成员：

名称	说明	类型
protectionDomainType	具有关联容错和故障恢复能力的保护域的类型。可能值： <ul style="list-style-type: none"> node: 任一节点。 chassis: 一个机箱中的任一节点或所有存储节点。 自定义: 单个客户定义的保护域中的所有存储节点。 	字符串
resiliency	从此保护域类型角度来看，此集群的当前故障恢复能力。	protectionDomainResiliency
tolerance	从此保护域类型角度来看，此集群的当前容错能力。	protectionDomainTolerance

protectionDomainResiliency

protectionDomainResiliency 对象包含此存储集群的故障恢复能力状态。故障恢复能力表示存储集群在一个相关类型的保护域发生一个或多个故障时自动自行修复的能力。如果存储集群能够在任一存储节点出现故障（称为节点容错状态）期间继续读取和写入数据，则被视为已修复。

对象成员

此对象包含以下成员：

名称	说明	类型
protectionSchemeResiliencies	一个对象列表（每个保护方案一个对象），包含此相关类型的保护域的故障恢复能力信息。	protectionSchemeResiliency 数组
singleFailureThresholdBytesForBlockData	存储集群上可以存储的最大字节数，超过此数量，此存储集群将无法自动修复到节点容错状态。	整型
sustainableFailuresForEnsemble	在可以自动修复到节点容错状态以进行集合仲裁的情况下可同时发生的预计故障数量。	整型

protectionDomainTolerance

protectionDomainTolerance 对象包含有关存储集群在一个相关类型的保护域发生一个或多个故障时是否能够继续读取和写入数据的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
protectionSchemeTolerances	一个对象列表（每个保护方案一个对象），包含此相关类型的保护域的故障容错信息。	<i>protectionSchemeTolerance</i> 数组
sustainableFailuresForEnsemble	在不丢失集合仲裁的情况下，此相关类型的保护域中可以同时发生的故障数量。	整型

protectionSchemeResiliency

protectionSchemeResiliency 对象包含有关存储集群在特定保护方案下是否能够在相关类型的保护域发生一个或多个故障中自动自行修复的信息。如果存储集群能够在任一存储节点出现故障（称为节点容错状态）期间继续读取和写入数据，则被视为已修复。

对象成员

此对象包含以下成员：

名称	说明	类型
protectionScheme	此存储集群的当前保护方案。唯一可能的值是 doubleHelix。	字符串
sustainableFailuresForBlockData	在可以自动修复到节点数据容错状态的情况下可同时发生的预计故障数量。	整型
sustainableFailuresForMetadata	在可以自动修复到节点元数据容错状态的情况下可同时发生的预计故障数量。	整型

protectionSchemeTolerance

protectionSchemeTolerance 对象包含有关存储集群在特定保护方案下是否能够在发生故障后继续读取和写入数据的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
protectionScheme	此存储集群的当前保护方案。唯一可能的值是 doubleHelix。	字符串
sustainableFailuresForBlockData	在不丢失关联保护方案下块数据可用性的情况下当前可以同时发生的故障数量。	整型
sustainableFailuresForMetadata	在不丢失关联保护方案下元数据可用性的情况下当前可以同时发生的故障数量。	整型

protocolEndpoint

protocolEndpoint 对象包含协议端点的属性。可以通过使用 API 方法 ListProtocolEndpoints 为集群中的所有协议端点检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
primaryProviderID	协议端点的主协议端点提供商对象的 ID。	整型
protocolEndpointID	协议端点的唯一 ID。	UUID
protocolEndpointState	协议端点的状态。可能值： <ul style="list-style-type: none"> Active: 协议端点正在使用中。 Start: 协议端点正在启动。 Failover: 已对协议端点执行故障转移。 Reserved: 已保留协议端点。 	字符串
providerType	协议端点提供程序的类型。可能值： <ul style="list-style-type: none"> Primary Secondary 	字符串
scsiNAADeviceID	协议端点的全局唯一 SCSI 设备标识符（采用 NAA IEEE 注册扩展格式）。	字符串
secondaryProviderID	协议端点的辅助协议端点提供商对象的 ID。	整型

相关参考

[ListProtocolEndpoints](#) on page 472

您可以使用 ListProtocolEndpoints 方法检索有关集群中所有协议端点的信息。协议端点控制对其关联虚拟卷存储容器的访问。

QoS

QoS 对象包含有关卷的服务质量 (QoS) 设置的信息。如果未指定 QoS 值，系统会用默认值创建卷。可以使用 GetDefaultQoS 方法查找默认值。

对象成员

此对象包含以下成员：

名称	说明	类型
burstIOPS	允许短时间内出现的最大“峰值”4 KB IOPS。允许超过 maxIOPS 值的 I/O 活动突发。	整型
burstTime	允许 burstIOPS 持续的时间长度。返回的值以秒为单位表示。此值由系统根据为 QoS 设置的 IOPS 计算而得。	整型
curve	此曲线是一组键-值对。密钥是 I/O 大小（以字节为单位）。值表示执行特定 I/O 大小的 IOP 所消耗的成本。此曲线以 100 IOPS 相对于 4096 字节的操作集计算而得。	JSON 对象
maxIOPS	允许长时间内出现的所需的最大 4 KB IOPS。	整型

名称	说明	类型
minIOPS	要保证的所需的最小 4 KB IOPS。仅当所有卷已被限制在其 minIOPS 值并且性能容量仍不足时，允许的 IOPS 才会降至此水平以下。	整型

相关参考

[GetDefaultQoS](#) on page 383

您可以使用 `GetDefaultQoS` 方法获取新建卷的默认服务质量 (QoS) 值。

QoSPolicy

`QoSPolicy` 对象包含有关运行 Element 软件的存储集群上的 QoS 策略的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
qosPolicyID	存储集群自动分配的 <code>QoSPolicy</code> 的唯一整数标识符。	整型
name	QoS 策略的名称。例如： <code>gold</code> 、 <code>platinum</code> 或 <code>silver</code> 。	字符串
qos	此策略表示的 QoS 设置。	QoS
volumeIDs	与此策略关联的卷的列表。	整型数组

相关参考

[GetQoSPolicy](#) on page 384

您可以使用 `GetQoSPolicy` 方法从系统获取有关特定 QoS 策略的详细信息。

remoteClusterSnapshotStatus

`remoteClusterSnapshotStatus` 对象包含远程存储集群上存储的快照的 UUID 和状态。

您可以使用 `ListSnapshots` 或 `ListGroupSnapshots` API 方法获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
remoteStatus	目标集群上远程快照的复制状态（与从源集群看到的信息一致）。可能值： <ul style="list-style-type: none"> Present：快照位于远程集群上。 NotPresent：快照不在远程集群上。 Syncing：这是一个目标集群，它当前正在复制快照。 Deleted：这是一个目标集群。快照已被删除，但它仍位于源上。 	字符串
volumePairUUID	卷对的通用标识符。	UUID

schedule

schedule 对象包含有关为自主生成卷快照而创建的计划的信息。可以使用 API 方法 ListSchedules 检索所有计划的计划信息。

对象成员

此对象包含以下成员：

名称	说明	类型
attributes	指示计划的运行频率。可能值： <ul style="list-style-type: none"> Day of Week Day of Month Time Interval 	JSON 对象
hasError	指示该计划是否存在错误。可能值： <ul style="list-style-type: none"> true false 	布尔型
hours	显示创建下一个快照之前经过的小时数。可能值为 0 到 24。	整型
lastRunStatus	指示上一个计划快照的状态。可能值： <ul style="list-style-type: none"> Success Failed 	字符串
lastRunTimeStart	指示计划的上次启动时间。	ISO 8601 日期字符串
minutes	显示创建下一个快照之前经过的分钟数。可能值为 0 到 59。	整型
monthdays	指示将在月份中的哪些天生成快照。	数组
paused	指示该计划是否已暂停。可能值： <ul style="list-style-type: none"> true false 	布尔型
recurring	指示该计划是否重复运行。可能值： <ul style="list-style-type: none"> true false 	布尔型
runNextInterval	指示该计划是否在计划程序下次处于活动状态时运行。如果为 true，计划将在计划程序下次处于活动状态时运行，然后此值重置为 false。可能值： <ul style="list-style-type: none"> true false 	布尔型
scheduleID	计划的唯一 ID。	整型
scheduleInfo	包括为计划指定的唯一名称、已创建的快照的保留期限以及从其创建快照的卷的卷 ID。	JSON 对象
scheduleName	分配给计划的唯一名称。	字符串

名称	说明	类型
scheduleType	目前仅支持计划类型的 snapshot。	字符串
snapMirrorLabel	要应用于所创建快照或组快照的 snapMirrorLabel，这些快照包含在 scheduleInfo 中。如果未设置，则此值为 null。	字符串
startingDate	指示首次开始该计划的日期；采用 UTC 时间格式。	ISO 8601 日期字符串
toBeDeleted	指示计划是否标记为待删除。可能值： <ul style="list-style-type: none"> true false 	布尔型
weekdays	指示将在周中的哪些天生生成快照。	数组

相关参考

[ListSchedules](#) on page 455

您可以使用 ListSchedules 获取已创建的所有计划快照的相关信息。

session（光纤通道）

session 对象包含有关集群可识别的每个光纤通道会话以及每个光纤通道会话所使用的目标端口的信息。可以使用 ListFibreChannelSessions API 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
initiatorWWPN	已登录到目标端口的启动程序的全球通用端口名称 (WWPN)。	字符串
nodeID	拥有光纤通道会话的节点。	整型
initiator	有关此光纤通道会话的服务器启动程序的信息。成员： <ul style="list-style-type: none"> alias：分配给该启动程序的友好名称。 attributes：此启动程序的属性。 initiatorID：此启动程序的 ID。 initiatorName：此启动程序的名称。 volumeAccessGroups：与此启动程序关联的卷访问组列表。 	JSON 对象
serviceID	参与该会话的目标端口的服务 ID。	整型
targetWWPN	参与该会话的目标端口的 WWPN。	字符串
volumeAccessGroupID	initiatorWWPN 所属的卷访问组的 ID。如果不在卷访问组中，则此值为 null。	整型

相关参考

[ListFibreChannelSessions](#) on page 167

您可以使用 `ListFibreChannelSessions` 方法列出有关集群上的光纤通道会话的信息。

session (iSCSI)

`session (iSCSI)` 对象包含有关每个卷的 iSCSI 会话的详细信息。可以使用 API 方法 `ListISCSISessions` 检索 iSCSI 会话信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>accountID</code>	用于 CHAP 身份验证的帐户的帐户 ID（如果有）。	整型
<code>accountName</code>	用于 CHAP 身份验证的帐户的名称（如果有）。	字符串
<code>createTime</code>	创建 iSCSI 会话的时间（UTC+0 格式）。	ISO 8601 日期字符串
<code>driveID</code>	与托管会话的传输服务相关联的驱动器 ID。	整型
<code>driveID</code>	报告故障的驱动器的驱动器 ID 列表。如果不适用，则为一个空列表。	整型数组
<code>initiator</code>	有关此 iSCSI 会话的服务器启动程序的信息。成员： <ul style="list-style-type: none"> <code>alias</code>：分配给该启动程序的友好名称。 <code>attributes</code>：此启动程序的属性。 <code>initiatorID</code>：此启动程序的 ID。 <code>initiatorName</code>：此启动程序的名称。 <code>volumeAccessGroups</code>：与此启动程序关联的卷访问组列表。 	JSON 对象
<code>initiatorIP</code>	iSCSI 服务器启动程序的 IP 地址和端口号。	字符串
<code>initiatorName</code>	iSCSI 服务器启动程序的 iSCSI 限定名称 (IQN)。	字符串
<code>initiatorPortName</code>	<code>initiatorName</code> 与 <code>initiatorSessionID</code> 相结合；标识启动程序端口。	字符串
<code>initiatorSessionID</code>	启动程序提供的 48 位 ID，可将 iSCSI 会话标识为属于该启动程序。	整型
<code>msSinceLastIscsiPDU</code>	距离为本次会话接收的最后一个 iSCSI PDU 的时间（以毫秒为单位）。	整型
<code>msSinceLastScsiCommand</code>	距离为本次会话接收的最后一个 SCSI 命令的时间（以毫秒为单位）。	整型
<code>nodeID</code>	与托管会话的传输服务相关联的节点 ID。	整型
<code>serviceID</code>	托管会话的传输服务的服务 ID。	整型
<code>sessionID</code>	iSCSI 会话 ID。	整型
<code>targetIP</code>	iSCSI 存储目标的 IP 地址和端口号。	字符串
<code>targetName</code>	iSCSI 目标的 IQN。	字符串
<code>targetPortName</code>	<code>targetName</code> 与目标门户组标记相结合；标识目标端口。	字符串

名称	说明	类型
virtualNetworkID	与会话相关联的虚拟网络 ID。	整型
volumeID	与会话相关联的卷的卷 ID（如果有）。	整型
volumeInstance	标识与 iSCSI 会话相关联的卷对象（如果有）。	整型

相关参考

[ListISCSISessions](#) on page 136

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

snapMirrorAggregate

`snapMirrorAggregate` 对象包含有关可用 ONTAP 聚合的信息，这些聚合是一组可作为存储供卷使用的磁盘。可以使用 `ListSnapMirrorAggregates` API 方法获取此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
aggregateName	聚合的名称。	字符串
nodeName	此聚合所属的 ONTAP 节点的名称。	字符串
sizeAvailable	聚合中剩余的可用字节数。	整型
sizeTotal	聚合的总大小（以字节为单位）。	整型
percentUsedCapacity	当前正在使用的磁盘空间百分比。	整型
volumeCount	聚合中的卷数。	整型

snapMirrorClusterIdentity

`snapMirrorClusterIdentity` 对象包含有关 `SnapMirror` 关系中远程 ONTAP 集群的标识信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
clusterName	目标 ONTAP 集群的名称。	字符串
clusterUUID	目标 ONTAP 集群的 128 位通用唯一标识符。	字符串
clusterSerialNumber	目标 ONTAP 集群的序列号。	字符串

snapMirrorEndpoint

`snapMirrorEndpoint` 对象包含有关与 Element 存储集群通信的远程 `SnapMirror` 存储系统的信息。可以使用 `ListSnapMirrorEndpoints` API 方法检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	本地集群中对象的唯一标识符。	整型
managementIP	端点的集群管理 IP 地址。	字符串
clusterName	ONTAP 集群名称。此值会自动使用 snapMirrorClusterIdentity 对象中的“clusterName”值进行填充。	字符串
username	ONTAP 系统的管理用户名。	字符串
ipAddresses	集群中所有节点的集群间存储 IP 地址列表。可通过 ListSnapMirrorNetworkInterfaces 方法获取这些 IP 地址。	字符串数组
isConnected	ONTAP 集群控制链路的连接状态。	布尔型

snapMirrorJobScheduleCronInfo

snapMirrorJobScheduleCronInfo 对象包含有关 ONTAP 系统上的 cron 作业计划的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
jobScheduleName	作业计划的名称。	字符串
jobScheduleDescription	自动生成的计划摘要，可供用户查看。	字符串

snapMirrorLunInfo

snapMirrorLunInfo 对象包含有关 ONTAP LUN 对象的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
creationTimestamp	LUN 的创建时间。	ISO 8601 日期字符串
lunName	LUN 的名称。	字符串
path	LUN 的路径。	字符串
size	LUN 的大小（以字节为单位）。	整型
sizeUsed	LUN 使用的字节数。	整型

名称	说明	类型
state	LUN 的当前访问状态。可能值： <ul style="list-style-type: none"> • online • offline • foreign_lun_error • nvfail • space_error 	字符串
volume	包含 LUN 的卷的名称。	字符串
vserver	包含 LUN 的 SVM。	字符串

snapMirrorNetworkInterface

snapMirrorNetworkInterface 对象包含有关集群间逻辑接口 (LIF) 的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
administrativeStatus	逻辑接口 (LIF) 是否已由管理员启用或禁用。可能值： <ul style="list-style-type: none"> • up • down 	字符串
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
interfaceName	LIF 名称。	字符串
networkAddress	LIF 的 IP 地址。	字符串
networkMask	LIF 的网络掩码。	字符串
interfaceRole	LIF 的角色。可能值： <ul style="list-style-type: none"> • undef • 集群 • data • node_mgmt • intercluster • cluster_mgmt 	字符串
operationalStatus	LIF 的运行状态（是否已成功建立连接）。如果出现网络问题，导致此接口无法正常运行，则此状态可能与管理状态不同。可能值： <ul style="list-style-type: none"> • up • down 	字符串
vserverName	SVM 的名称。	字符串

snapMirrorNode

snapMirrorNode 对象包含有关 SnapMirror 关系中目标 ONTAP 集群的节点的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
name	ONTAP 节点的名称。	字符串
model	ONTAP 节点的型号。	字符串
serialNumber	ONTAP 节点的序列号。	字符串
productVersion	ONTAP 产品版本。	字符串
isNodeHealthy	ONTAP 集群中节点的运行状况。可能值： <ul style="list-style-type: none"> • true • false 	字符串
isNodeEligible	节点是否符合加入 ONTAP 集群的条件。可能值： <ul style="list-style-type: none"> • true • false 	字符串

snapMirrorPolicy

snapMirrorPolicy 对象包含有关 ONTAP 系统上存储的 SnapMirror 策略的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
policyName	分配给策略的唯一名称。	字符串
policyType	策略类型。可能值： <ul style="list-style-type: none"> • async_mirror • mirror_vault 	字符串
comment	与 SnapMirror 策略关联的描述，可供用户查看。	字符串
transferPriority	SnapMirror 传输的运行优先级。可能值： <ul style="list-style-type: none"> • normal：默认优先级。这些传输安排在大多数 low 优先级传输之前进行。 • low：这些传输的优先级最低，并安排在大多数 normal 优先级传输之后进行。 	字符串
policyRules	用于描述策略规则的对象列表。	snapMirrorPolicyRule 数组
totalKeepCount	策略中所有规则的保留总数。	整型
totalRules	策略中的规则总数。	整型

名称	说明	类型
vserverName	SnapMirror 策略对应的 SVM 名称。	字符串

snapMirrorPolicyRule

snapMirrorPolicyRule 对象包含有关 SnapMirror 策略中的规则的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorLabel	Snapshot 副本标签，用于在扩展数据保护关系中选择 Snapshot 副本。	字符串
keepCount	指定某个规则中 SnapMirror 目标卷上保留的最多 Snapshot 副本数。	整型

snapMirrorRelationship

snapMirrorRelationship 对象包含有关 Element 卷和 ONTAP 卷之间的 SnapMirror 关系的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
snapMirrorRelationshipID	数组中每个 snapMirrorRelationship 对象的唯一标识符，此对象会在 ListSnapMirrorRelationships 中返回。此 UUID 在 ONTAP 系统中创建并从该系统中返回。	字符串
sourceVolume	用于描述源卷的对象。	snapMirrorVolumeInfo
destinationVolume	用于描述目标卷的对象。	snapMirrorVolumeInfo
currentMaxTransferRate	源卷和目标卷之间的当前最大传输速率（以 KB/秒为单位）。	整型
isHealthy	关系运行状况是否正常。可能值： <ul style="list-style-type: none"> • true: 关系运行正常。 • false: 关系运行异常。造成运行异常的原因可能是，手动更新或计划更新失败或被迫中止，或者上次计划更新推迟。 	布尔型
lagtime	目标卷上的数据滞后于源卷上数据的时间长度（以秒为单位）。	整型
lastTransferDuration	完成上次传输所用时间长度（以秒为单位）。	整型

名称	说明	类型
lastTransferError	用于描述上次传输失败的原因的消息。	字符串
lastTransferSize	上次传输过程中传输的总字节数。	整型
lastTransferEndTimestamp	上次传输结束的时间戳。	ISO 8601 日期字符串
lastTransferType	上次在关系中传输的类型。	字符串
maxTransferRate	指定卷之间的最大数据传输速率（以 KB/秒为单位）。默认值 0 表示无限制，允许 SnapMirror 关系充分利用可用的网络带宽。	整型
mirrorState	SnapMirror 关系的镜像状态。可能值： <ul style="list-style-type: none"> uninitialized: 目标卷尚未初始化。 snapmirrored: 目标卷已初始化并已准备好接收 SnapMirror 更新。 broken-off: 目标卷为读写卷，并且存在快照。 	字符串
newestSnapshot	目标卷上最新 Snapshot 副本的名称。	字符串
policyName	指定关系的 ONTAP SnapMirror 策略名称。可通过 ListSnapMirrorPolicies 检索可用策略列表。示例值为 “MirrorLatest” 和 “MirrorAndVault”。	字符串
policyType	关系的 ONTAP SnapMirror 策略类型。请参见 ListSnapMirrorPolicies。例如：“async_mirror” 或 “mirror_vault”。	字符串
relationshipProgress	迄今为止在关系的当前活动中处理的总字节数，此活动已在 relationship-status 中返回。只有当 “relationshipStatus” 成员指示正在执行活动时，才能设置此项。	整型
relationshipStatus	SnapMirror 关系的状态。可能值： <ul style="list-style-type: none"> idle transferring 检查 quiescing quiesced queued preparing finalizing aborting 中断 	字符串

名称	说明	类型
relationshipType	SnapMirror 关系的类型。在运行 Element 软件的存储集群上，此值始终为“extended_data_protection”。	字符串
scheduleName	ONTAP 系统上用于更新 SnapMirror 关系的原有 cron 计划的名称。可通过 ListSnapMirrorSchedules 检索可用计划列表。	字符串
unhealthyReason	关系运行异常的原因。	字符串

snapMirrorVolume

snapMirrorVolume 对象包含有关 ONTAP 卷的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
name	卷的名称。	字符串
type	卷的类型。可能值： <ul style="list-style-type: none"> rw: 读写卷 ls: 负载共享卷 dp: 数据保护卷 	字符串
vserver	此卷所属的 SVM 的名称。	字符串
aggrName	所属聚合名称。	字符串
state	卷的状态。可能值： <ul style="list-style-type: none"> online restricted offline mixed 	字符串
size	卷的文件系统总大小（以字节为单位）。	字符串
availSize	卷中可用空间的大小（以字节为单位）。	字符串

snapMirrorVolumeInfo

snapMirrorVolumeInfo 对象包含有关 SnapMirror 关系中卷位置的信息（例如其名称和类型）。

对象成员

此对象包含以下成员：

名称	说明	类型
type	卷的类型。可能值： <ul style="list-style-type: none"> • solidfire: 卷位于运行 Element 软件的存储集群中。 • ontap: 卷位于远程 ONTAP 集群中。 	字符串
volumeID	卷的 ID。只有当“type”为 solidfire 时才有效。	整型
vserver	此卷所属的 SVM 的名称。只有当“type”为 ontap 时才有效。	字符串
name	卷的名称。	字符串

snapMirrorVserver

snapMirrorVserver 对象包含有关目标 ONTAP 集群中的 Storage Virtual Machine (SVM) 的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
snapMirrorEndpointID	目标 ONTAP 系统的 ID。	整型
vserverName	SVM 的名称。	字符串
vserverType	SVM 的类型。可能值： <ul style="list-style-type: none"> • data • admin • system • node 	字符串
vserverSubtype	SVM 的子类型。可能值： <ul style="list-style-type: none"> • 默认 • dp_destination • data • sync_source • sync_destination 	字符串
rootVolume	SVM 的根卷。	字符串
rootVolumeAggregate	要创建根卷的聚合。	字符串
vserverAggregateInfo	snapMirrorVserverAggregateInfo 对象数组。	JSON 对象
adminState	SVM 的详细管理状态。可能值： <ul style="list-style-type: none"> • running • stopped • 启动 • 停止 • initializing • 删除 	字符串

名称	说明	类型
operationalState	SVM 的基本运行状态。可能值： <ul style="list-style-type: none"> • running • stopped 	字符串

snapMirrorVserverAggregateInfo

snapMirrorVserverAggregateInfo 对象包含有关目标 ONTAP 集群中可用数据 Storage Virtual Machine (SVM) 的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
aggrName	分配给 SVM 的聚合的名称。	字符串
aggrAvailSize	分配聚合的可用大小。	整型

snapshot

snapshot 对象包含有关为卷生成的快照的信息。您可以使用 ListSnapshots API 方法检索一个卷或所有卷的快照信息列表。该对象包含有关活动快照以及为卷创建的每个快照的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
checksum	以短字符串表示已存储快照中的数据。此校验和以后可用于比较其他快照，以检测数据中存在的错误。	字符串
createTime	快照的创建时间（UTC+0 格式）。	ISO 8601 日期字符串
enableRemoteReplication	标识是否为快照启用远程复制。	布尔型
expirationReason	指示如何设置快照到期时间。可能值： <ul style="list-style-type: none"> • Api：使用 API 设置到期时间。 • None：不设置到期时间。 • Test：所设置的到期时间用于测试。 	字符串
expirationTime	该快照到期并从集群中清除的时间。	ISO 8601 日期字符串
groupID	如果快照属于某个组快照，则为组 ID。	整型
groupsnapshotUUID	包含有关组中每个快照的信息。每个成员都具有一个 UUID 参数，该参数表示快照的 UUID。	字符串
instanceCreateTime	在本地集群中创建快照的时间。	ISO 8601 日期字符串
instanceSnapshotUUID	本地集群中快照的通用唯一 ID。此 ID 不会复制到其他集群。	字符串

名称	说明	类型
name	分配给快照的唯一名称。如果未指定任何名称，则此名称为快照的创建时间戳（采用 UTC +0 格式）。	字符串
remoteStatuses	一个数组，其中包含目标集群上每个远程快照的通用标识符和复制状态（与从源集群看到的信息一致）。	<i>remoteClusterSnapshotStatus</i> 数组
snapMirrorLabel	SnapMirror 软件用于在 SnapMirror 端点上指定快照保留策略的标签。如果未设置，则此值为 null。	字符串
snapshotID	现有快照的唯一 ID。	字符串
snapshotUUID	现有快照的通用唯一 ID。在集群间复制快照时，此 ID 会随此快照一起复制，并用于在集群间标识此快照。	字符串
status	快照的当前状态。可能值： <ul style="list-style-type: none"> Unknown: 获取此快照的状态时出错。 Preparing: 正在准备此快照以供使用，此快照目前不可写入。 RemoteSyncing: 正在从远程集群复制此快照。 Done: 此快照已完成准备或复制，现可供使用。 Active: 此快照为活动分支。 Cloning: 此快照参与 CopyVolume 操作。 	字符串
totalSize	快照的总大小（以字节为单位）。	整型
virtualVolumeID	与此快照关联的虚拟卷的 ID。	UUID
volumeID	从中创建该快照的卷的 ID。	整型
volumeName	创建快照时的卷名称。	字符串

相关参考

[ListSnapshots](#) on page 456

您可以使用 `ListSnapshots` 返回在卷上生成的每个快照的属性。

snmpTrapRecipient

`snmpTrapRecipient` 对象包含有关配置为接收存储集群生成的 SNMP 陷阱的主机的信息。您可以使用 `GetSnmpTrapInfo` API 方法获取配置为接收 SNMP 陷阱的主机的列表。

对象成员

此对象包含以下成员：

名称	说明	类型
host	目标主机的 IP 地址或主机名。	字符串
port	主机上要发送陷阱的 UDP 端口号。有效范围为 1 至 65535。0（零）不是有效端口号。默认端口为 162。	整型

名称	说明	类型
community	SNMP 社区字符串。	字符串

storageContainer

storageContainer 对象包含虚拟卷存储容器的属性。可以使用 API 方法 ListStorageContainers 为集群中的每个存储容器检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
accountID	与该存储容器关联的存储系统帐户的 ID。	整型
initiatorSecret	与该存储容器关联的启动程序的 CHAP 身份验证密钥。	字符串
name	存储容器的名称。	字符串
protocolEndpointType	存储容器的协议端点类型。SCSI 是唯一的有效值。	字符串
status	存储容器的状态。可能值： <ul style="list-style-type: none"> Active: 存储容器正在使用中。 Locked: 存储容器已锁定。 	字符串
storageContainerID	存储容器的唯一 ID。	UUID
targetSecret	与该存储容器关联的目标的 CHAP 身份验证密钥。	字符串
virtualVolumes	与此存储容器关联的虚拟卷的 ID 列表。	UUID 数组

相关参考

[ListStorageContainers](#) on page 474

您可以使用 ListStorageContainers 方法检索有关系统已知的所有虚拟卷存储容器的信息。

syncJob

syncJob 对象包含有关正在集群上运行的克隆、远程复制或分区同步作业的信息。

可以使用 ListSyncJobs API 方法检索同步信息。

对象成员

此对象包含以下成员：

名称	说明	类型
blocksPerSecond	每秒从源集群传输到目标集群的数据块数量。只有当 type 成员设置为 remote 时才具有此成员。	浮点型
branchType	仅为远程复制同步作业返回。可能值： <ul style="list-style-type: none"> snapshot volume 	字符串
bytesPerSecond	克隆每秒处理的字节数。只有当 type 成员设置为 clone 或 slice 时才具有此成员。	浮点型

名称	说明	类型
cloneID	正在进行的克隆操作的标识符。只有当 type 成员设置为 clone 时才具有此成员。	整型
currentBytes	克隆已在源卷中处理的字节数。只有当 type 成员设置为 clone 或 slice 时才具有此成员。	整型
dstServiceID	托管卷的主副本的服务的服务标识符。只有当 type 成员设置为 remote 时才具有此成员。	整型
dstVolumeID	目标卷 ID。只有当 type 成员设置为 remote 或 clone 时才具有此成员。	整型
elapsedTime	自同步作业启动以来经过的时间（以秒为单位）。	浮点型
groupCloneID	正在进行的组克隆操作的 ID。	整型
nodeID	指定发生克隆操作的节点。只有当 type 成员设置为 clone 时才具有此成员。	整型
percentComplete	同步作业完成的百分比。	整型
remainingTime	估计的操作完成时间（以秒为单位）。	整型
sliceID	所同步的分区驱动器的 ID。	整型
stage	只有当 type 成员设置为 remote 或 clone 时才具有此成员。可能值： <ul style="list-style-type: none"> • metadata: 复制正在确定需要将哪些数据传输到远程集群。复制过程的这一阶段不会报告状态。 • data: 复制正在将大量数据传输到远程集群。 • whole: 为分区同步作业指示分区的向后兼容性。 	字符串
snapshotID	从其创建该克隆的快照的 ID。只有当 type 成员设置为 clone 时才具有此成员。	整型
srcServiceID	源服务 ID。	整型
srcVolumeID	源卷 ID。	整型
totalBytes	克隆的总字节数。只有当 type 成员设置为 clone 或 slice 时才具有此成员。	整型
type	同步操作的类型。可能值： <ul style="list-style-type: none"> • clone • slice • block • remote 	字符串

相关参考

[ListSyncJobs](#) on page 396

您可以使用 `ListSyncJobs` 方法获取有关在 Element 存储集群上运行的同步作业的信息。此方法可返回有关分区、克隆、块和远程同步作业的信息。

task (虚拟卷)

`task` 对象包含系统中当前正在运行或已完成的虚拟卷任务的相关信息。您可以使用 `ListVirtualVolumeTasks` 方法为所有虚拟卷任务检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>cancelled</code>	指示该任务是否已取消。可能值： <ul style="list-style-type: none"> <code>true</code> <code>false</code> 	布尔型
<code>cloneVirtualVolumeID</code>	要克隆的虚拟卷的唯一虚拟卷 ID（用于克隆任务）。	UUID
<code>parentMetadata</code>	包含用于克隆虚拟卷或创建虚拟卷快照的任务的父级元数据的对象。	JSON 对象
<code>parentTotalSize</code>	克隆或快照任务的父级的可用总空间（以字节为单位）。	整型
<code>parentUsedSize</code>	克隆或快照任务的父级的已用空间（以字节为单位）。	整型
<code>operation</code>	任务所执行操作的类型。可能值： <ul style="list-style-type: none"> <code>unknown</code>: 此任务操作未知。 <code>prepare</code>: 此任务正在准备虚拟卷。 <code>snapshot</code>: 此任务正在创建虚拟卷的快照。 <code>rollback</code>: 此任务正在将虚拟卷回滚到快照。 <code>clone</code>: 此任务正在创建虚拟卷的克隆。 <code>fastClone</code>: 此任务正在创建虚拟卷的快速克隆。 <code>copyDiffs</code>: 此任务正在将不同块复制到虚拟卷。 	字符串
<code>status</code>	虚拟卷任务的当前状态。可能值： <ul style="list-style-type: none"> <code>Error</code>: 此任务已失败并返回错误。 <code>Queued</code>: 此任务正在等待运行。 <code>Running</code>: 此任务当前正在运行。 <code>Success</code>: 此任务已成功完成。 	字符串
<code>virtualVolumeHostID</code>	已启动任务的主机的唯一 ID。	UUID
<code>virtualVolumeID</code>	新的唯一虚拟卷 ID（用于创建新虚拟卷的任务）。	UUID
<code>virtualVolumeTaskID</code>	任务的唯一 ID。	UUID

相关参考

[ListVirtualVolumeTasks](#) on page 479

您可以使用 `ListVirtualVolumeTasks` 方法获取系统中的虚拟卷任务的列表。

usmUser

您可以将 `usmUser` 对象与 `SetSnmpInfo` API 方法结合使用来为存储集群配置 SNMP。

对象成员

此对象包含以下成员：

名称	说明	类型
访问	此用户的 SNMP 访问类型。可能值： <ul style="list-style-type: none"> <code>rouser</code>：只读访问权限。 <code>rwuser</code>：读写访问权限。所有 Element 软件 MIB 对象均为只读对象。 	字符串
name	用户的名称。	字符串
密码	用户的密码。	字符串
密码短语	用户的密码短语。	字符串
secLevel	此用户所需的凭据类型。可能值： <ul style="list-style-type: none"> <code>noauth</code>：不需要密码或密码短语。 <code>auth</code>：用户访问需要密码。 <code>priv</code>：用户访问需要密码和密码短语。 	字符串

相关参考

[SetSnmpInfo](#) on page 341

您可以使用 `SetSnmpInfo` 方法在集群节点上配置 SNMP 版本 2 和版本 3。使用此接口设置的值将应用于集群中的所有节点，而且所传递的值将替换在先前任意 `SetSnmpInfo` 调用中设置的所有值。

virtualNetwork

`virtualNetwork` 对象包含有关特定虚拟网络的信息。您可以使用 `ListVirtualNetworks` API 方法为系统中的所有虚拟网络检索此信息的列表。

对象成员

此对象包含以下成员：

名称	说明	类型
addressBlocks	当前分配给虚拟网络的地址块的范围。成员： <ul style="list-style-type: none"> <code>available</code>：用多个“1”和“0”表示的二进制字符串。“1”表示该 IP 地址可用，“0”表示该 IP 不可用。系统按从右到左的顺序读取该字符串，并以最右端旁边的数位作为地址块列表中的第一个 IP 地址。 <code>size</code>：此地址块的大小。 <code>start</code>：块中的第一个 IP 地址。 	JSON 对象数组
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
name	分配给虚拟网络的名称。	字符串
网络掩码	虚拟网络的网络掩码的 IP 地址。	字符串

名称	说明	类型
svip	虚拟网络的存储 IP 地址。	字符串
gateway	用于虚拟网络的网关。	字符串
virtualNetworkID	虚拟网络的唯一标识符。	整型
virtualNetworkTag	VLAN 标记标识符。	整型

相关参考

[ListVirtualNetworks](#) on page 352

您可以使用 `ListVirtualNetworks` 方法列出为集群配置的所有虚拟网络。

virtualVolume

`virtualVolume` 对象包含有关虚拟卷的配置信息，以及有关虚拟卷的快照的信息。它不包含运行时信息或使用情况信息。您可以使用 `ListVirtualVolumes` 方法为集群检索此信息。

对象成员

此对象包含以下成员：

名称	说明	类型
bindings	此虚拟卷的绑定 ID 的列表。	UUID 数组
children	作为此虚拟卷的子级的虚拟卷 UUID 的列表。	UUID 数组
descendants	如果将 <code>recursive: true</code> 传递到 <code>ListVirtualVolumes</code> 方法，则包含此虚拟卷的后代虚拟卷 UUID 的列表。	UUID 数组
metadata	虚拟卷元数据的键-值对，例如虚拟卷类型、子操作系统类型等。	JSON 对象
parentVirtualVolumeID	父虚拟卷的虚拟卷 ID。如果 ID 全为零，则它是一个未链接到父虚拟卷的独立虚拟卷。	UUID
snapshotID	底层卷快照的 ID。如果虚拟卷不表示快照，此值为“0”。	整型
snapshotInfo	关联快照的 <code>snapshot</code> 对象（如果不存在，则为 <code>null</code> ）。	snapshot
status	虚拟卷的当前状态。可能值： <ul style="list-style-type: none"> <code>cloning</code>: 虚拟卷正在进行处理以响应克隆或快照操作。 <code>waiting</code>: 虚拟卷正在等待快照操作完成。 <code>ready</code>: 虚拟卷已准备好用于一般用途。 	字符串
storageContainer	描述拥有此虚拟卷的存储容器的对象。	storageContainer
virtualVolumeID	虚拟卷的唯一 ID。	UUID
virtualVolumeType	虚拟卷的类型。	字符串
volumeID	底层卷的 ID。	整型
volumeInfo	如果将 <code>details: true</code> 传递到 <code>ListVirtualVolumes</code> 方法，则此成员是一个用于说明此卷的对象。	volume

相关参考

[ListVirtualVolumes](#) on page 477

您可以使用 `ListVirtualVolumes` 方法列出系统中当前所含的虚拟卷。您可以使用此方法列出所有虚拟卷或者仅列出一部分虚拟卷。

[snapshot](#) on page 65

`snapshot` 对象包含有关为卷生成的快照的信息。您可以使用 `ListSnapshots` API 方法检索一个卷或所有卷的快照信息列表。该对象包含有关活动快照以及为卷创建的每个快照的信息。

[storageContainer](#) on page 67

`storageContainer` 对象包含虚拟卷存储容器的属性。可以使用 API 方法 `ListStorageContainers` 为集群中的每个存储容器检索此信息。

[volume](#) on page 72

`volume` 对象包含未配对或已配对卷的相关配置信息。它不包含运行时信息或使用情况信息，也不包含虚拟卷的特定信息。

volume

`volume` 对象包含未配对或已配对卷的相关配置信息。它不包含运行时信息或使用情况信息，也不包含虚拟卷的特定信息。

对象成员

此对象包含以下成员：

名称	说明	类型
访问	卷允许的访问类型。可能值： <ul style="list-style-type: none"> <code>readOnly</code>: 仅允许执行读取操作。 <code>readWrite</code>: 允许执行读取和写入。 <code>locked</code>: 不允许执行任何读取或写入。 <code>replicationTarget</code>: 指定为复制的卷对中的目标卷。 	字符串
<code>accountID</code>	包含卷的帐户的帐户 ID。	整型
<code>attributes</code>	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
<code>blockSize</code>	卷上的块的大小。	整型
<code>createTime</code>	卷的创建时间（UTC+0 格式）。	ISO 8601 字符串
<code>currentProtectionScheme</code>	此卷正在使用的保护方案。如果要将卷从一种保护方案转换为另一种保护方案，则此成员将反映要将此卷转换到的保护方案。	字符串
<code>deleteTime</code>	卷的删除时间（UTC+0 格式）。	ISO 8601 字符串
<code>enable512e</code>	如果设置为 <code>true</code> ，则卷提供 512 字节的扇区模拟。	布尔型

名称	说明	类型
enableSnapMirrorReplication	卷是否可用于复制 SnapMirror 端点。	布尔型
iqn	卷的 iSCSI 限定名称。	字符串
lastAccessTime	上次对卷进行任何访问（包括执行 I/O）的时间（采用 UTC+0 格式）。如果不知道上次访问时间，则此值为 null。	ISO 8601 字符串
lastAccessTimeIO	上次对卷执行任何 I/O 的时间（采用 UTC+0 格式）。如果不知道上次访问时间，则此值为 null。	ISO 8601 字符串
name	在创建卷时为卷提供的名称。	字符串
previousProtectionScheme	如果卷从一个保护方案转换到另一个保护方案，此成员反映了卷从中转换的保护方案。此成员只有在转换开始后才会发生更改。如果从未转换过卷，则此成员为 null。	字符串
purgeTime	从系统中清除卷的时间（采用 UTC+0 格式）。	ISO 8601 字符串
qos	该卷的新服务质量设置。	<i>QoS</i>
qosPolicyID	与卷关联的 QoS 策略 ID。如果卷不与策略关联，则此值为 null。	整型
scsiEUIDeviceID	卷的全局唯一 SCSI 设备标识符（采用基于 EUI-64 的 16 字节格式）。	字符串
scsiNAADeviceID	卷的全局唯一 SCSI 设备标识符（采用 NAA IEEE 注册扩展格式）。	字符串
sliceCount	卷上的分区数量。此值始终为“1”。	整型
status	卷的当前状态。可能值： <ul style="list-style-type: none"> • init: 卷正在进行初始化并且尚未准备好进行连接。 • active: 卷处于活动状态并且已准备好进行连接。 • deleted: 卷已标记为待删除，但尚未清除。 	字符串
totalSize	已配置容量的总字节数。	整型
virtualVolumeID	与该卷关联的唯一虚拟卷 ID（如果有）。	UUID
volumeAccessGroups	卷所属卷访问组的 ID 列表。如果卷不属于任何卷访问组，则此值为空列表。	整型数组
volumeConsistencyGroupUUID	卷所属的卷一致性组的通用唯一 ID。	UUID

名称	说明	类型
volumeID	卷的唯一卷 ID。	整型
volumePairs	有关已配对卷的信息。只有当卷已配对时，才会显示此信息。如果卷未配对，则此值为空列表。	<i>volumePair</i> 数组
volumeUUID	卷的通用唯一 ID。	UUID

相关参考

[ListActiveVolumes](#) on page 389

您可以使用 `ListActiveVolumes` 方法获取系统中当前所含活动卷的列表。卷列表按卷 ID 排序，并且可以在多个部分（页面）中返回。

[ListDeletedVolumes](#) on page 392

您可以使用 `ListDeletedVolumes` 方法检索已标记为待删除且已从系统中清除的卷的列表。

[ListVolumes](#) on page 399

您可以使用 `ListVolumes` 方法获取集群中所含卷的列表。可以使用可用参数来指定要在列表中返回的卷。

[ListVolumesForAccount](#) on page 404

您可以使用 `ListVolumesForAccount` 方法列出帐户的活动卷和（待定）已删除卷。

[QoS](#) on page 52

QoS 对象包含有关卷的服务质量 (QoS) 设置的信息。如果未指定 QoS 值，系统会用默认值创建卷。可以使用 `GetDefaultQoS` 方法查找默认值。

volumeAccessGroup

`volumeAccessGroup` 对象包含有关特定卷访问组的信息。可以使用 API 方法 `ListVolumeAccessGroups` 为所有访问组检索此信息的列表。

对象成员

此对象包含以下成员：

名称	说明	类型
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象
deletedVolumes	已从尚未从系统中清除的卷访问组中删除的卷的数组。	整型数组
initiatorIDs	映射到卷访问组的启动程序的 ID 列表。	整型数组
initiators	映射到该卷访问组的启动程序的唯一 IQN/WWPN 数组。	字符串数组
name	卷访问组的名称。	字符串
volumeAccessGroupID	卷访问组的唯一卷访问组 ID 标识符。	整型
volumes	属于该卷访问组的卷 ID 的列表。	整型数组

相关参考

[ListVolumeAccessGroups](#) on page 403

您可以使用 `ListVolumeAccessGroups` 方法获取系统中当前所含卷访问组的相关消息。

volumePair

`volumePair` 对象包含与其他集群上的其他卷配对的卷的相关信息。如果卷未配对，则此对象为空。您可以使用 `ListActivePairedVolumes` 和 `ListActiveVolumes` API 方法返回有关已配对卷的信息。

对象成员

此对象包含以下成员：

名称	说明	类型
<code>clusterPairID</code>	卷在其中进行配对的集群。	整型
<code>remoteReplication</code>	卷复制的详细信息。成员： <ul style="list-style-type: none"> <code>mode</code>: (字符串) “Async”、“Sync” 或 “SnapshotsOnly” 之一。 <code>pauseLimit</code>: (整型) 仅供内部使用。 <code>remoteServiceID</code>: (整型) 远程分区服务 ID。 <code>resumeDetails</code>: (字符串) 保留以供将来使用。 <code>snapshotReplication</code> (JSON 对象) <ul style="list-style-type: none"> <code>state</code>: (字符串) 正在执行的快照复制的状态 (如果有)。 <code>stateDetails</code>: (字符串) 保留以供将来使用。 <code>state</code>: (字符串) 卷复制的状态。 <code>stateDetails</code>: (字符串) 保留以供将来使用。 	JSON 对象
<code>remoteSliceID</code>	远程集群上集群定义的分区 ID。	整型
<code>remoteVolumeID</code>	远程集群上与本地卷配对的卷的 ID。	整型
<code>remoteVolumeName</code>	远程卷的名称。	字符串
<code>volumePairUUID</code>	此配对的集群定义的通用唯一标识符 (采用规范格式)。	字符串

相关参考

[ListActivePairedVolumes](#) on page 264

您可以使用 `ListActivePairedVolumes` 方法列出与某个卷配对的所有活动卷。此方法将返回有关活动和待定配对的卷的信息。

[ListActiveVolumes](#) on page 389

您可以使用 `ListActiveVolumes` 方法获取系统中当前所含活动卷的列表。卷列表按卷 ID 排序，并且可以在多个部分 (页面) 中返回。

volumeStats

`volumeStats` 对象包含单个卷的统计数据。

对象成员

您可以使用以下方法获取部分或所有卷的 `volumeStats` 对象：

- [GetVolumeStats](#)
- [ListVolumeStatsByAccount](#)

- [ListVolumeStatsByVolume](#)
- [ListVolumeStatsByVolumeAccessGroup](#)

此对象包含以下成员：

名称	说明	计算方式	类型
accountID	卷所有者的帐户 ID。	不适用	整型
actualIOPS	卷在过去 500 毫秒的当前实际 IOPS。	时间点	整型
asyncDelay	自卷上次与远程集群同步以来经过的时间长度。如果卷未配对，则此成员为 null。 注： 处于活动复制状态的目标卷的 asyncDelay 始终为 0（零）。目标卷在复制期间能够感知系统，并假设 asyncDelay 始终准确无误。	不适用	ISO 8601 持续时间字符串或为 null
averageIOPSize	卷在过去 500 毫秒的最新 I/O 的平均大小（以字节为单位）。	时间点	整型
burstIOPSCredit	用户可用的 IOPS 额度的总数。如果卷使用的 IOPS 未达到配置的最大 IOPS，则额度会积累。	不适用	整型
clientQueueDepth	对卷执行的未完成读取和写入操作的数量。	不适用	整型
clusterUtilization	正在利用的集群容量。	不适用	浮点型
desiredMetadataHosts	在元数据服务之间迁移卷元数据时要迁移到的元数据（分区）服务。“null”值表示此卷不迁移。	不适用	JSON 对象
latencyUsec	过去 500 毫秒对卷完成操作的平均时间（以微秒为单位）。值为“0”（零）表示卷不具有任何 I/O。	时间点	整型
metadataHosts	卷元数据所在的元数据（分区）服务。可能值： <ul style="list-style-type: none"> • primary: 托管卷的主元数据服务。 • liveSecondaries: 当前处于“活动”状态的二级元数据服务。 • deadSecondaries: 处于非活动状态的二级元数据服务。 	不适用	JSON 对象
nonZeroBlocks	上次垃圾回收操作完成之后包含数据的 4KiB 块的总数。	不适用	整型
readBytes	自卷创建以来从卷读取的总累积字节数。	单调递增	整型
readBytesLastSample	在最后一个采样周期内从卷读取的总字节数。	时间点	整型

名称	说明	计算方式	类型
readLatencyUsec	过去 500 毫秒内完成对卷执行读取操作所需的平均时间（以微秒为单位）。	时间点	整型
readLatencyUsecTotal	从卷执行读取操作所用的总时间。	单调递增	整型
readOps	自卷创建以来卷的总读取操作数。	单调递增	整型
readOpsLastSample	最后一个采样周期内读取操作的总数。	时间点	整型
samplePeriodMsec	采样周期的长度（以毫秒为单位）。	不适用	整型
throttle	一个介于 0 到 1 之间的浮点值，表示系统由于数据重新复制、瞬时错误和生成快照而使客户端 IOPS 低于其最大 IOPS 多少。	不适用	浮点型
timestamp	当前时间（UTC+0 格式）。	不适用	ISO 8601 日期字符串
unalignedReads	自卷创建以来卷的总累积未对齐读取操作数。	单调递增	整型
unalignedWrites	自卷创建以来卷的总累积未对齐写入操作数。	单调递增	整型
volumeAccessGroups	卷所属的卷访问组的 ID 列表。	不适用	整型数组
volumeID	卷的 ID。	不适用	整型
volumeSize	总的已配置容量（以字节为单位）。	不适用	整型
volumeUtilization	一个浮点值，说明客户端当前的卷利用率。可能值： <ul style="list-style-type: none"> • 0: 客户端未使用卷。 • 1: 客户端正在使用其最大值。 • >1: 客户端正在使用其突发。 	不适用	浮点型
writeBytes	自卷创建以来写入卷的总累积字节数。	单调递增	整型
writeBytesLastSample	在最后一个采样周期内写入卷的总字节数。	单调递增	整型
writeLatencyUsec	过去 500 毫秒内完成对卷执行写入操作所需的平均时间（以微秒为单位）。	时间点	整型
writeLatencyUsecTotal	对卷执行写入操作所用的总时间。	单调递增	整型
writeOps	自卷创建以来卷的总累积写入操作数。	单调递增	整型
writeOpsLastSample	最后一个采样周期内写入操作的总数。	时间点	整型
zeroBlocks	上次垃圾回收操作完成之后不包含数据的空 4KiB 块的总数。	时间点	整型

通用方法

通用方法是指用于检索有关存储集群、API 本身或正在执行的 API 操作的信息的方法。

GetAPI

可以使用 GetAPI 方法获取可在系统中使用的所有 API 方法和受支持 API 端点的列表。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<version>	此软件版本支持的所有 API 方法的列表，其中 <version> 是此系统正在运行的当前软件版本。	字符串数组
currentVersion	存储集群软件的当前版本。	字符串
supportedVersions	系统支持的所有 API 端点的列表。	字符串数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetAPI",
  "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": {
    "12.0" : [
      "AbortSnapMirrorRelationship",
      "AddAccount",
      "AddClusterAdmin",
      "AddDrives",
      "AddDdpClusterAdmin" ,
      "AddInitiatorsToVolumeAccessGroup",
      "AddKeyServerToProviderKmpip" ,
      "AddLdapClusterAdmin",
      "AddNodes",
      "AddVirtualNetwork",
      "AddVolumesToVolumeAccessGroup",
      "BreakSnapMirrorRelationship",
      "BreakSnapMirrorVolume",
      "CancelClone",
      "CancelGroupClone",
      "CheckpingOnVLAN" ,
      "CheckProposedCluster" ,

```

"CheckProposedNodeAddies" 、
"ClearClusterFaults",
"CloneMultipleVolumes",
"CloneVolume",
"CompleteClusterPairing",
"CompleteVolumePairing",
"CopyVolume",
"CreateBackupTarget",
" CreateClusterInterface 首选项 " 、
"CreateClusterSupportBundle",
"CreateGroupSnapshot",
" 创建 IDPDConfiguration " 、
"CreateInitiators",
" CreateKeyProviderKMIP " 、
" CreateKeyServerKMIP " 、
" CreatePublicPrivateKeyKeypair " 、
"CreateQoSPolicy",
"CreateSchedule",
"CreateSnapMirrorEndpoint",
"CreateSnapMirrorEndpointUnmanaged",
"CreateSnapMirrorRelationship",
"CreateSnapMirrorVolume",
"CreateSnapshot",
"CreateStorageContainer",
"CreateSupportBundle",
"CreateVolume",
"CreateVolumeAccessGroup",
"DeleteAllSupportBundles",
"DeleteAuthSession" 、
"DeleteAuthSessionSBYClusterAdmin" 、
"DeleteAuthSessionSBYUsername" 、
" 删除接口首选项 " 、
"DeleteGroupSnapshot",
"DeleteIDPDConfiguration" 、
"DeleteInitiators",
"DeleteKeyProviderkmp" 、
"DeleteKeyServerkmp" 、
"DeleteQoSPolicy",
"DeleteSnapMirrorEndpoints",
"DeleteSnapMirrorRelationships",
"DeleteSnapshot",
"DeleteStorageContainers",
"DeleteVolume",
"DeleteVolumeAccessGroup",
"DeleteVolumes",
"DisableAutoip",
"DisableBmcColdReset",
"DisableClusterSsh",
"DisableEncryptionAtRest",
" 禁用 IDPDAuthentication " 、
"DisableLdapAuthentication",
"DisableSnmp",
"EnableAutoip",
"EnableBmcColdReset",
"EnableClusterSsh",
"EnableEncryptionAtRest",
"EnableFeature",
" 启用 IDPAuthentication " 、
"EnableLdapAuthentication",

```
"EnableSnmp",
"GetAccountByID",
"GetAccountByName",
"GetAccountEfficiency",
"GetActiveToSciphers"、
"GetAsyncResult",
"GetBackupTarget",
"GetBinAssignmentProperties"、
"GetClientCertificateSignRequest"、
"GetClusterCapacity",
"GetClusterConfig",
"GetClusterFullThreshold",
"GetClusterHardwareInfo",
"GetClusterInfo",
"GetClusterInterface 首选项"、
"GetClusterMasterNodeID",
"GetClusterSshInfo",
"GetClusterState",
"GetClusterStats",
"GetClusterStructure",
"GetClusterVersionInfo",
"GetCompleteStats",
"GetConfig",
"GetCurrentClusterAdmin",
"GetDefaultQoS",
"GetDriveHardwareInfo",
"GetDriveStats",
"GetFeatureStatus",
"GetFipsReport",
"GetHardwareConfig",
"GetHardwareInfo",
"GetIDPAuthenticationState"、
"GetIpmiConfig",
"GetIpmiInfo",
"GetKeyProviderKmp"、
"GetKeyServerKMIP"、
"GetLdapConfiguration",
"GetLimits",
"getlldpinfo"、
"GetLoginBanner",
"GetLoginSessionInfo",
"GetNetworkConfig",
"GetNetworkInterface",
"GetNodefipsDriveReport"、
"GetNodeHardwareInfo",
"GetNodeStats",
"GetNtpInfo",
"GetNvramInfo",
"GetOntapVersionInfo",
"GetOrigin",
"GetPendingOperation",
"GetProtectionDomainLayout"、
"GetQoSPolicy",
"GetRawStats",
"GetRemoteLoggingHosts",
"GetSSLCertificate",
"GetSchedule",
"GetSnapMirrorClusterIdentity",
"GetSnmpACL",
"GetSnmpInfo",
```

```
"GetSnmpState",
"GetSnmpTrapInfo",
"GetStorageContainerEfficiency",
"getSupportedtLsciphers" 、
"GetSystemStatus",
"GetVirtualVolumeCount",
"GetVolumeAccessGroupEfficiency",
"GetVolumeAccessGroupLunAssignments",
"GetVolumeCount",
"GetVolumeEfficiency",
"GetVolumeStats",
"InitializeSnapMirrorRelationship",
"ListAccounts",
" ListActiveAuthSessions " 、
"ListActiveNodes",
"ListActivePairedVolumes",
"ListActiveVolumes",
"ListAllNodes",
"ListAsyncResults",
"ListAuthSessionSBYClusterAdmin" 、
"ListAuthSessionSBYUsername" 、
"ListBackupTargets",
"ListBulkVolumeJobs",
"ListClusterAdmins",
"ListClusterFaults",
"ListClusterInterface 自选设置 " 、
"ListClusterPairs",
"ListDeletedVolumes",
"ListDriveHardware",
"ListDriveStats",
"ListDrives",
"ListEvents",
"ListFibreChannelPortInfo",
"ListFibreChannelSessions",
"ListGroupSnapshots",
"ListISCSISessions",
" LISTIDPDCConfigurations" 、
"ListInitiators",
"ListKeyProviderSkmp" 、
"ListKeyServersKmp" 、
"ListNetworkInterfaces",
"ListNodeFibreChannelPortInfo",
"ListNodeStats",
"ListPendingActiveNodes",
"ListPendingNodes",
" ListProtectionDomainLevels " 、
"ListProtocolEndpoints",
"ListQoS Policies",
"ListSchedules",
"ListServices",
"ListSnapMirrorAggregates",
"ListSnapMirrorEndpoints",
"ListSnapMirrorLuns",
"ListSnapMirrorNetworkInterfaces",
"ListSnapMirrorNodes",
"ListSnapMirrorPolicies",
"ListSnapMirrorRelationships",
"ListSnapMirrorSchedules",
"ListSnapMirrorVolumes",
"ListSnapMirrorVservers",
```

```
"ListSnapshots",
"ListStorageContainers",
"ListSyncJobs",
"ListTests",
"ListUtilities",
"ListVirtualNetworks",
"ListVirtualVolumeBindings",
"ListVirtualVolumeHosts",
"ListVirtualVolumeTasks",
"ListVirtualVolumes",
"ListVolumeAccessGroups",
"ListVolumeStats",
"ListVolumeStatsByAccount",
"ListVolumeStatsByVirtualVolume",
"ListVolumeStatsByVolume",
"ListVolumeStatsByVolumeAccessGroup",
"ListVolumes",
"ListVolumesForAccount",
"ModifyAccount",
"ModifyBackupTarget",
"ModifyClusterAdmin",
"ModifyClusterFullThreshold",
"ModifyClusterInterface 首选项 "、
"ModifyGroupSnapshot",
"ModifyInitiators",
"ModifyKeyServerKMIP "、
"ModifyQoSPolicy",
"ModifySchedule",
"ModifySnapMirrorEndpoint",
"ModifySnapMirrorEndpointUnmanaged",
"ModifySnapMirrorRelationship",
"ModifySnapshot",
"ModifyStorageContainer",
"ModifyVirtualNetwork",
"ModifyVolume",
"ModifyVolumeAccessGroup",
"ModifyVolumeAccessGroupLunAssignments",
"ModifyVolumePair",
"ModifyVolumes",
"PurgeDeletedVolume",
"PurgeDeletedVolumes",
"QuiesceSnapMirrorRelationship",
"RemoveAccount",
"RemoveBackupTarget",
"RemoveClusterAdmin",
"RemoveClusterPair",
"RemoveDrives",
"RemoveInitiatorsFromVolumeAccessGroup",
"RemoveKeyServerFromProviderKmpip"、
"RemoveNodes",
"RemoveSSLCertificate",
"RemoveVirtualNetwork",
"RemoveVolumePair",
"RemoveVolumesFromVolumeAccessGroup",
"ResetDrives",
"ResetNetworkConfig"、
"ResetNode",
"ResetSupplementalTLSCIPHERs"、
"RestartNetworking",
"RestartServices",
```

```
"RestoreDeletedVolume" ,
"ResumeSnapMirrorRelationship" ,
"ResyncSnapMirrorRelationship" ,
"RollbackToGroupSnapshot" ,
"RollbackToSnapshot" ,
"SecureEraseDrives" ,
"SetClusterConfig" ,
"SetClusterStructure" ,
"SetConfig" ,
"SetDefaultQoS" ,
"SetLoginBanner" ,
"SetLoginSessionInfo" ,
"SetNetworkConfig" ,
"SetNtpInfo" ,
"SetProtectionDomainLayout " 、
"SetRemoteLoggingHosts" ,
"SetSSLCertificate" ,
"SetSnmpACL" ,
"SetSnmpInfo" ,
"SetSnmpTrapInfo" ,
"SetSupplementalTLSCIPPHERs" 、
"Shutdown" ,
"SnmpSendTestTraps" ,
"StartBulkVolumeRead" ,
"StartBulkVolumeWrite" ,
"StartClusterPairing" ,
"StartVolumePairing" ,
"TestAddressAvailability " 、
"TestConnectEnsemble" ,
"TestConnectMvip" ,
"TestConnectSvip" ,
"TestDrives" ,
"TestHardwareConfig" ,
"TestKeyProviderK mip" 、
"TestKeyServerKMIP " 、
"TestLdapAuthentication" ,
"TestLocalConnectivity" ,
"TestLocateCluster" ,
"TestNetworkConfig" ,
"TestPing" ,
"TestRemoteConnectivity" ,
"UpdateBulkVolumeStatus" ,
"updateidpdconfiguration" 、
"UpdateSnapMirrorRelationship"
],
"CurrentVersion" : "12.0" 、
"supportedVersions": [
    "1.0" ,
    "2.0" ,
    "3.0" ,
    "4.0" ,
    "5.0" ,
    "5.1" ,
    "6.0" ,
    "7.0" ,
    "7.1" ,
    "7.2" ,
    "7.3" ,
    "7.4" ,
    "8.0" ,
```

```

        "8.1",
        "8.2",
        "8.3",
        "8.4",
        "8.5",
        "8.6",
        "8.7",
        "9.0",
        "9.1",
        "9.2",
        "9.3",
        "9.4",
        "9.5",
        "9.6" 、
        "10.0",
        "10.1",
        "10.2",
        "10.3",
        "10.4" 、
        "10.5",
        "10.6" 、
        "10.7",
        "11.0" 、
        "11.1" 、
        "11.3",
        "11.5" 、
        "11.7" 、
        "11.8" 、
        "12.0"
    ]
}
}

```

GetAsyncResult

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

`GetAsyncResult` 以标准方式返回操作的总体状态（正在进行、已完成或错误），但为操作返回的实际数据取决于原始方法调用，并且系统会将返回数据与每个方法记录在一起。

如果 `keepResult` 参数缺失或错误，则返回结果后，`asyncHandle` 将变为非活动，并且以后尝试查询 `asyncHandle` 时会返回错误。通过将 `keepResult` 参数设置为 `true`，您可以使 `asyncHandle` 保持活动状态，以便于日后进行查询。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>asyncHandle</code>	从原始异步方法调用返回的一个值。	整型	无	是
<code>keepResult</code>	如果为 <code>true</code> ，则 <code>GetAsyncResult</code> 不会在返回异步结果后删除它，这样可以在日后查询 <code>asyncHandle</code> 时使用。	布尔型	<code>false</code>	否

返回值

此方法具有以下返回值：

名称	说明	类型
status	异步方法调用的状态。可能值： <ul style="list-style-type: none"> running: 该方法仍在运行。 complete: 该方法已完成，并返回了结果或错误。 	字符串
result	如果异步方法成功完成，则此结果为异步操作的结果。如果异步操作已失败，则此成员不存在。	字符串
error	如果状态为 complete 并且异步方法已失败，则此成员包含错误详细信息。如果异步操作已成功，则此成员不存在。	字符串
resultType	异步方法调用现在或过去执行的操作的类型。	字符串
details	如果状态为 running，则此成员包含与方法的当前操作相关的信息。如果异步方法未在运行，则此成员不存在。	JSON 对象
createTime	调用该异步方法时的时间（UTC+0 格式）。	ISO 8601 日期字符串
lastUpdateTime	上次更新异步方法的状态时的时间（UTC+0 格式）。	ISO 8601 日期字符串

注：GetAsyncResult 的返回值实质上是一个嵌套版本的标准 JSON 响应，外加一个状态字段。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetAsyncResult",
  "params": {
    "asyncHandle" : 389
  },
  "id" : 1
}
```

响应示例：方法错误

此方法返回类似于以下示例的响应：

```
{
  "error": {
    "code": 500,
    "message": "DBClient operation requested on a non-existent path at [/asyncresults/1]",
    "name": "xDBNoSuchPath"
  },
  "id": 1
}
```

如果“response”是 GetAsyncResult 调用返回的 JSON 响应对象，则“response.error”将对应于 GetAsyncResult 方法本身返回的错误（例如查询不存在的 asyncHandle）。

响应示例：异步任务错误

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "createTime": "2016-01-01T02:05:53Z",
    "error": {
      "bvID": 1,
      "message": "Bulk volume job failed",
      "name": "xBulkVolumeScriptFailure",
      "volumeID": 34
    },
    "lastUpdateTime": "2016-01-21T02:06:56Z",
    "resultType": "BulkVolume",
    "status": "complete"
  }
}
```

“response.result.error”与来自原始方法调用的错误结果相对应。

响应示例：异步任务成功

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "createTime": "2016-01-01T22:29:18Z",
    "lastUpdateTime": "2016-01-01T22:45:51Z",
    "result": {
      "cloneID": 25,
      "message": "Clone complete.",
      "volumeID": 47
    },
    "resultType": "Clone",
    "status": "complete"
  }
}
```

“response.result.result”是原始方法调用的返回值（如果该调用已成功完成）。

发布以下版本后新增

9.6

GetCompleteStats

NetApp 工程部会使用 `GetCompleteStats` API 方法测试新功能。从 `GetCompleteStats` 返回的数据不会进行文档记录且会经常更改，无法保证其准确性。您不应使用 `GetCompleteStats` 收集性能数据或与运行 `Element` 软件的存储集群的任何其他管理集成。

请使用以下受支持的 API 方法来检索统计信息：

- [GetVolumeStats](#)
- [GetClusterStats](#)
- [GetNodeStats](#)
- [GetDriveStats](#)

发布以下版本后新增

9.6

GetLimits

您可以使用 `GetLimits` 方法获取通过 API 设置的限制值。这些值可能因 Element 版本而异，但如果不更新系统，这些值就不会发生更改。了解由 API 设置的限制值有助于为面向用户的工具编写 API 脚本。

注：无论使用哪种 API 端点版本传递 `GetLimits` 方法，此方法都将返回当前软件版本的限制。

参数

此方法没有任何输入参数。

返回值

此方法将返回一个 JSON 对象，其中包含名称-值对（包括 API 限制）。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetLimits",
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "accountCountMax": 5000,
    "accountNameLengthMax": 64,
    "accountNameLengthMin": 1,
    "backupTargetNameLengthMax": 64,
    "backupTargetNameLengthMin": 1,
    "bulkVolumeJobsPerNodeMax": 8,
    "bulkVolumeJobsPerVolumeMax": 2,
    "  查伯查纳斯库特最大值 " : 15000 、
    "cloneJobsPerNodeMax": 8,
    "cloneJobsPerVirtualVolumeMax": 8,
    "cloneJobsPerVolumeMax": 2,
    "clusterAdminAccountMax": 5000,
    "clusterAdminInfoNameLengthMax": 1024,
    "clusterAdminInfoNameLengthMin": 1,
    "clusterPairsCountMax": 4,
    "fibreChannelVolumeAccessMax": 16384,
    "initiatorAliasLengthMax": 224,
    "initiatorCountMax": 10000,
    "initiatorNameLengthMax": 224,
    "initiatorsPerVolumeAccessGroupCountMax": 128,
    "iscsiSessionsFromFibreChannelNodesMax": 4096,
    "MaxAuthSessionsForCluster" : 1024 、
    "MaxAuthSessionSperUser" : 1024 、
    "NonokeClusterCountMax" : 100 、
  }
}
```

```
"NonokeClusterCountMin" : 3 、
"qosPolicyCountMax": 500,
"qosPolicyNameLengthMax": 64,
"qosPolicyNameLengthMin": 1,
"scheduleNameLengthMax": 244,
"secretLengthMax": 16,
"secretLengthMin": 12,
"snapMirrorEndpointIPAddressesCountMax": 64,
"snapMirrorEndpointsCountMax": 4,
"snapMirrorLabelLengthMax": 31,
"snapMirrorObjectAttributeValueInfoCountMax": 9900000,
"snapshotNameLengthMax": 255,
"snapshotsPerVolumeMax": 32,
"StorageNoSnapS德 ClusterCountMin" : 2 、
"VirtualVolumeCountMax" : 8000 、
"virtualVolumesPerAccountCountMax": 10000,
"volumeAccessGroupCountMax": 1000,
"volumeAccessGroupLunMax": 16383,
"volumeAccessGroupNameLengthMax": 64,
"volumeAccessGroupNameLengthMin": 1,
"volumeAccessGroupsPerInitiatorCountMax": 1,
"volumeAccessGroupsPerVolumeCountMax": 64,
"volumeBurstIOPSMax": 200000,
"volumeBurstIOPSMin": 100,
" 卷数最大值 " : 4000 、
"volumeMaxIOPSMax": 200000,
"volumeMaxIOPSMin": 100,
"volumeMinIOPSMax": 15000,
"volumeMinIOPSMin": 50,
"volumeNameLengthMax": 64,
"volumeNameLengthMin": 1,
"VolumeSizemax" : 17592186044416 、
"volumeSizeMin": 1000000000,
"volumesPerAccountCountMax": 2000,
"volumesPerGroupSnapshotMax": 32,
" 卷普遍性 AccessGroupCountMax" : 2000 、
" 证人 nobsoleClusterCountMax" : 4.
}
}
```

发布以下版本后新增

9.6

GetOrigin

您可以使用 GetOrigin 方法获取构建节点的原始证书。

参数

注: 如果没有原始证书, 此方法将返回 “null”。

此方法没有任何输入参数。

返回值

此方法返回供应商原始证书信息。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetOrigin",
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "integrator": "SolidFire",
  "<signature>": {
    "pubkey": [public key info],
    "version": 1,
    "data": [signature info]
  },
  "contract-id": "none",
  "location": "Boulder, CO",
  "organization": "Engineering",
  "type": "element-x"
}
]
```

发布以下版本后新增

9.6

GetRawStats

NetApp 工程部会使用 `GetRawStats` API 方法测试新功能。从 `GetRawStats` 返回的数据不会进行文档记录且会经常更改，无法保证其准确性。您不应使用 `GetRawStats` 收集性能数据或与运行 Element 软件的存储集群的任何其他管理集成。

请使用以下受支持的 API 方法来检索统计信息：

- [GetVolumeStats](#)
- [GetClusterStats](#)
- [GetNodeStats](#)
- [GetDriveStats](#)

发布以下版本后新增

9.6

ListAsyncResults

您可以使用 `ListAsyncResults` 列出系统上所有当前正在运行的和已完成的异步方法的结果。使用 `ListAsyncResults` 查询异步结果不会导致已完成的 `asyncHandle` 到期；您可以使用 `GetAsyncResult` 查询由 `ListAsyncResults` 返回的任何 `asyncHandle`。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
asyncResultTypes	<p>结果类型的可选列表。您可以使用此列表将结果限制为仅显示这些类型的操作。可能值：</p> <ul style="list-style-type: none"> • DriveAdd: 系统正在将驱动器添加到集群中的相关操作。 • BulkVolume: 卷之间的复制操作，例如备份或还原。 • Clone: 卷克隆操作。 • DriveRemoval: 系统正在从驱动器中复制数据以准备将其从集群中删除的相关操作。 • RtftiPendingNode: 系统正在将兼容软件安装到节点中以便稍后将其添加到集群中的相关操作。 	字符串数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
asyncHandles	序列化异步方法结果数组。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListAsyncResults",
  "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": {
    "asyncHandles": [
      {
        "asyncResultID": 47,
        "completed": true,
        "createTime": "2016-01-01T22:29:19Z",
        "data": {
          "cloneID": 26,
          "message": "Clone complete.",
          "volumeID": 48
        }
      }
    ]
  }
}
```

```
        "lastUpdateTime": "2016-01-01T22:45:43Z",  
        "resultType": "Clone",  
        "success": true  
    },  
    ...]  
}  
}
```

发布以下版本后新增

9.6

相关参考

[GetAsyncResult](#) on page 84

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

帐户方法

使用帐户方法可以添加、删除、查看和修改帐户及安全信息。

AddAccount

您可以使用 `AddAccount` 向系统添加新帐户。创建新帐户后，您还可以使用此方法在该帐户下创建新卷。为帐户指定的 CHAP 设置将应用于此帐户拥有的所有卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
username	此帐户的唯一用户名。（长度必须是 1 到 64 个字符。）	字符串	无	是
initiatorSecret	用于启动程序的 CHAP 密钥。此密钥的长度必须为 12 到 16 个字符，且不可破解。启动程序 CHAP 密钥必须唯一，且不得与目标 CHAP 密钥相同。如果未指定，将创建随机密钥。	字符串	无	否
targetSecret	用于目标的 CHAP 密钥（相互 CHAP 身份验证）。此密钥的长度必须为 12 到 16 个字符，且不可破解。目标 CHAP 密钥必须唯一，且不得与启动程序 CHAP 密钥相同。如果未指定，将创建随机密钥。	字符串	无	否
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
account	包含新创建帐户的相关信息的对象。	<i>account</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "AddAccount",
  "params": {
    "username" : "bobsmith",
    "initiatorSecret" : "168[#5A757ru268)",
    "targetSecret" : "tlt&lt;,8TUYa7bC",
    "attributes" : {
      "billingcode" : 2345
    }
  }
}
```

```
    },
    "id" : 1
  }
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{
  "id": 1,
  "result": {
    "account": {
      "accountID": 90,
      "attributes": {
        "billingcode": 2345
      },
      "initiatorSecret": "168[#5A757ru268)",
      "status": "active",
      "storageContainerID": "00000000-0000-0000-0000-000000000000",
      "targetSecret": "tlt&lt;,8TUYa7bC",
      "username": "bobsmith",
      "volumes": []
    },
    "accountID": 90
  }
}
```

发布以下版本后新增

9.6

GetAccountByID

您可以使用 GetAccountByID 获取有关具有给定帐户 ID 的帐户的详细信息。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
accountID	指定收集其详细信息的帐户。	整型	无	是

返回值

此方法具有以下返回值:

名称	说明	类型
account	帐户详细信息。	<i>account</i>

请求示例

此方法的请求类似于以下示例:

```
{
  "method": "GetAccountByID",
  "params": {
    "accountID" : 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" : {
    "account" : {
      "accountID" : 3,
      "attributes" : {},
      "initiatorSecret" : "initiatorsecret",
      "status" : "active",
      "targetSecret" : "initiatorsecret",
      "username" : "account3",
      "volumes" : [
        14,
        15,
        16
      ]
    }
  }
}
```

发布以下版本后新增

9.6

GetAccountByName

您可以使用 `GetAccountByName` 获取有关具有给定用户名的特定帐户的详细信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
username	帐户的用户名。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
account	帐户详细信息。	account

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetAccountByName",
  "params": {
    "username" : "jimmyd"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id" : 1,
  "result" : {
    "account" : {
      "accountID" : 3,
      "attributes" : {},
      "initiatorSecret" : "initiatorsecret",
      "status" : "active",
      "targetSecret" : "initiatorsecret",
      "username" : "jimmyd",
      "volumes" : [
        14,
        15,
        16
      ]
    }
  }
}
```

发布以下版本后新增

9.6

GetAccountEfficiency

您可以使用 `GetAccountEfficiency` 获取有关卷帐户的效率统计信息。此方法仅为作为参数而提供的帐户返回效率信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
accountID	指定为其返回效率统计信息的卷帐户。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
compression	通过数据压缩为帐户中的所有卷节省的空间量。用一个比率表示，其中值“1”表示已在未压缩的情况下存储了数据。	浮点型
deduplication	通过不复制帐户中所有卷的数据来节省的空间量。用一个比率表示。	浮点型
missingVolumes	无法在其中查询效率数据的卷。造成卷缺失的原因可能包括垃圾收集 (GC) 周期不到一小时、网络连接暂时性丢失或服务在 GC 周期之后重新启动。	整型数组
thinProvisioning	已用空间与已分配用于存储数据的空间量之比。用一个比率表示。	浮点型

名称	说明	类型
timestamp	垃圾收集 (GC) 后最后一次收集效率数据的时间 (采用 UTC+0 格式)。	ISO 8601 日期字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetAccountEfficiency",
  "params": {
    "accountID": 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": {
    "compression": 2.020468042933262,
    "deduplication": 2.042488619119879,
    "missingVolumes": [],
    "thinProvisioning": 1.010087163391013,
    "timestamp": "2014-03-10T14:06:02Z"
  }
}
```

发布以下版本后新增

9.6

ListAccounts

您可以使用ListAccounts获取存储租户帐户的完整列表、并提供可选的寻呼支持。元素帐户允许访问卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
includeStorageContainers	默认情况下，响应中包含虚拟卷存储容器。要排除存储容器，请设置为 false。	布尔型	true	否

名称	说明	类型	默认值	必需
startAccountID	要返回的起始帐户 ID。如果不存在具有此帐户 ID 的帐户，则按帐户 ID 顺序排列的下一个帐户将用作列表的第一个帐户。要逐页浏览该列表，请传递前一响应中最后一个帐户的帐户 ID + 1。	整型	无	否
limit	要返回的帐户对象的最大数量。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
accounts	帐户的列表。	account 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListAccounts",
  "params": {
    "startAccountID" : 0,
    "limit" : 1000
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "result" : {
    "accounts" : [
      {
        "accountID" : 25,
        "username" : "jimmyd",
        "status" : "active",
        "initiatorSecret" : "168[#5A757ru268)",
        "targetSecret" : "168[#5A757ru268)",
        "attributes" : {},
        "volumes" : [6,7,20]
      },
      {
        "accountID" : 26,
        "username" : "jamesw",
        "status" : "active",
        "initiatorSecret" : "initiatorsecret",
        "targetSecret" : "targetsecret",
        "attributes" : {
          "billingcode" : "1234R"
        }
      }
    ]
  }
}
```

```

        },
        "volumes" : [23,64]
    },
    ]
},
" id" : 1
}

```

发布以下版本后新增

9.6

ModifyAccount

您可以使用 `ModifyAccount` 方法修改现有帐户。

当您锁定某个帐户时，来自该帐户的所有现有连接都将立即终止。当您更改某个帐户的 CHAP 设置时，所有现有连接都将保持活动状态，而后续连接或重新连接将使用新的 CHAP 设置。要清除帐户的属性，请为属性参数指定 {}。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
accountID	要修改的帐户的帐户 ID。	整型	无	是
status	帐户状态： active: 帐户处于活动状态，允许连接。 locked: 帐户处于锁定状态，拒绝连接。	字符串	无	否
initiatorSecret	用于启动程序的 CHAP 密钥。此密钥的长度必须为 12-16 个字符，且不可破解。启动程序 CHAP 密钥必须唯一，且不得与目标 CHAP 密钥相同。	字符串	无	否
targetSecret	用于目标的 CHAP 密钥（相互 CHAP 身份验证）。此密钥的长度必须为 12-16 个字符，且不可破解。目标 CHAP 密钥必须唯一，且不得与启动程序 CHAP 密钥相同。	字符串	无	否
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
username	用于更改与帐户关联的用户名。（长度必须是 1 到 64 个字符。）	字符串	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例。在此示例中，通过为属性指定 {} 将其清除：

```
{
  "method": "ModifyAccount",
  "params": {
    "accountID" : 25,
    "status"    : "locked",
    "attributes" : {}
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

RemoveAccount

您可以使用 `RemoveAccount` 方法删除现有帐户。必须先使用 `DeleteVolume` 删除和清除与帐户相关的所有卷，然后才能删除此帐户。如果此帐户的相关卷仍处于待删除状态，则无法使用 `RemoveAccount` 删除此帐户。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
accountID	要删除的帐户的 ID。	整型	无	是

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": "RemoveAccount",
  "params": {
    "accountID" : 25
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[Delete Volume](#) on page 376

您可以使用 `DeleteVolume` 方法将活动卷标记为待删除。标记后的卷会在清理间隔过后被清除（永久删除）。

管理员 API 方法

您可以使用管理员 API 方法创建、修改、查看和删除存储集群管理员，以及为那些能够访问存储集群的管理员分配访问级别和权限。

AddClusterAdmin

您可以使用 AddClusterAdmin 方法添加新的集群管理员帐户。集群管理员可通过 API 和管理工具来管理集群。集群管理员是完全独立的，与标准租户帐户无关。

可以限制每个集群管理员仅能使用一部分 API。您应使用多个集群管理员帐户来管理不同的用户和应用程序。作为最佳实践，请为每个集群管理员授予所需的最少权限；这样可以减少凭据损坏造成的潜在影响。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
access	控制集群管理员可以使用的方法。	字符串数组	无	是
acceptEula	接受最终用户许可协议。设置为 true，可将集群管理员帐户添加到系统。如果省略或设置为 false，则该方法调用失败。	布尔型	无	是
attributes	名称/值对列表，采用 JSON 对象格式。	JSON 对象	无	否
password	用于对此集群管理员进行身份验证的密码。	字符串	无	是
username	此集群管理员的唯一用户名。长度必须介于 1 到 1024 个字符之间。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
clusterAdminID	新创建的集群管理员的集群管理员 ID。	整型

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "AddClusterAdmin",  
}
```

```
"params": {  
  "username": "joeadmin",  
  "password": "68!5Aru268)$",  
  "attributes": {},  
  "acceptEula": true,  
  "access": ["volumes", "reporting", "read"]  
},  
"id": 1  
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id":1,  
  "result" : {  
    "clusterAdminID": 2  
  }  
}
```

发布以下版本后新增

9.6

相关参考

[访问控制](#) on page 493

可用的元素 API 方法因您设置的访问类型而异。

GetCurrentClusterAdmin

您可以使用 `GetCurrentClusterAdmin` 方法返回当前主集群管理员的信息。主集群管理员是在创建集群之时创建的。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值:

名称	说明	类型
clusterAdmin	集群管理员的相关信息。	<i>clusterAdmin</i>

请求示例

此方法的请求类似于以下示例:

```
{  
  "method": "GetCurrentClusterAdmin",  
  "id" : 1  
}
```

响应示例

此方法返回类似于以下示例的响应:

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

```
    "clusterAdmin":  
      "access": [  
        "administrator"  
      ],  
      "attributes": null,  
      "authMethod": "Cluster"  
      "clusterAdminID": 1,  
      "username": "admin"  
    }  
  }  
}
```

发布以下版本后新增

10.0

GetLoginBanner

您可以通过 `GetLoginBanner` 方法获取用户登录到 `Element Web` 界面时看到的当前有效“使用条款”横幅。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
banner	“使用条款”横幅的当前文本。即使该横幅已禁用，此值也可以包含文本。	字符串
enabled	“使用条款”横幅的状态。可能值： <ul style="list-style-type: none">• <code>true</code>：登录 <code>Web</code> 界面时显示“使用条款”横幅。• <code>false</code>：登录 <code>Web</code> 界面时不显示“使用条款”横幅。	布尔型

请求示例

此方法的请求类似于以下示例：

```
{  
  "id": 3411,  
  "method": "GetLoginBanner",  
  "params": {}  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{  
  "id": 3411,  
  "result": {  
    "loginBanner": {  
      "banner": "Welcome to NetApp!",  
      "enabled": false  
    }  
  }  
}
```

发布以下版本后新增

10.0

ListClusterAdmins

您可以使用 `ListClusterAdmins` 方法返回集群的所有集群管理员的列表。

系统中可以存在多个具有不同权限级别的集群管理员帐户，但只能有一个主集群管理员。主集群管理员是在创建集群之时创建的管理员。在集群上设置 LDAP 系统时，也可以创建 LDAP 管理员。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
showHidden	显示隐藏的集群管理员用户，如 SNMP Admin 。	布尔型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
clusterAdmins	有关集群的所有现有集群管理员和 LDAP 管理员的信息。	<i>clusterAdmin</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListClusterAdmins",
  "params": {},
  "showhide": 正确
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id":1,
  "result":{
    "clusterAdmins":[
      {
        "access":[
          "administrator"
        ],
        "attributes":null,
        "authMethod":"Cluster",
        "clusterAdminID":1,
        "username":"admin"
      },
      {
        "access":[
          "read",
          "administrator"
        ]
      }
    ]
  }
}
```

```

        ],
        "attributes":{
        },
        "authMethod":"Ldap",
        "clusterAdminID":7,
        "username":"john.smith"
    },
    {
        "access":[
            "read",
            "administrator"
        ],
        "attributes":{ },
        "authMethod":"Ldap",
        "clusterAdminID":6,
        "username":"cn=admin1_jones,ou=ptusers,c=prodtest,dc=solidfire,dc=net"
    }
]
}
}

```

发布以下版本后新增

9.6

ModifyClusterAdmin

您可以使用ModifyClusterAdmin该方法更改集群管理员、LDAP 集群管理员或第三方身份认证提供者（IDP）集群管理员的设置。但是，不能更改管理员的集群管理员帐户的访问权限。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
access	控制此集群管理员可以使用的方法。	字符串数组	无	否
attributes	名称-值对的列表，采用JSON 对象格式。	JSON 对象	无	否
clusterAdminID	要修改的集群管理员、LDAP 集群管理员或 IDP 集群管理员的集群管理员。	整型	无	是
password	用于对此集群管理员进行身份验证的密码。此参数不适用于 LDAP 或 IDP 集群管理员。	字符串	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyClusterAdmin",
  "params": {
    "clusterAdminID" : 2,
    "password"      : "7925Brc429a"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[访问控制](#) on page 493

可用的元素 API 方法因您设置的访问类型而异。

RemoveClusterAdmin

您可以使用 `RemoveClusterAdmin` 该方法删除集群管理员、LDAP 集群管理员或第三方身份认证提供者（IDP）集群管理员。不能删除“admin”集群管理员帐户。

参数

如果移除了 IDP 集群管理员、并且该管理员已对与第三方身份认证提供者 IDP 关联的会话进行了身份验证、则这些会话将在其当前会话中注销或可能会丢失访问权限。访问权限丢失取决于已删除的 IDP 集群管理员是否与给定用户的 SAML 属性中的多个 IDP 集群管理员之一匹配。剩下的一组匹配 IDP 集群管理员将导致聚合访问权限减少。删除集群管理员时，其他集群管理员用户类型也会注销。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
clusterAdminID	要删除的集群管理员的集群管理员 ID。	整型	无	是

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": "RemoveClusterAdmin",
  "params": {
    "clusterAdminID" : 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" : { }
}
```

发布以下版本后新增

9.6

SetLoginBanner

您可以通过 SetLoginBanner 方法配置用户登录到 Element Web 界面时看到的“使用条款”横幅。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
banner	“使用条款”横幅的所需文本。允许的最大长度为 4,096 个字符。	字符串	无	否
enabled	“使用条款”横幅的状态。可能值： <ul style="list-style-type: none"> • true: 登录 Web 界面时显示“使用条款”横幅。 • false: 登录 Web 界面时不显示“使用条款”横幅。 	布尔型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
banner	“使用条款”横幅的当前文本。即使该横幅已禁用，此值也可以包含文本。	字符串
enabled	“使用条款”横幅的状态。可能值： <ul style="list-style-type: none"> • true: 登录 Web 界面时显示“使用条款”横幅。 • false: 登录 Web 界面时不显示“使用条款”横幅。 	布尔型

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 3920,
  "method": "SetLoginBanner",
  "params": {
    "banner": "Welcome to NetApp!",
    "enabled": true
  }
}
```

```
}  
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id": 3920,  
  "result": {  
    "loginBanner": {  
      "banner": "Welcome to NetApp!",  
      "enabled": true  
    }  
  }  
}
```

发布以下版本后新增

10.0

集群 API 方法

您可以使用 Element 软件集群 API 方法管理存储集群以及属于此存储集群的节点的配置和拓扑。

某些集群 API 方法在属于集群一部分的节点或已配置为加入集群的节点上运行。可以将节点添加到新集群或现有集群中。准备添加到集群中的节点处于“pending”状态，表示已配置这些节点但尚未将其添加到集群中。

AddNodes

您可以使用 AddNodes 方法将一个或多个新节点添加到集群中。

当节点在尚未配置的情况下首次启动时，系统会提示您配置该节点。配置该节点之后，该节点会注册为集群中的“待定节点”。运行 Element 软件的存储集群会自动根据集群上的版本创建节点映像。在添加待定节点时，此方法响应会包含一个 asyncHandle 值，您可以在 GetAsyncResult 方法中使用此值来查询自动映像过程的状态。

向集群中添加光纤通道节点的过程与添加 Element iSCSI 存储节点的过程相同。光纤通道节点会使用 NodeID 在系统中注册。节点变为可访问后，系统会将其置于“待定节点”状态。ListAllNodes 方法将返回 iSCSI 节点以及任何可添加到集群中的光纤通道节点的 pendingNodeID。

在将节点添加到为虚拟网络配置的集群时，系统需要足够数量的虚拟存储 IP 地址，以便为新节点分配虚拟 IP。如果没有可用于新节点的虚拟 IP 地址，AddNode 操作将失败。可以使用 ModifyVirtualNetwork 方法向虚拟网络添加更多存储 IP 地址。

添加了节点之后，该节点上的任何驱动器将变为可用，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。

注：添加新节点之后，该节点可能在几秒钟后才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：

名称	说明	类型	默认值	必需
pendingNodes	要添加的节点的 PendingNodeID。可以使用 ListPendingNodes 方法获取待定节点列表。	整型数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
nodes	将上一个“pendingNodeID”映射到“nodeID”的对象的数组。在添加运行不兼容软件版本的待定节点时，此数组响应会包含一个 asyncHandle 值，您可以在 GetAsyncResult 方法中使用此值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": {
    "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"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

相关参考

[AddDrives](#) on page 154

您可以使用 `AddDrives` 方法将一个或多个可用驱动器添加到集群中，使这些驱动器可以托管集群的一部分数据。

[GetAsyncResult](#) on page 84

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

[ListAllNodes](#) on page 129

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

[ModifyVirtualNetwork](#) on page 350

您可以使用 `ModifyVirtualNetwork` 方法更改现有虚拟网络的属性。

ClearClusterFaults

您可以使用 `ClearClusterFaults` 方法清除当前和以前检测到的故障的信息。可以清除已解决的和未解决的错误。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>faultTypes</code>	确定要清除的故障类型。可能值： <ul style="list-style-type: none"><code>current</code>：当前检测到但尚未解决的故障。<code>resolved</code>：以前检测到并已解决的故障。<code>all</code>：当前故障和已解决的故障。可以通过故障对象的“resolved”字段确定故障状态。	字符串	<code>resolved</code>	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" : {}  
}
```

发布以下版本后新增

9.6

CreateClusterInterfacePreference

与运行 Element 软件的存储集群集成的系统可通过 `CreateClusterInterfacePreference` 方法在此存储集群上创建和存储任意信息。此方法供内部使用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
name	集群接口首选项的名称。	字符串	无	是
value	集群接口首选项的值。	字符串	无	是

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": {}
}
```

发布以下版本后新增

11.0

DeleteClusterInterfacePreference

与运行 Element 软件的存储集群集成的系统可通过 DeleteClusterInterfacePreference 方法删除现有集群接口首选项。此方法供内部使用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
name	要删除的集群接口首选项的名称。	字符串	无	是

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": {}
}
```

发布以下版本后新增

11.0

EnableFeature

您可以使用 EnableFeature 方法启用默认处于禁用状态的集群功能（例如 VVol）。

参数

此方法具有以下输入参数。

注: 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将导致集群保护域功能仅在节点级别起作用。

名称	说明	类型	默认值	必需
feature	启用集群功能。可能值： <ul style="list-style-type: none"> • 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。 • vvols: 启用 Element 软件 VVol 集群功能。 	字符串	无	是

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": {}
}
```

发布以下版本后新增

9.6

GetClusterCapacity

可以使用 `GetClusterCapacity` 返回整个存储集群的容量测量结果概要。此方法将返回一些字段，可用于计算 Element Web UI 中显示的效率等级。可以在脚本中使用效率计算，以返回精简配置、重复数据删除、数据压缩和整体效率的效率等级。

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。

- $\text{thinProvisioningFactor} = (\text{nonZeroBlocks} + \text{zeroBlocks}) / \text{nonZeroBlocks}$
- $\text{deDuplicationFactor} = (\text{nonZeroBlocks} + \text{snapshotNonZeroBlocks}) / \text{uniqueBlocks}$
- $\text{compressionFactor} = (\text{uniqueBlocks} * 4096) / (\text{uniqueBlocksUsedSpace} * 0.93)$

整体效率等级计算

通过使用精简配置、重复数据删除和数据压缩效率计算的结果，可使用以下公式计算整体集群效率。

- $\text{efficiencyFactor} = \text{thinProvisioningFactor} * \text{deDuplicationFactor} * \text{compressionFactor}$

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>clusterCapacity</code>	存储集群的容量测量结果。	<i>clusterCapacity</i>

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
  }
}
```

发布以下版本后新增

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量指标。

注: 当集群达到块集群填充度的“Error”阶段时，随着集群接近“Critical”阶段，所有卷上的最大 IOPS 将呈线性减小至卷的最小 IOPS。这有助于防止集群达到块集群填充度的“Critical”阶段。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
blockFullness	当前为该集群计算所得的块填充度级别。 <ul style="list-style-type: none"> • stage1Happy: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage2Aware: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage3Low: 系统无法针对双非同步节点故障提供冗余数据保护。对应于 Web UI 中的Warning状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量为“Error”状态以下 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。无法创建任何新卷或克隆。对应于 Element UI 中的Error状态。 • stage5CompletelyConsumed: 完全消耗。集群处于只读状态，并保持 iSCSI 连接，但会暂停所有写入。对应于 Element UI 中的Critical状态。 	字符串
fullness	反映“blockFullness”和“metadataFullness”之间的最高填充度级别。	字符串
maxMetadataOverProvisionFactor	一个值，表示可以相对于可用空间量过度配置元数据空间的倍数。例如，如果有足够的元数据空间可用于存储容量为 100 TiB 的卷，并且此数字设置为 5，则可以创建容量为 500 TiB 的卷。	整型
metadataFullness	当前为该集群计算所得的元数据填充度级别。 <ul style="list-style-type: none"> • stage1Happy: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage2Aware: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage3Low: 系统无法针对双非同步节点故障提供冗余数据保护。对应于 Web UI 中的Warning状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量为“Error”状态以下 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。无法创建任何新卷或克隆。对应于 Element UI 中的Error状态。 • stage5CompletelyConsumed: 完全消耗。集群处于只读状态，并保持 iSCSI 连接，但会暂停所有写入。对应于 Element UI 中的Critical状态。 	字符串
sliceReserveUsedThresholdPct	错误情况。如果保留的片利用率大于此值，则会触发系统警报。	整型
stage2AwareThreshold	警告情况。为阶段 2 集群阈值级别设置的值。	整型
stage2BlockThresholdBytes	集群使用的、存在阶段 2 条件的字节数。	整型
stage2 metadataatathresholdbytes	集群使用的元数据字节数，其中存在 stage2 全满条件。	

名称	说明	类型
stage3BlockThresholdBytes	集群使用的存储字节数量，在该数量中将存在阶段 3 的完全性条件。	整型
stage3BlockThresholdPercent	为阶段 3 设置的百分比值。当达到此百分比时，将在警报日志中发布警告。	整型
stage3LowThreshold	错误情况。因集群容量较低而导致创建系统警报的阈值。	整型
stage3 metadatathresholdbytes	集群使用的元数据字节数，其中存在 stage3 全满条件。	整型
阶段 3 元数据库地址保持百分比	为元数据全满阶段 3 设置的百分比值。当达到此百分比时，将在警报日志中发布警告。	整型
stage4BlockThresholdBytes	集群使用的存储字节数量，在该数量中将存在阶段 4 的满状态。	整型
stage4CriticalThreshold	错误情况。为警告集群容量严重不足而创建系统警报的阈值。	整型
stage4 metadatathresholdbytes	集群使用的元数据字节数、其中存在 stage4 全满条件。	整型
stage5BlockThresholdBytes	集群使用的存储字节数量，在该数量中将存在阶段 5 的满状态。	整型
阶段 5 元数据库地址保持字节	集群使用的元数据字节数，在该字段中将存在阶段 5 的全满条件。	整型
sumTotalClusterBytes	集群的物理容量（以字节计算）。	整型
sumTotalMetadataClusterBytes	可用于存储元数据的空间总量。	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
    }
}

```

发布以下版本后新增

9.6

相关参考

[ModifyClusterFullThreshold](#) on page 142

您可以使用 `ModifyClusterFullThreshold` 方法更改存储集群达到一定容量利用率时系统生成事件的级别。可以使用阈值设置来指示系统生成警告之前可接受的已利用块存储数量。

GetClusterHardwareInfo

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。具体信息通常包含制造商、供应商、版本和其他相关的硬件标识信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
type	仅将以下类型之一的硬件信息包括在响应中。可能值： <ul style="list-style-type: none"> drives: 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: 同时将驱动器和节点信息包括在响应中。 如果忽略此参数，则假设类型为 all。	字符串	all	否

返回值

此方法具有以下返回值：

名称	说明	类型
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节点 ID。	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

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

相关参考

[GetClusterHardwareInfo](#) on page 499

GetClusterHardwareInfo 方法返回类似于以下示例的响应。

GetClusterInfo

您可以使用 GetClusterInfo 方法返回有关集群的配置信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
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": {},  
      "encryptionAtRestState": "disabled",  
      "ensemble": [  
        "10.10.21.23",
```

```

        "10.10.21.24",
        "10.10.21.25"
    ],
    "mvip": "10.10.1.57",
    "mvipInterface": "Bond1G",
    "mvipNodeID": 2,
    "mvipVlanTag": "120",
    "name": "Hulk",
    "repCount": 2,
    "svip": "10.10.21.57",
    "svipInterface": "Bond10G",
    "svipNodeID": 2,
    "svipVlanTag": "0",
    "uniqueID": "17nx",
    "uuid": "8d385a91-9acf-4371-8939-b29e085191fd"
}
}
}

```

发布以下版本后新增

9.6

GetClusterInterfacePreference

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

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
name	集群接口首选项的名称。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
preference	所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"  
      }  
    }  
  }  
}
```

发布以下版本后新增

11.0

GetClusterMasterNodeID

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
nodeID	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  
  }  
}
```

发布以下版本后新增

9.6

GetClusterStats

您可以使用 `GetClusterStats` 方法检索集群的活动测量结果概要。返回值从创建集群之后开始累计。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
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
    }
  }
}
```

发布以下版本后新增

9.6

GetClusterVersionInfo

您可以使用 `GetClusterVersionInfo` 方法检索有关在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：

名称	说明	类型
<code>nodeID</code>	节点的 ID。	整型
<code>nodeInternalRevision</code>	节点的内部软件版本。	字符串
<code>nodeVersion</code>	节点的软件版本。	字符串

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>clusterAPIVersion</code>	集群中的当前 API 版本。	字符串
<code>clusterVersion</code>	集群中当前运行的 Element 软件版本。	字符串
<code>clusterVersionInfo</code>	集群中节点的列表和每个节点的版本信息。	JSON 对象数组
<code>pendingClusterVersion</code>	如果存在，则此值表示当前要升级或还原到的集群软件版本。	字符串
<code>softwareVersionInfo</code>	升级状态。对象成员： <ul style="list-style-type: none"><code>currentVersion</code>: 节点上的当前软件版本。<code>nodeID</code>: 要从 <code>currentVersion</code> 升级到 <code>pendingVersion</code> 的节点的 ID。如果未在进行升级，则此字段为 0（零）。<code>packageName</code>: 要安装的软件包的名称。<code>pendingVersion</code>: 要安装的软件版本。<code>startTime</code>: 启动安装的日期和时间（UTC +0 格式）。	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": ""
    }
  }
}
```

发布以下版本后新增

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:

名称	说明	类型	默认值	必需
feature	集群功能的状态。如果不提供值，则系统将返回所有功能的状态。可能值： <ul style="list-style-type: none"> • Vvols: 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。 	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
功能	指示功能名称及其状态的功能对象数组。对象成员： <ul style="list-style-type: none"> • 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"  
    }  
  ]  
}
```

发布以下版本后新增

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 均有效的时段。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>loginSessionInfo</code>	包含身份验证的有效期的对象。返回的可能对象： <ul style="list-style-type: none"><code>timeout</code>: 此会话的超时和到期时间（分钟）。采用 <code>H:mm:ss</code> 格式。例如：1:30:00、20:00、5:00。无论所输入的超时采用哪种格式，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"  
      }  
    }  
  }
```

发布以下版本后新增

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。具体信息通常包含制造商、供应商、版本和其他相关的硬件标识信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
nodeID	正在请求硬件信息的节点的 ID。如果指定了光纤通道节点，则返回有关光纤通道节点的信息。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
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节点 ID。	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

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

相关参考

[GetNodeHardwareInfo \(光纤通道节点的输出\)](#) on page 536
光纤通道节点的 `GetNodeHardwareInfo` 方法返回类似于以下示例的响应。

[GetNodeHardwareInfo \(iSCSI 的输出\)](#) on page 535

iSCSI 的 `getNodeHardwareInfo` 方法返回类似于以下示例的响应。

GetNodeStats

您可以使用 `getNodeStats` 方法检索单个节点的活动测量结果概要。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
nodeID	指定将为其返回统计信息的节点的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
nodeStats	节点活动信息。	<i>nodeStats</i>

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
    }
  }
}
```

发布以下版本后新增

9.6

ListActiveNodes

您可以使用 `ListActiveNodes` 方法返回集群中当前处于活动状态的节点的列表。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
nodes	集群中的活动节点的列表。	<i>node</i> 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

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

相关参考

[ListActiveNodes](#) on page 549

`ListActiveNodes` 方法返回类似于以下示例的响应。

ListAllNodes

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
nodes	描述集群中的活动节点的对象列表。	<i>node</i>
pendingActiveNodes	描述集群中的待定活动节点的对象列表。	<i>pendingActiveNode</i> 数组
pendingNodes	描述集群中的待定节点的对象列表。	<i>pendingNode</i> 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": []
  }
}
```

发布以下版本后新增

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。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>bestPractices</code>	包括由次优系统配置触发的故障。可能值： <ul style="list-style-type: none"><code>true</code><code>false</code>	布尔型	无	否
<code>faultTypes</code>	确定返回的故障类型。可能值： <ul style="list-style-type: none"><code>current</code>：列出未解决的活跃故障。<code>resolved</code>：列出以前检测到并已解决的故障。<code>all</code>：列出当前故障和已解决的故障。您可以在 <code>fault</code> 对象的“resolved”成员中查看故障状态。	字符串	<code>all</code>	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>faults</code>	描述请求的集群故障的对象。	<i>fault</i>

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"
  }
]
}
```

发布以下版本后新增

9.6

ListClusterInterfacePreferences

与运行 Element 软件的存储集群集成的系统可以使用 ListClusterInterfacePreference 方法列出系统上存储的现有集群接口首选项。此方法供内部使用。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
preferences	存储集群上当前存储的集群接口对象的列表，其中每个对象都包含相关的首选项名称和值。	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"
      }
    ]
  }
}
```

发布以下版本后新增

11.0

ListEvents

您可以使用 `ListEvents` 方法列出在集群上检测到的事件，按从旧到新的顺序排列。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
driveID	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
endPublishTime	指定仅返回在当前时间之前发布的事件。	字符串	0	否
endReportTime	指定仅返回在当前时间之前报告的事件。	字符串	0	否
eventType	指定要返回的事件类型。请参见 event 了解可能的事件类型。	字符串	0	否
maxEvents	指定返回的最大事件数量。	整型	(无限制)	否
nodeID	指定仅返回具有此节点 ID 的事件。	整型		

名称	说明	类型	默认值	必需
serviceID	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开头。	整型	0	否
startPublishTime	指定仅返回在当前时间之后发布的事件。	字符串	0	否
startReportTime	指定仅返回在当前时间之后报告的事件。	字符串	0	否

返回值

此方法具有以下返回值：

名称	说明	类型
事件	事件的列表。	<i>event</i> 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":
```

[5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,24,25,26,27,28,29,30,31,40,41,42,43,44,45,46,47,5

```
2,53,54,55,56,57,58,59,60],
    "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"
}
]
}
}
```

发布以下版本后新增

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活动测量结果概要。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
nodeStats	存储节点活动信息。	<i>nodeStats</i>

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
        }
      ]
    }
  }
}
```

发布以下版本后新增

9.6

ListISCSISessions

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
sessions	有关每个 iSCSI 会话的信息。	<i>session</i>

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
      }
      ...
    ]
  }
}
```

发布以下版本后新增

9.6

ListServices

您可以使用 `ListServices` 方法列出正在集群上运行的节点、驱动器、当前软件及其他服务的服务信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
services	正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"
    }
}
]
```

发布以下版本后新增

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且已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` 不会列出对管理 IP 地址 (MIP) 和管理虚拟 IP 地址 (MVIP) 使用不同地址类型的待定节点。例如，如果某个待定节点具有一个 IPv6 MVIP 和一个 IPv4 MIP，则 `ListPendingNodes` 不会在结果中列出此节点。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
pendingNodes	集群中的待定节点的列表。	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"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

相关参考

[AddNodes](#) on page 109

您可以使用 `AddNodes` 方法将一个或多个新节点添加到集群中。

ListPendingActiveNodes

您可以使用 `ListPendingActiveNodes` 方法列出集群中处于“PendingActive”状态（介于待定和活动状态之间）的节点。处于此状态的节点正还原出厂映像。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>pendingActiveNodes</code>	用于详细介绍系统中所有 PendingActive 节点相关信息的对象列表。	<i>pendingActiveNode</i> 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"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ModifyClusterFullThreshold

您可以使用 `ModifyClusterFullThreshold` 方法更改存储集群达到一定容量利用率时系统生成事件的级别。可以使用阈值设置来指示系统生成警告之前可接受的已利用块存储数量。

例如，如果希望在系统达到“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”。当达到此级别时，系统会向集群管理控制台中的事件日志发送警报。

参数

此方法具有以下输入参数：

注：必须至少选择一个参数。

名称	说明	类型	默认值	必需
<code>maxMetadataOverProvisionFactor</code>	一个值，表示可以相对于可用空间量过度配置元数据空间的倍数。例如，如果有足够的元数据空间可用于存储容量为 100 TiB 的卷，并且此数字设置为 5，则可以创建容量为 500 TiB 的卷。	整型	5	否
<code>stage2AwareThreshold</code>	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
<code>stage3BlockThresholdPercent</code>	“Error”阈值以下的块存储利用率百分比，系统将在达到此级别之后触发集群“Warning”警报。	整型	无	否
<code>stage3MetadataThresholdPercent</code>	元数据存储利用率低于“错误”阈值的百分比会导致系统触发集群“警告”警报。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
blockFullness	当前为该集群计算所得的块填充度级别。 <ul style="list-style-type: none"> • stage1Happy: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage2Aware: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage3Low: 系统无法针对双非同步节点故障提供冗余数据保护。对应于 Web UI 中的Warning状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量为“Error”状态以下 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。无法创建任何新卷或克隆。对应于 Element UI 中的Error状态。 • stage5CompletelyConsumed: 完全消耗。集群处于只读状态，并保持 iSCSI 连接，但会暂停所有写入。对应于 Element UI 中的Critical状态。 	字符串
fullness	反映“blockFullness”和“metadataFullness”之间的最高填充度级别。	字符串
maxMetadataOverProvisionFactor	一个值，表示可以相对于可用空间量过度配置元数据空间的倍数。例如，如果有足够的元数据空间可用于存储容量为 100 TiB 的卷，并且此数字设置为 5，则可以创建容量为 500 TiB 的卷。	整型
metadataFullness	当前为该集群计算所得的元数据填充度级别。 <ul style="list-style-type: none"> • stage1Happy: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage2Aware: 不存在任何警报或错误情况。对应于 Web UI 中的Healthy状态。 • stage3Low: 系统无法针对双非同步节点故障提供冗余数据保护。对应于 Web UI 中的Warning状态。您可以在 Web UI 中配置此级别（默认情况下，系统会在容量为“Error”状态以下 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。无法创建任何新卷或克隆。对应于 Element UI 中的Error状态。 • stage5CompletelyConsumed: 完全消耗。集群处于只读状态，并保持 iSCSI 连接，但会暂停所有写入。对应于 Element UI 中的Critical状态。 	字符串
sliceReserveUsedThresholdPct	错误情况。在预留的分区利用率超过返回的 sliceReserveUsedThresholdPct 值时将触发系统警报。	整型
stage2AwareThreshold	警告情况。为“stage 2”集群阈值级别设置的值。	整型

名称	说明	类型
stage2BlockThresholdBytes	集群使用的、存在 stage2 全满条件的字节数。	整型
stage2 metadatathresholdbytes	集群使用的元数据字节数，其中存在 stage2 全满条件。	
stage3BlockThresholdBytes	集群使用的存储字节数量，在该数量中将存在阶段 3 的完全性条件。	整型
stage3BlockThresholdPercent	为阶段 3 设置的百分比值。当达到此百分比时，将在警报日志中发布警告。	整型
stage3LowThreshold	错误情况。因集群容量较低而导致创建系统警报的阈值。	整型
stage3 metadatathresholdbytes	集群使用的元数据字节数，其中存在 stage3 全满条件。	
stage4BlockThresholdBytes	集群使用的存储字节数量，在该数量中将存在阶段 4 的满状态。	整型
stage4CriticalThreshold	错误情况。为警告集群容量严重不足而创建系统警报的阈值。	整型
stage4 metadatathresholdbytes	集群使用的元数据字节数、其中存在 stage4 全满条件。	
stage5BlockThresholdBytes	集群使用的存储字节数量，在该数量中将存在阶段 5 的满状态。	整型
阶段 5 元数据库地址保持字节	集群使用的元数据字节数，在该字段中将存在阶段 5 的全满条件。	
sumTotalClusterBytes	集群的物理容量（以字节计算）。	整型
sumTotalMetadataClusterBytes	可用于存储元数据的空间总量。	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  
  }  
}
```

发布以下版本后新增

9.6

ModifyClusterInterfacePreference

与运行 Element 软件的存储集群集成的系统可通过 ModifyClusterInterfacePreference 方法更改现有集群接口首选项。此方法供内部使用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
name	要修改的集群接口首选项的名称。	字符串	无	是
value	集群接口首选项的新值。	字符串	无	是

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": {}
}
```

发布以下版本后新增

11.0

RemoveNodes

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

删除节点之前，您必须使用 `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迁移出某个节点之前，不能删除此节点。删除此节点之后，此节点会将其自身注册为待定节点。您可以重新添加此节点，或者将此节点关闭（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，则此方法可能会在返回响应之前发生超时。如果此方法调用无法删除此节点，请重新执行此方法调用。请注意，如果要将集群主节点与其他节点一起删除，应使用一个单独的调用来删除集群主节点本身。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>ignoreEnsembleToleranceChange</code>	删除节点时，请忽略对整体效果节点容错的更改。 如果存储群集使用的数据保护方案允许多个节点出现故障、并且删除节点会降低整体的节点故障容错能力、则节点删除通常会失败并出现错误。您可以将此参数设置为 <code>true</code> 、以禁用整体容差检查、以便成功删除节点。	布尔型	<code>false</code>	否
<code>nodes</code>	要删除的节点的节点 ID 列表。	整型数组	无	是

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" : {},  
}
```

发布以下版本后新增

9.6

SetLoginSessionInfo

您可以使用 SetLoginSessionInfo 方法设置登录身份验证对会话有效的时间段。当登录时间段已过并且系统上未进行任何活动之后，身份验证将过期。在登录时间段已过之后，需要使用新的登录凭据才能继续访问集群。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
timeout	集群身份验证的有效期。采用 HH:mm:ss 格式。例如，01:30:00、00:90:00 和 00:00:5400 均等于 90 分钟超时期限。	字符串	30 分钟	是

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" : {}  
}
```

发布以下版本后新增

9.6

Shutdown

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或多个节点，还可以关闭集群中的所有节点。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
nodes	要重新启动或关闭的节点的节点 ID 列表。	整型数组	无	是
option	要对集群采取的操作：可能值： <ul style="list-style-type: none">restart：重新启动集群。halt：执行完全关闭。	字符串	restart	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": "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
    ]
  }
}
```

发布以下版本后新增

9.6

集群创建 API 方法

您可以使用以下 API 方法来创建存储集群。需要针对单个节点上的 API 端点使用所有这些方法。

CheckProposedCluster

您可以在CheckProposedCluster使用一组存储节点创建存储集群之前使用该方法来测试这些节点、以确定可能出现的错误或故障、例如双节点存储集群不支持的不平衡混合节点功能或节点类型。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
nodes	构成存储集群的一组初始存储节点的存储 IP 地址列表。	字符串数组	无	是
force	设置为 true 可对存储集群中的所有存储节点运行。	布尔型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
proposedClusterValid	指示建议的存储节点是否会构成有效的存储集群。可能值： <ul style="list-style-type: none">• true• false	布尔型
proposedClusterErrors	使用建议的存储节点创建存储集群时可能发生的错误。	字符串数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CheckProposedCluster",
  "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": [ ]
  }
}
```

发布以下版本后新增

11.0

CreateCluster

您可以使用 `CreateCluster` 方法初始化集群中拥有“`mvip`”和“`svip`”地址的节点。每个新集群将使用该集群中第一个节点的管理 IP (MIP) 进行初始化。此方法还会自动添加要配置到集群中的所有节点。每次初始化新集群时，您只需要使用此方法一次。

注：在运行 `GetBootStrapConfig` 方法获取要包含在集群中的其余节点的 IP 地址后，可以对集群的主节点运行 `CreateCluster` 方法。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>acceptEula</code>	指示创建此集群时接受最终用户许可协议。要接受 EULA，请将此参数设置为 <code>true</code> 。	布尔型	无	是
<code>attributes</code>	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
<code>mvip</code>	集群在管理网络上的浮动（虚拟）IP 地址。	字符串	无	是
<code>nodes</code>	构成集群的初始节点集的 CIP/SIP 地址。该节点的 IP 必须位于此列表中。	字符串数组	无	是
<code>password</code>	集群管理员帐户的初始密码。	字符串	无	是
<code>repCount</code>	要存储在集群中的每个数据段的副本数。有效值为“2”。	整型	2	是
<code>svip</code>	集群在存储 (iSCSI) 网络上的浮动（虚拟）IP 地址。	字符串	无	是
<code>username</code>	集群管理员的用户名。	字符串	无	是

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": "CreateCluster",
  "params": {
    "acceptEula": true,
    "mvip": "10.0.3.1",
    "svip": "10.0.4.1",
    "repCount": 2,
    "username": "Admin1",
    "password": "9R7ka4rEPa2uREtE",
    "attributes": {
      "clusteraccountnumber": "axdf323456"
    },
    "nodes": [
      "10.0.2.1",
      "10.0.2.2",
      "10.0.2.3",
      "10.0.2.4"
    ]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[GetBootstrapConfig](#) on page 151

您可以使用 `GetBootstrapConfig` 方法从启动配置文件获取集群和节点信息。将单个节点加入到集群中之前，应在该节点上使用此 API 方法。创建集群时，您需要在集群配置界面中使用此方法所返回的信息。

GetBootstrapConfig

您可以使用 `GetBootstrapConfig` 方法从启动配置文件获取集群和节点信息。将单个节点加入到集群中之前，应在该节点上使用此 API 方法。创建集群时，您需要在集群配置界面中使用此方法所返回的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
clusterName	集群名称。	字符串
mvip	集群 MVIP 地址。如果节点不是群集的一部分，则为空。	字符串
nodeName	节点名称。	字符串
nodes	有关正在等待加入该集群的每个节点的信息列表。 可能值： <ul style="list-style-type: none"> • chassisType: (字符串) 节点的硬件平台。 • cip: (字符串) 节点的集群 IP 地址。 • compatible: (布尔值) 指示节点是否与其执行 API 调用的节点兼容。 • hostname: (字符串) 节点的主机名。 • mip: (字符串) 节点的 IPv4 管理 IP 地址。 • mipV6: (字符串) 节点的 IPv6 管理 IP 地址。 • nodeType: (字符串) 节点的型号名称。 • version: (字符串) 节点上当前安装的软件版本。 	JSON 对象数组
svip	集群 SVIP 地址。如果此节点不属于集群，则为 null。	字符串
version	此 API 方法调用的节点上当前安装的 Element 软件版本。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetBootstrapConfig",
  "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":{
    "clusterName":"testname",
    "nodeName":"testnode",
    "svip": "10.117.1.5",
    "mvip": "10.117.1.6",
    "nodes":[
      {
        "chassisType":"R630",
        "cip":"10.117.115.16",
        "compatible":true,
        "hostname":"NLABP1132",
        "mip":"10.117.114.16",
        "mipV6": "fd20:8b1e:b256:45a::16",
        "nodeType":"SF2405",

```

```
        "role": "Storage",  
        "version": "11.0"  
    },  
    {  
        "chassisType": "R630",  
        "cip": "10.117.115.17",  
        "compatible": true,  
        "hostname": "NLABP1133",  
        "mip": "10.117.114.17",  
        "mipV6": "fd20:8b1e:b256:45a::17",  
        "nodeType": "SF2405",  
        "role": "Storage",  
        "version": "11.0"  
    },  
    {  
        "chassisType": "R630",  
        "cip": "10.117.115.18",  
        "compatible": true,  
        "hostname": "NLABP1134",  
        "mip": "10.117.114.18",  
        "mipV6": "fd20:8b1e:b256:45a::18",  
        "nodeType": "SF2405",  
        "role": "Storage",  
        "version": "11.0"  
    }  
  ],  
  "version": "11.0"  
}  
}
```

发布以下版本后新增

9.6

相关参考

[CreateCluster](#) on page 150

您可以使用 `CreateCluster` 方法初始化集群中拥有“`mvip`”和“`svip`”地址的节点。每个新集群将使用该集群中第一个节点的管理 IP (MIP) 进行初始化。此方法还会自动添加要配置到集群中的所有节点。每次初始化新集群时，您只需要使用此方法一次。

驱动器 API 方法

您可以使用驱动器 API 方法添加和管理可供存储集群使用的驱动器。向存储集群添加存储节点或在现有存储节点中安装新驱动器后，这些驱动器便可添加到此存储集群中。

AddDrives

您可以使用 `AddDrives` 方法将一个或多个可用驱动器添加到集群中，使这些驱动器可以托管集群的一部分数据。

向集群添加存储节点或在现有节点中安装新驱动器后，新驱动器会标记为可用，您必须先通过 `AddDrives` 添加这些驱动器，然后才能使用它们。使用 `ListDrives` 方法可显示可添加的驱动器。添加驱动器时，系统会自动确定此驱动器的类型。

此方法为异步方法，在启动集群驱动器重新平衡过程之后会立即返回。但是，要使用新添加的驱动器重新平衡集群中的数据，则可能需要更长的时间；即使在 `AddDrives` 方法调用完成之后，此重新平衡操作仍会继续执行。您可以使用 `GetAsyncResult` 方法查询此方法返回的 `asyncHandle`。`AddDrives` 方法返回后，您可以使用 `ListSyncJobs` 方法查看使用新驱动器重新平衡数据的进度。

注：添加多个驱动器时，在单个 `AddDrives` 方法调用中添加所有驱动器比使用多个调用逐个添加驱动器更高效。这样可减少使集群的存储负载保持稳定所需平衡的数据量。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
drives	<p>关于要添加到集群的每个驱动器的信息。可能值：</p> <ul style="list-style-type: none"> <code>driveID</code>：要添加的驱动器的 ID（整型）。 <code>type</code>：要添加的驱动器的类型（字符串）。有效值为“slice”、“block”或“volume”。如果省略，则系统将分配正确的类型。 	JSON 对象数组	无	是（ <code>type</code> 为可选）

返回值

此方法具有以下返回值：

名称	说明	类型
asyncHandle	用于获取操作结果的句柄值。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 1,
  "method": "AddDrives",
  "params": {
    "drives": [
      {
        "driveID": 1,
```

```
        "type": "slice"
      },
      {
        "driveID": 2,
        "type": "block"
      },
      {
        "driveID": 3,
        "type": "block"
      }
    ]
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[GetAsyncResult](#) on page 84

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

[ListDrives](#) on page 158

您可以使用 `ListDrives` 方法列出集群的活动节点中的驱动器。此方法返回已作为卷元数据或块驱动器添加的驱动器，以及尚未添加但可用的驱动器。

[ListSyncJobs](#) on page 396

您可以使用 `ListSyncJobs` 方法获取有关在 `Element` 存储集群上运行的同步作业的信息。此方法可返回有关分区、克隆、块和远程同步作业的信息。

GetDriveHardwareInfo

您可以使用 `GetDriveHardwareInfo` 方法获取给定驱动器的所有硬件信息。具体信息通常包含制造商、供应商、版本和其他相关的硬件标识信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
driveID	请求的驱动器的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
result	为指定的 driveID 返回的硬件信息。	hardwareInfo

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetDriveHardwareInfo",
  "params": {
    "driveID": 5
  },
  "id" : 100
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id" : 100,
  "result" : {
    "driveHardwareInfo" : {
      "description" : "ATA Drive",
      "dev" : "8:80",
      "devpath" : "/devices/pci0000:40/0000:40:01.0/0000:41:00.0/host6/port-6:0/expander-6:0/port-6:0:4/end_device-6:0:4/target6:0:4/6:0:4:0/block/sdf",
      "driveSecurityAtMaximum" : false,
      "driveSecurityFrozen" : false,
      "driveSecurityLocked" : false,
      "logicalname" : "/dev/sdf",
      "product" : "INTEL SSDSA2CW300G3",
      "securityFeatureEnabled" : false,
      "securityFeatureSupported" : true,
      "serial" : "CVPR121400NT300EGN",
      "size" : "300069052416",
      "uuid" : "7e1fd5b9-5acc-8991-e2ac-c48f813a3884",
      "version" : "4PC10362"
    }
  }
}
```

发布以下版本后新增

9.6

相关参考

[ListDrives](#) on page 158

您可以使用 `ListDrives` 方法列出集群的活动节点中的驱动器。此方法返回已作为卷元数据或块驱动器添加的驱动器，以及尚未添加但可用的驱动器。

GetDriveStats

您可以使用 `GetDriveStats` 方法获取单个驱动器的活动测量结果概要。值从驱动器添加到集群中之后开始累计。部分值特定于块驱动器。运行此方法时，将返回块或元数据驱动器类型的统计数据。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
driveID	请求的驱动器的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
driveStats	具有指定 driveID 的驱动器的活动信息。	<i>driveStats</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetDriveStats",
  "params": {
    "driveID": 3
  },
  "id" : 1
}
```

响应示例（块驱动器）

此方法返回类似于以下块驱动器示例的响应：

```
{
  "id": 1,
  "result": {
    "driveStats": {
      "driveID": 10,
      "failedDieCount": 0,
      "lifeRemainingPercent": 99,
      "lifetimeReadBytes": 26471661830144,
      "lifetimeWriteBytes": 13863852441600,
      "powerOnHours": 33684,
      "readBytes": 10600432105,
      "readOps": 5101025,
      "reallocatedSectors": 0,
      "reserveCapacityPercent": 100,
      "timestamp": "2016-10-17T20:23:45.456834Z",
      "totalCapacity": 300069052416,
      "usedCapacity": 6112226545,
      "usedMemory": 114503680,
      "writeBytes": 53559500896,
    }
  }
}
```

```
        "writeOps": 25773919
      }
    }
  }
```

响应示例（卷元数据驱动器）

此方法返回类似于以下卷元数据驱动器示例的响应：

```
{
  "id": 1,
  "result": {
    "driveStats": {
      "activeSessions": 8,
      "driveID": 12,
      "failedDieCount": 0,
      "lifeRemainingPercent": 100,
      "lifetimeReadBytes": 2308544921600,
      "lifetimeWriteBytes": 1120986464256,
      "powerOnHours": 16316,
      "readBytes": 1060152152064,
      "readOps": 258826209,
      "reallocatedSectors": 0,
      "reserveCapacityPercent": 100,
      "timestamp": "2016-10-17T20:34:52.456130Z",
      "totalCapacity": 134994670387,
      "usedCapacity": null,
      "usedMemory": 22173577216,
      "writeBytes": 353346510848,
      "writeOps": 86266238
    }
  }
}
```

发布以下版本后新增

9.6

相关参考

[ListDrives](#) on page 158

您可以使用 `ListDrives` 方法列出集群的活动节点中的驱动器。此方法返回已作为卷元数据或块驱动器添加的驱动器，以及尚未添加但可用的驱动器。

ListDrives

您可以使用 `ListDrives` 方法列出集群的活动节点中的驱动器。此方法返回已作为卷元数据或块驱动器添加的驱动器，以及尚未添加但可用的驱动器。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
drives	列出集群中的驱动器。	<i>drive</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListDrives",
  "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" : {
    "drives" : [
      {
        "attributes" : {},
        "capacity" : 299917139968,
        "driveID" : 35,
        "nodeID" : 5,
        "serial" : "scsi-SATA_INTEL_SSDSA2CW6CVPR141502R3600FGN-part2",
        "slot" : 0,
        "status" : "active",
        "type" : "volume"
      },
      {
        "attributes" : {},
        "capacity" : 600127266816,
        "driveID" : 36,
        "nodeID" : 5,
        "serial" : "scsi-SATA_INTEL_SSDSA2CW6CVPR1415037R600FGN",
        "slot" : 6,
        "status" : "active",
        "type" : "block"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListDriveStats

您可以使用 `ListDriveStats` 方法列出集群中多个驱动器的活动测量结果概要。默认情况下，此方法将返回集群中所有驱动器的统计信息，这些测量结果是随着向集群添加驱动器而累计得出的。此方法返回的某些值适用于块驱动器，某些值适用于元数据驱动器。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
drives	要返回驱动器统计信息的驱动器 ID (driveID) 列表。如果省略此参数，则返回所有驱动器的测量结果。	整型数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
driveStats	列出每个驱动器的驱动器活动信息。	<i>driveStats</i> 数组
errors	此列表包含 driveID 以及相关错误消息。此参数始终存在，如果没有错误，则为空白。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 1,
  "method": "ListDriveStats",
  "params": {
    "drives": [22, 23]
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "driveStats": [
      {
        "driveID": 22,
        "failedDieCount": 0,
        "lifeRemainingPercent": 84,
        "lifetimeReadBytes": 30171004403712,
        "lifetimeWriteBytes": 103464755527680,
        "powerOnHours": 17736,
        "readBytes": 14656542,
        "readOps": 3624,
        "reallocatedSectors": 0,
        "reserveCapacityPercent": 100,
        "timestamp": "2016-03-01T00:19:24.782735Z",
        "totalCapacity": 300069052416,
        "usedCapacity": 1783735635,
        "usedMemory": 879165440,
        "writeBytes": 2462169894,
        "writeOps": 608802
      }
    ],
    "errors": [
      {
        "driveID": 23,
```

```

        "exception": {
            "message": "xStatCheckpointDoesNotExist",
            "name": "xStatCheckpointDoesNotExist"
        }
    }
}
]
}
}

```

发布以下版本后新增

9.6

相关参考

[GetDriveStats](#) on page 157

您可以使用 `GetDriveStats` 方法获取单个驱动器的活动测量结果概要。值从驱动器添加到集群中之后开始累计。部分值特定于块驱动器。运行此方法时，将返回块或元数据驱动器类型的统计数据。

RemoveDrives

您可以使用 `RemoveDrives` 方法主动删除集群中的驱动器。如果要减少集群容量或准备更换接近使用年限的驱动器，您可以使用此方法。在将驱动器从集群中删除之前，会将驱动器上的所有数据删除并迁移到集群中的其他驱动器上。

`RemoveDrives` 是异步方法。根据要删除的驱动器的总容量，迁移所有数据可能需要几分钟的时间。

如果要删除多个驱动器，只需使用一个 `RemoveDrives` 方法调用即可，而不是多次调用此方法，一次删除一个驱动器。这样可减少使集群的存储负载保持稳定所需平衡的数据量。

您还可以使用 `RemoveDrives` 删除状态为“failed”的驱动器。删除状态为“failed”的驱动器之后，该驱动器不会恢复到“available”或“active”状态。此驱动器不能再在集群中使用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
drives	列出要从集群中删除的 driveIDs。	整型数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
asyncHandle	用于获取操作结果的句柄值。	整型

请求示例

此方法的请求类似于以下示例：

```

{
    "method": "RemoveDrives",
    "params": {
        "drives" : [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" : {  
    "asyncHandle": 1  
  }  
}
```

发布以下版本后新增

9.6

相关参考

[GetAsyncResult](#) on page 84

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间,并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果,请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

[ListDrives](#) on page 158

您可以使用 `ListDrives` 方法列出集群的活动节点中的驱动器。此方法返回已作为卷元数据或块驱动器添加的驱动器,以及尚未添加但可用的驱动器。

SecureEraseDrives

您可以使用 `SecureEraseDrives` 方法从状态为“available”的驱动器中删除所有残留数据。如果要更换服务寿命即将结束且包含敏感数据的驱动器,您可以使用此方法。此方法会使用安全擦除单元命令将预先确定的模式写入驱动器中,并重置驱动器上的加密密钥。此异步方法可能需要几分钟的时间才能完成。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
drives	要安全擦除的驱动器 ID 列表。	整型数组	无	是

返回值

此方法具有以下返回值:

名称	说明	类型
asyncHandle	用于获取操作结果的句柄值。	整型

请求示例

此方法的请求类似于以下示例:

```
{  
  "method": "SecureEraseDrives",  
  "params": {  
    "drives" : [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" : {  
    "asyncHandle" : 1  
  }  
}
```

发布以下版本后新增

9.6

相关参考

[GetAsyncResult](#) on page 84

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

[ListDrives](#) on page 158

您可以使用 `ListDrives` 方法列出集群的活动节点中的驱动器。此方法返回已作为卷元数据或块驱动器添加的驱动器，以及尚未添加但可用的驱动器。

光纤通道 API 方法

您可以使用光纤通道 API 方法添加、修改或删除存储集群的光纤通道节点成员。

GetVolumeAccessGroupLunAssignments

您可以使用 `GetVolumeAccessGroupLunAssignments` 方法检索有关指定卷访问组的 LUN 映射的详细信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>volumeAccessGroupID</code>	用于返回信息的唯一卷访问组 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumeAccessGroupLunAssignments</code>	列出所有物理光纤通道端口或某个节点的一个端口。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetVolumeAccessGroupLunAssignments",
  "params": {
    "volumeAccessGroupID": 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" : {
    "volumeAccessGroupLunAssignments" : {
      "volumeAccessGroupID" : 5,
      "lunAssignments" : [
        {"volumeID" : 5, "lun" : 0},
        {"volumeID" : 6, "lun" : 1},
        {"volumeID" : 7, "lun" : 2},
        {"volumeID" : 8, "lun" : 3}
      ],
      "deletedLunAssignments" : [
        {"volumeID" : 44, "lun" : 44}
      ]
    }
  }
}
```

```
}  
}
```

发布以下版本后新增

9.6

ListFibreChannelPortInfo

您可以使用 `ListFibreChannelPortInfo` 方法列出有关光纤通道端口的信息。

此 API 方法可在各个节点上使用；要访问各个光纤通道节点，需要用户 ID 和密码。但是，如果 `force` 参数设置为 `true`，则可以在集群上使用此方法。如果在集群上使用，则会列出所有光纤通道接口。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>force</code>	设置为 <code>true</code> ，可在集群中的所有节点上运行。	布尔型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>fibreChannelPorts</code>	列出所有物理光纤通道端口或某个节点的一个端口。	fibreChannelPort 数组

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "ListFibreChannelPortInfo",  
  "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": {  
    "fibreChannelPortInfo": {  
      "5": {  
        "result": {  
          "fibreChannelPorts": [  
            {  
              "firmware": "7.04.00 (d0d5)",  
              "hbaPort": 1,  
              "model": "QLE2672",  
              "nPortID": "0xc70084",  
              "pciSlot": 3,  
              "serial": "BFE1335E03500",  
              "speed": "8 Gbit",
```

```
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:98:a3:41",
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:0a"
  },
  {
    "firmware": "7.04.00 (d0d5)",
    "hbaPort": 2,
    "model": "QLE2672",
    "nPortID": "0x0600a4",
    "pciSlot": 3,
    "serial": "BFE1335E03500",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:9c:71:01",
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:0b"
  },
  {
    "firmware": "7.04.00 (d0d5)",
    "hbaPort": 1,
    "model": "QLE2672",
    "nPortID": "0xc70044",
    "pciSlot": 2,
    "serial": "BFE1335E04029",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:98:a3:41",
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:08"
  },
  {
    "firmware": "7.04.00 (d0d5)",
    "hbaPort": 2,
    "model": "QLE2672",
    "nPortID": "0x060044",
    "pciSlot": 2,
    "serial": "BFE1335E04029",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:9c:71:01",
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:09"
  }
]
},
"6": {
  "result": {
    "fibreChannelPorts": [
      {
        "firmware": "7.04.00 (d0d5)",
        "hbaPort": 1,
        "model": "QLE2672",
        "nPortID": "0x060084",
        "pciSlot": 3,
        "serial": "BFE1335E04217",
        "speed": "8 Gbit",
        "state": "Online",
        "switchWwn": "20:01:00:2a:6a:9c:71:01",
```

```
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:02"
  },
  {
    "firmware": "7.04.00 (d0d5)",
    "hbaPort": 2,
    "model": "QLE2672",
    "nPortID": "0xc700a4",
    "pciSlot": 3,
    "serial": "BFE1335E04217",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:98:a3:41",
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:03"
  },
  {
    "firmware": "7.04.00 (d0d5)",
    "hbaPort": 1,
    "model": "QLE2672",
    "nPortID": "0xc70064",
    "pciSlot": 2,
    "serial": "BFE1341E09515",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:98:a3:41",
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:00"
  },
  {
    "firmware": "7.04.00 (d0d5)",
    "hbaPort": 2,
    "model": "QLE2672",
    "nPortID": "0x060064",
    "pciSlot": 2,
    "serial": "BFE1341E09515",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:9c:71:01",
    "wwnn": "5f:47:ac:c8:3c:e4:95:00",
    "wwpn": "5f:47:ac:c0:3c:e4:95:01"
  }
]
}
}
}
}
```

发布以下版本后新增

9.6

ListFibreChannelSessions

您可以使用 ListFibreChannelSessions 方法列出有关集群上的光纤通道会话的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
sessions	描述集群上的活动光纤通道会话的对象列表。	<i>session</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListFibreChannelSessions",
  "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" : [
      {
        "initiatorWWPN" : "21:00:00:0e:1e:14:af:40",
        "nodeID" : 5,
        "serviceID" : 21,
        "targetWWPN": "5f:47:ac:c0:00:00:00:10",
        "volumeAccessGroupID": 7
      },
      {
        "initiatorWWPN" : "21:00:00:0e:1e:14:af:40",
        "nodeID" : 1,
        "serviceID" : 22,
        "targetWWPN": "5f:47:ac:c0:00:00:00:11",
        "volumeAccessGroupID": 7
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListNodeFibreChannelPortInfo

您可以使用 `ListNodeFibreChannelPortInfo` 方法列出有关节点上的光纤通道端口的信息。

此 API 方法可在各个节点上使用；要访问各个光纤通道节点，需要用户 ID 和密码。如果在集群上使用，则会列出所有光纤通道接口。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
fibreChannelPorts	列出所有物理光纤通道端口或某个节点的一个端口。	<i>fibreChannelPort</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListNodeFibreChannelPortInfo",
  "params": {
    "nodeID": 5,
    "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": {
    "fibreChannelPorts": [
      {
        "firmware": "7.04.00 (d0d5)",
        "hbaPort": 1,
        "model": "QLE2672",
        "nPortID": "0xc7002c",
        "pciSlot": 3,
        "serial": "BFE1335E03500",
        "speed": "8 Gbit",
        "state": "Online",
        "switchWwn": "20:01:00:2a:6a:98:a3:41",
        "wwnn": "5f:47:ac:c8:35:54:02:00",
        "wwpn": "5f:47:ac:c0:35:54:02:02"
      },
      {
        "firmware": "7.04.00 (d0d5)",
        "hbaPort": 2,
        "model": "QLE2672",
        "nPortID": "0x06002d",
        "pciSlot": 3,
        "serial": "BFE1335E03500",
        "speed": "8 Gbit",
        "state": "Online",
        "switchWwn": "20:01:00:2a:6a:9c:71:01",
        "wwnn": "5f:47:ac:c8:35:54:02:00",
        "wwpn": "5f:47:ac:c0:35:54:02:03"
      },
      {
        "firmware": "7.04.00 (d0d5)",
        "hbaPort": 1,
        "model": "QLE2672",
        "nPortID": "0xc7002a",

```

```
    "pciSlot": 2,
    "serial": "BFE1335E04029",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:98:a3:41",
    "wwnn": "5f:47:ac:c8:35:54:02:00",
    "wwpn": "5f:47:ac:c0:35:54:02:00"
  },
  {
    "firmware": "7.04.00 (d0d5)",
    "hbaPort": 2,
    "model": "QLE2672",
    "nPortID": "0x06002a",
    "pciSlot": 2,
    "serial": "BFE1335E04029",
    "speed": "8 Gbit",
    "state": "Online",
    "switchWwn": "20:01:00:2a:6a:9c:71:01",
    "wwnn": "5f:47:ac:c8:35:54:02:00",
    "wwpn": "5f:47:ac:c0:35:54:02:01"
  }
]
}
```

发布以下版本后新增

9.6

ModifyVolumeAccessGroupLunAssignments

您可以使用 `ModifyVolumeAccessGroupLunAssignments` 方法为特定卷定义自定义 LUN 分配。

在卷访问组中，此方法仅更改在 `lunAssignments` 参数上设置的 LUN 值。所有其他 LUN 分配均保持不变。

对于卷访问组中的卷，LUN 分配值必须唯一。您不能在卷访问组中定义重复的 LUN 值。但是，您可以在不同的卷访问组中重复使用相同的 LUN 值。

注：有效的 LUN 值为 0 到 16383。如果传递的 LUN 值超过此范围，则系统会生成异常。存在异常时，系统不会对任何指定的 LUN 分配进行修改。



警告：如果针对某个具有活动 I/O 的卷更改 LUN 分配，则此 I/O 可能会中断。更改卷 LUN 分配之前，您应该更改服务器配置。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>volumeAccessGroupID</code>	将修改其 LUN 分配的唯一卷访问组 ID。	整型	无	是
<code>lunAssignments</code>	具有新分配的 LUN 值的卷 ID。	整型数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
volumeAccessGroupLunAssignments	包含已修改卷访问组 LUN 分配详细信息的对象。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyVolumeAccessGroupLunAssignments",
  "params": {
    "volumeAccessGroupID" : 218,
    "lunAssignments" : [
      { "volumeID" : 832, "lun" : 0 },
      { "volumeID" : 834, "lun" : 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": {
    "volumeAccessGroupLunAssignments": {
      "deletedLunAssignments": [],
      "lunAssignments": [
        {
          "lun": 0,
          "volumeID": 832
        },
        {
          "lun": 1,
          "volumeID": 834
        }
      ]
    },
    "volumeAccessGroupID": 218
  }
}
```

发布以下版本后新增

9.6

LDAP API 方法

可以使用轻型目录访问协议 (Lightweight Directory Access Protocol, LDAP) 对 Element 存储访问进行身份验证。使用本节中所述的 LDAP API 方法可以配置对存储集群的 LDAP 访问。

AddLdapClusterAdmin

您可以使用 AddLdapClusterAdmin 添加新的 LDAP 集群管理员用户。LDAP 集群管理员可以使用 API 和管理工具来管理集群。LDAP 集群管理员帐户与标准租户帐户完全独立且不相关。

参数

您还可以使用此方法添加已在 Active Directory® 中定义的 LDAP 组。授予此组的访问级别将传递给 LDAP 组中的各个用户。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
access	控制此集群管理员可以使用的方法。	字符串数组	无	是
acceptEula	接受最终用户许可协议。设置为 true, 可将集群管理员帐户添加到系统。如果省略或设置为 false, 则该方法调用失败。	布尔型	无	是
attributes	名称-值对的列表, 采用 JSON 对象格式。	JSON 对象	无	否
username	新 LDAP 集群管理员的可分辨用户名。	字符串	无	是

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": "AddLdapClusterAdmin",
  "params": { "username": "cn=mike jones,ou=ptusers,dc=prodtest,dc=solidfire,dc=net",
    "access": ["administrator", "read"]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[访问控制](#) on page 493

可用的元素 API 方法因您设置的访问类型而异。

EnableLdapAuthentication

您可以使用 `EnableLdapAuthentication` 方法配置 LDAP 目录连接，以便对集群进行 LDAP 身份验证。之后，LDAP 目录中的成员用户可以使用其 LDAP 凭据登录到存储系统。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
authType	标识要使用的用户身份验证方法。可能值： <ul style="list-style-type: none"> • DirectBind • SearchAndBind 	字符串	SearchAndBind	否
groupSearchBaseDN	树中用作组子树搜索起点的基础 DN。	字符串	无	否
groupSearchType	控制所使用的组搜索筛选器。可能值： <ul style="list-style-type: none"> • NoGroups: 无组支持。 • ActiveDirectory: 用户的所有 Active Directory 组的嵌套成员资格。 • MemberDN: MemberDN 样式的组（单层）。 	字符串	ActiveDirectory	否
serverURIs	以逗号分隔的 LDAP 或 LDAPS 服务器 URI 列表。可以使用冒号后跟端口号将自定义端口添加到 LDAP 或 LDAPS URI 的末尾。例如、URI "LDAP://1.2.3.4." 使用默认端口、URI "LDAPS://1.2.3.4.:123" 使用自定义端口 123。	字符串数组	无	是
userSearchBaseDN	树中用作子树搜索起点的基础 DN。使用 SearchAndBind 的 authType 时，需要此参数。	字符串	无	否

名称	说明	类型	默认值	必需
searchBindDN	为了对用户执行 LDAP 搜索而登录时所使用的完全限定 DN。该 DN 需要对 LDAP 目录具有读取访问权限。使用 SearchAndBind 的 authType 时，需要此参数。	字符串	无	是
searchBindPassword	用于搜索的 searchBindDN 帐户的密码。使用 SearchAndBind 的 authType 时，需要此参数。	字符串	无	是
userSearchFilter	查询 LDAP 服务器时要使用的 LDAP 搜索筛选器。此字符串应包含占位符文本 “%USERNAME%”，该文本将替换为身份验证用户的用户名。例如，(&(objectClass=person)(sAMAccountName=%USERNAME%)) 将使用 Active Directory 中的 sAMAccountName 字段匹配在集群登录时输入的用户名。使用 SearchAndBind 的 authType 时，需要此参数。	字符串	无	是
userDNTemplate	一个字符串模板，用于定义用户的完整可分辨名称 (DN) 的构造模式。此字符串应包含占位符文本 “%USERNAME%”，该文本将替换为身份验证用户的用户名。使用 DirectBind 的 authType 时，需要此参数。	字符串	无	是
groupSearchCustomFilter	用于返回用户组 DN 的 LDAP 筛选器，适用于 CustomFilter 搜索类型。此字符串可以根据需要包含占位符文本 %USERNAME% 和 %USERDN%，此文本可以替换为用户名和完整用户 DN。	字符串	无	是

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": "EnableLdapAuthentication",
  "params": {
    "authType": "SearchAndBind",
    "groupSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
    "groupSearchType": "ActiveDirectory",
    "searchBindDN": "SFReadOnly@prodtest.solidfire.net",
    "searchBindPassword": "zsw@#edcASD12",
    "sslCert": "",
    "userSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
    "userSearchFilter": "(&(objectClass=person)(sAMAccountName=%USERNAME%))",
    "serverURIs": [
      "ldaps://111.22.333.444",
      "ldap://555.66.777.888"
    ]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

DisableLdapAuthentication

您可以使用 `DisableLdapAuthentication` 方法禁用 LDAP 身份验证并删除所有 LDAP 配置设置。此方法不会删除已为用户或组配置的任何集群管理员帐户。禁用 LDAP 身份验证后，已配置为使用 LDAP 身份验证的集群管理员不能再访问群集。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "DisableLdapAuthentication",
  "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": {}  
}
```

发布以下版本后新增

9.6

GetLdapConfiguration

您可以使用 `GetLdapConfiguration` 方法获取集群上当前处于活动状态的 LDAP 配置。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值。

名称	说明	类型
ldapConfiguration	列出当前的 LDAP 配置设置。此 API 调用不会返回搜索帐户密码的纯文本。 注: 如果 LDAP 身份验证当前已禁用，则返回的所有设置为空白，但“authType”和“groupSearchType”除外，两者将分别设置为“SearchAndBind”和“ActiveDirectory”。	ldapConfiguration

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "GetLdapConfiguration",  
  "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": {  
    "ldapConfiguration": {  
      "authType": "SearchAndBind",  
      "enabled": true,  
      "groupSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",  
      "groupSearchCustomFilter": "",  
      "groupSearchType": "ActiveDirectory",  
      "searchBindDN": "SFReadOnly@prodtest.solidfire.net",  
      "serverURIs": [  

```

```

        "ldaps://111.22.333.444",
        "ldap://555.66.777.888"
    ],
    "userDNTemplate": "",
    "userSearchBaseDN": "dc=prodtest,dc=solidfire,dc=net",
    "userSearchFilter": "(&(objectClass=person)(sAMAccountName=%USERNAME%))"
}
}
}

```

发布以下版本后新增

9.6

TestLdapAuthentication

您可以使用 TestLdapAuthentication 方法验证当前已启用的 LDAP 身份验证设置。如果配置正确，API 调用将返回测试用户的组成员资格。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
username	要测试的用户名。	字符串	无	是
password	要测试的用户名的密码。	字符串	无	是
ldapConfiguration	要测试的 ldapConfiguration 对象。如果提供此参数，即使 LDAP 身份验证当前已禁用，系统也会测试提供的配置。	LdapConfiguration	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
groups	包括测试用户作为成员的 LDAP 组列表。	数组
userDN	测试用户的完整 LDAP 可分辨名称。	字符串

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "TestLdapAuthentication",
  "params": {
    "username": "admin1",
    "password": "admin1PASS"
  },
  "id": 1
}

```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "groups": [
      "CN=StorageMgmt,OU=PTUsers,DC=prodtest,DC=solidfire,DC=net"
    ],
    "userDN": "CN=Admin1 Jones,OU=PTUsers,DC=prodtest,DC=solidfire,DC=net"
  }
}
```

发布以下版本后新增

9.6

多因素身份验证 API 方法

您可以使用多因素身份验证（MFA）通过安全声明标记语言（SAML）使用第三方身份认证提供者（IDP）管理用户会话。

AddDdpClusterAdmin

您可以使用AddIpdcClusterAdmin该方法添加由第三方身份认证提供者（IDP）验证的集群管理员用户。IDP 集群管理帐户是根据与用户关联的 IDP SAML 声明中提供的 SAML 属性值信息进行配置的。如果用户成功地通过 IDP 进行身份验证、并且在 SAML 声明中具有与多个 IDP 群集管理帐户匹配的 SAML 属性语句、则用户将具有这些匹配的 IDP 群集管理帐户的组合访问级别。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
access	控制此 IDP 集群管理员可以使用的方法。	字符串数组	无	是
acceptEula	接受最终用户许可协议。设置为 true，可将集群管理员帐户添加到系统。如果省略或设置为 false，则该方法调用失败。	布尔型	无	是
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
username	与 IDP 集群管理员的 SAML attribute-value 映射（例如，email=test@example.com）。可以使用 NameID 或 SAML 属性语句中的条目（如 EDUperson 联合）使用特定的 SAML 主题来定义此属性。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
clusterAdminID	新创建的集群管理员的唯一标识符。	整型

请求示例

此方法的请求类似于以下示例：

```
{  
  "方法" : "AddDdpClusterAdmin" ,  
  "params": {  
    "用户名" : "电子邮件 = test@example.com" ,  
    "acceptEula": true ,  
    "访问" : ["管理员"]  
  }  
}
```

```
}
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "result": {
    "ClusterAdminista" : 13.
  }
}
```

发布以下版本后新增

12.0

CreateIDPConfiguration

您可以使用CreateIpdcConfiguration该方法为使用第三方身份认证提供者（IDP）进行集群创建潜在的信任关系。IDP 通信需要 SAML 服务提供者证书。此证书是根据需要生成的、并由此 API 调用返回。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
idpMetadata	要存储的 IDP 元数据。	字符串	无	是
idpName	用于标识 SAML 2.0 单点登录的 IDP 提供者的名称。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
IDPConfigInfo	有关第三方身份认证提供者（IDP）配置的信息。	<i>IDPConfigInfo</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "CreateIDPConfiguration" ,
  "params": {
    "IDPmetadata" : "<?XML version = \"1.0\" encoding= \"UTF-8 \" ? >
      < 实体描述符
        xmlns = \" URN:OASIS : 名称: TC:SAML : 2.0 : 元数据 \"
        xmlns : ds=\"http://www.w3.org/2000/09/xmldsig#\"
        xmlns : shibmd=\"urn:mace:shibboleth:metadata:1.0\"
        xmlns : xml=\"http://www.w3.org/xml/1998/nameespace\"
        ...</organization>
      </entitydescriptor> " ,
    "idpname" : "https://provider.name.url.com"
  },
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "result": {
    "IDPConfigInfo": {
      "enabled": false,
      "IDPConfigurationID": " F983C602-12F9-4C67-B214-BF505185 CFED ",
      "IDPmetadata": "<?XML version=\"1.0\" encoding=\"UTF-8 \"?>\r\n
      <entitydescriptor xmlns=\"URN:OASOASIS : 名称: tc:SAML : 2.0 : 元数据 \"\r\n
      xmlns : ds=\"http://www.w3.org/2000/09/xmlldsig#\" \r\n
      xmlns : shibmd=\"urn:mace:shibboleth:metadata:1.0\" \r\n
      xmlns : xml=\"http://www.w3.org/xml/1998/namespace\" \r\n
      ... </organization> \r\n
      </entitydescriptor> " ,
      "IDPName": "https://privider.name.url.com" ,
      "ServiceProviderCertificate": "--begin Certificate --\n
      MIID...SLBIH\n
      ---- 结束证书 --\n" ,
      "spmetadataurl": "https://10.193.100100/auth/ui/Saml2"
    }
  }
}
```

发布以下版本后新增

12.0

DeleteAuthSession

您可以使用该方法删除DeleteAuthSession单个身份验证会话。如果调用用户不在clusteradmins/administrator accessgroup 中、则只能删除属于调用用户的 auth 会话。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
sessionID	要删除的验证会话的唯一标识符。	UUID	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
session	删除验证会话的会话信息。	AuthSessionInfo

请求示例

此方法的请求类似于以下示例：

```
{
  "方法": "DeleteAuthSession" ,
  "params": {
    "SessionID": "A862A8BB-2C5B-4774-A592-2148E2304713"
  }
}
```

```
    "id": 1
  }
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{
  "id": 1,
  "result": {
    "会话": {
      "AccessGroupList": [
        "administrator"
      ],
      "authmethod": "集群",
      "ClusterAdministsds": [
        1
      ],
      "FinalTimeout": "2020-04-09t17:51:30 z",
      "IDPConfigVersion": 0,
      "lastAccessTimeout": "2020-04-06T18:21:33Z",
      "SessionCreationTime": "2020-04-06T17:51:30 Z",
      "SessionID": "A862A8BB-2C5B-4774-A592-2148E2304713",
      "username": "admin"
    }
  }
}
```

发布以下版本后新增

12.0

DeleteAuthSessionSBYClusterAdmin

可以使用DeleteAuthSessionsByClusterAdmin该方法删除与指定的 ClusterAdministaD 关联的所有身份验证会话。如果指定的 ClusterAdminID 映射到一组用户，则该组所有成员的所有身份验证会话都将被删除。要查看可能要删除的会话列表、请将 ListAuthSessionsByClusterAdmin方法与 ClusterAdminID 参数一起使用。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
clusterAdminID	集群管理员的唯一标识符。	整型	无	是

返回值

此方法具有以下返回值:

名称	说明	类型
sessions	已删除验证会话的会话信息。	AuthSessionInfo

请求示例

此方法的请求类似于以下示例:

```
{
  "方法": "DeleteAuthSessionsByClusterAdmin",
}
```

```
"params": {  
  "ClusterAdminista" : 1.  
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "sessions": [  
    {  
      "AccessGroupList" : [  
        "administrator"  
      ],  
      "authmethod" : " 集群 " 、  
      "ClusterAdministds" : [  
        1  
      ],  
      "FinalTimeout" : "2020-03-14T19:21:24Z" 、  
      "IDPConfigVersion" : 0 、  
      "lastAccessTimeout" : "2020-03-11T19 : 51:24z" 、  
      "SessionCreationTime" : "2020-03-11T19 : 21 : 24Z" 、  
      "SessionID" : "B12BFC64-F233-44DF-8B9F-6FB6C011ABF7" 、  
      "username": "admin"  
    }  
  ]  
}
```

发布以下版本后新增

12.0

DeleteAuthSessionSBYUserName

您可以使用DeleteAuthSessionsByUsername该方法删除给定用户的所有身份验证会话。不在 AccessGroup ClusterAdmins/Administrator 中的调用方只能删除自己的会话。具有 ClusterAdmins/Administrator 权限的调用方可以删除属于任何用户的会话。要查看可以删除的会话列表, 请使用具有相同参数的 ListAuthSessionSBYUserName。要查看可能删除的会话列表、请使用ListAuthSessionsByUsername具有相同参数的方法。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
authMethod	要删除的用户会话的身份验证方法。只有 ClusterAdmins/Administrator AccessGroup 中的调用方可以提供此参数。可能的值包括： <ul style="list-style-type: none"> • authMethod=cluster 指定 clusteradmin 用户名。 • AuthMethod=LDAP 指定用户的 LDAP DN 。 • AuthMethod=IdP 指定用户的 IDP UUID 或 NameID 。如果未将 IDP 配置为返回任一选项、则指定创建会话时发出的随机 UUID 。 	authMethod	无	否
username	用户的唯一标识符。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
sessions	已删除验证会话的会话信息。	AuthSessionInfo

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "DeleteAuthSessionSBYUsername" 、
  "params": {
    "authmethod" : " 集群 " 、
    "username": "admin"
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "sessions": [
    {
      "AccessGroupList" : [
        "administrator"
      ],
      "authmethod" : " 集群 " 、
      "ClusterAdministds" : [
        1
      ],
      "FinalTimeout" : "2020-03-14T19:21:24Z" 、
      "IDPConfigVersion" : 0 、
      "lastAccessTimeout" : "2020-03-11T19 : 51:24z" 、
      "SessionCreationTime" : "2020-03-11T19 : 21 : 24Z" 、
      "SessionID" : "B12BFC64-F233-44DF-8B9F-6FB6C011ABF7" 、
    }
  ]
}
```

```
      "username": "admin"
    }
  ]
}
```

发布以下版本后新增

12.0

DeleteIDPConfiguration

您可以使用DeleteIdpConfiguration该方法删除集群的第三方 IDP 的现有配置。删除最后一个 IDP 配置将从集群中删除 SAML 服务提供者证书。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
idpConfigurationID	第三方 IDP 配置的 UUID。	UUID	无	否
idpName	用于标识和检索 SAML 2.0 单点登录的 IDP 提供者的名称。	字符串	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "方法": "DeleteIDPConfiguration",
  "params": {
    "IDPConfigurationID": "F983C602-12F9-4C67-B214-BF505185 CFED",
    "idpname": "https://provider.name.url.com"
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "结果": {}
}
```

发布以下版本后新增

12.0

DisableIDDPAuthentication

您可以使用DisableIdpAuthentication该方法禁用对使用第三方 IDP 进行集群身份验证的支持。禁用后、由第三方 IDP 身份验证的用户将无法再访问该集群、并且任何活动的已验证会话都将失效 / 断开连接。LDAP 和集群管理员可以通过支持的 UI 访问集群。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{  
  "方法" : "DisableIDDPAuthentication" ,  
  "params": {}  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{  
  "result": {}  
}
```

发布以下版本后新增

12.0

启用 IDPAuthentication

您可以使用EnableIdpAuthentication该方法为集群使用第三方 IDP 启用身份验证支持。启用 IDP 身份验证后、LDAP 和集群管理员将无法再通过支持的 UIS 访问集群、并且任何活动的身份验证会话都将失效 / 断开连接。只有经过第三方 IDP 身份验证的用户才能通过支持的 UIS 访问集群。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
idpConfigurationID	第三方 IDP 配置的 UUID 。如果仅存在一个 IDP 配置，则默认情况下启用该配置。 如果只有一个 IDPConfiguration、则不需要提供idpConfigurationID该参数。	UUID	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{  
  " 方法 " : "EnableIDPAuthentication " 、  
  "params": {  
    " IDPConfigurationID " : " F983C602-12F9-4C67-B214-BF505185 CFED " 、  
  }  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{  
  "result": {}  
}
```

发布以下版本后新增

12.0

GetIDPAuthenticationState

您可以使用GetIdpAuthenticationState该方法返回有关使用第三方 IDP 的身份验证状态的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
enabled	指示是否启用第三方 IDP 身份验证。	布尔型

请求示例

此方法的请求类似于以下示例：

```
{  
  " 方法 " : "GetIDPAuthenticationState"  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{  
  " 结果 " : {" 已启用 " : true }  
}
```

发布以下版本后新增

12.0

ListActiveAuthSessions

您可以使用ListActiveAuthSessions该方法列出所有活动的已验证会话。只有具有管理访问权限的用户才能调用此方法。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
sessions	验证会话的会话信息。	AuthSessionInfo

请求示例

此方法的请求类似于以下示例：

```
{  
  " 方法 " : "ListActiveAuthSessions"  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{  
  "sessions": [  
    {  
      "AccessGroupList" : [  
        "administrator"  
      ],  
      "authmethod" : " 集群 " 、  
      "ClusterAdministds" : [  
        1  
      ],  
      "FinalTimeout" : "2020-03-14T19:21:24Z" 、  
      "IDPConfigVersion" : 0 、  
      "lastAccessTimeout" : "2020-03-11T19 : 51:24z" 、  
      "SessionCreationTime" : "2020-03-11T19 : 21 : 24Z" 、  
      "SessionID" : "B12BFC64-F233-44DF-8B9F-6FB6C011ABF7" 、  
      "username": "admin"  
    }  
  ]  
}
```

发布以下版本后新增

12.0

LISTIDPDConfigurations

您可以使用ListIdpConfigurations该方法列出第三方 IDP 的配置。或者，您可以提供 EnableDOnly 标志来检索当前启用的 IDP 配置、或者提供 IDP 元数据 UUID 或 IDP 名称来查询特定 IDP 配置的信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
enabledOnly	过滤结果以返回当前启用的 IDP 配置。	布尔型	无	否
idpConfigurationID	第三方 IDP 配置的 UUID。	UUID	无	否
idpName	检索特定 IDP 名称的 IDP 配置信息。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
IDPConfigInfoOS	有关第三方 IDP 配置的信息。	IDPConfigInfo 阵列

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "listidpconfigures" ,
  "params": {}
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "result": {
    "IDPConfigInfo" : {
      "enabled": true,
      "IDPConfigurationID" : " F983C602-12F9-4C67-B214-BF505185 CFED " ,
      "IDPmetadata" : "<?XML version=\"1.0\" encoding=\"UTF-8 \"?>\r\n
      <entitydescriptor xmlns=\"URN:OASOASIS : 名称: tc:SAML : 2.0 : 元数据 \">\r\n
      xmlns : ds=\"http://www.w3.org/2000/09/xmlldsig#\"\r\n
      xmlns : shibmd=\"urn:mace:shibboleth:metadatal:1.0\"\r\n
      xmlns : xml=\"http://www.w3.org/xml/1998/namespace\"\r\n
      ...</organization> \r\n
      </entitydescriptor> " ,
      "IDPName" : "https://priver.name.url.com" ,
      "ServiceProviderCertificate" : "--begin Certificate --\n
      Mi . . . bhi \n
      ---- 结束证书 --\n" ,
      "spmetadataurl" : "https://10.193.100100/auth/ui/Saml2"
    }
  }
}
```

发布以下版本后新增

12.0

更新 IDEDPConfiguration

您可以使用UpdateIdpConfiguration该方法为集群使用第三方 IDP 更新现有配置。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
generateNewCertificate	如果指定为 true 、则会生成新的 SAML 密钥和证书并替换现有的对。注：更换现有证书将中断集群和 IDP 之间建立的信任关系，直到在 IDP 上重新加载集群的服务提供商元数据。如果未提供或设置为 false ， 则 SAML 证书和密钥将保持不变。	布尔型	无	否
idpConfigurationID	第三方 IDP 配置的 UUID 。	UUID	无	否
idpMetadata	用于 SAML 2.0 单点登录的配置和集成详细信息的 IDP 元数据。	字符串	无	否
idpName	用于标识和检索 SAML 2.0 单点登录的 IDP 提供者的名称。	字符串	无	否
newIdpName	如果指定了该名称，则该名称将替换旧的 IDP 名称。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
IDPConfigInfo	有关第三方 IDP 配置的信息。	IDPConfigInfo

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "updateidpdconfiguration" 、
  "params": {
    " IDPConfigurationID " : " F983C602-12F9-4C67-B214-BF505185 CFED " 、
    "GenerateNewCertificate" : true
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "result": {
    "IDPConfigInfo" : {
      "enabled": true,

```

```
    "IDPConfigurationID" : " F983C602-12F9-4C67-B214-BF505185 CFED " 、  
    "IDPmetadata" : "<?XML version=\"1.0\" encoding=\"UTF-8 \"?>\r\n  
<entitydescriptor xmlns=\"URN:OASOASIS : 名称: tc:SAML : 2.0 : 元数据 \"\r\n  
xmlns : ds=\"http://www.w3.org/2000/09/xmllsig#\" \r\n  
xmlns : shibmd=\"urn:mace:shibboleth:metadatal:1.0\" \r\n  
xmlns : xml=\"http://www.w3.org/xml/1998/nameespace\" \r\n  
...</organization> \r\n  
</entitydescriptor> " 、  
    "IDPName" : "https://privider.name.url.com" 、  
    "ServiceProviderCertificate" : "--begin Certificate --\n  
Mi . . . bhi \n  
---- 结束证书 --\n" 、  
    "spmetadataurl" : "https://10.193.100100/auth/ui/Saml2"  
  }  
}
```

发布以下版本后新增

12.0

会话身份验证 API 方法

您可以使用基于会话的身份验证来管理用户会话。

ListAuthSessionsByClusterAdmin

可以使用ListAuthSessionsByClusterAdmin该方法列出与指定的 ClusterAdministaD 关联的所有验证会话。如果指定的 ClusterAdminID 映射到一组用户，则该组所有成员的所有验证会话都将列出。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
clusterAdminID	集群管理员的唯一标识符。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
sessions	验证会话的会话信息列表。	AuthSessionInfo

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "ListAuthSessionsByClusterAdmin" ,
  "ClusterAdminista" : 1.
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "sessions": [
    {
      "AccessGroupList" : [
        "administrator"
      ],
      "authmethod" : " 集群 " ,
      "ClusterAdministds" : [
        1
      ],
      "FinalTimeout" : "2020-03-14T19:21:24Z" ,
      "IDPConfigVersion" : 0 ,
      "lastAccessTimeout" : "2020-03-11T19 : 51:24z" ,
      "SessionCreationTime" : "2020-03-11T19 : 21 : 24Z" ,
      "SessionID" : "B12BFC64-F233-44DF-8B9F-6FB6C011ABF7" ,
      "username": "admin"
    }
  ]
}
```

发布以下版本后新增

12.0

ListAuthSessionSBYUserName

您可以使用ListAuthSessionsByUsername该方法列出指定用户的所有验证会话。不在AccessGroup ClusterAdmins/Administrator 权限中的调用方只能列出自己的会话。具有clusteradmins/administrator 权限的调用方可以列出属于任何用户的会话。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
authMethod	要列出的用户会话的身份验证方法。只有 ClusterAdmins/Administrator AccessGroup 中的调用方可以提供此参数。可能的值包括： <ul style="list-style-type: none"> • authMethod=cluster 指定 clusteradmin 用户名。 • AuthMethod=LDAP 指定用户的 LDAP DN 。 • AuthMethod=IdP 指定用户的 IDP UUID 或 NameID 。如果未将 IDP 配置为返回任一选项、则指定创建会话时发出的随机 UUID 。 	authMethod	无	是
username	用户的唯一标识符。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
sessions	验证会话的会话信息列表。	AuthSessionInfo

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "ListAuthSessionSBYUsername" 、
  "authmethod" : " 集群 " 、
  "username": "admin"
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "sessions": [
    {
      "AccessGroupList" : [
```

```
        "administrator"  
      ],  
      "authmethod" : " 集群 " 、  
      "ClusterAdministds" : [  
        1  
      ],  
      "FinalTimeout" : "2020-03-14T19:21:24Z" 、  
      "IDPConfigVersion" : 0 、  
      "lastAccessTimeout" : "2020-03-11T19 : 51:24z" 、  
      "SessionCreationTime" : "2020-03-11T19 : 21 : 24Z" 、  
      "SessionID" : "B12BFC64-F233-44DF-8B9F-6FB6C011ABF7" 、  
      "username": "admin"  
    }  
  ]  
}
```

发布以下版本后新增

12.0

节点 API 方法

您可以使用节点 API 方法来配置各个节点。这些方法适用于需要配置的单节点、已配置但尚未加入集群的单节点或正在积极加入集群的单节点。利用节点 API 方法，您可以查看并修改个别节点以及用于与该节点通信的集群网络的设置。您必须对单节点运行这些方法；您不能对集群地址运行每节点 API 方法。

CheckPingOnVlan

您可以在执行部署前网络验证时使用 CheckPingOnVlan 方法测试临时 VLAN 上的网络连接。CheckPingOnVlan 创建临时 VLAN 接口、使用 VLAN 接口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

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
attempts	指定系统应重复测试 ping 的次数。	整型	5	否
hosts	指定要执行 ping 操作的设备的地址或主机名的列表，以英文逗号分隔。	字符串	集群中的节点	否
interface	用于发送 ping 的现有（基本）接口。可能值： <ul style="list-style-type: none"> Bond10G：从绑定 10G 接口发送 ping。 Bond1G：从绑定 1G 接口发送 ping。 	字符串	无	是
packetSize	指定要在发送给每个 IP 的 ICMP 数据包中发送的字节数。该字节数必须小于在网络配置中指定的最大 MTU。	整型	无	否
pingTimeoutMsec	指定等待各 ping 响应的的时间（毫秒）。	整型	500 毫秒	否
prohibitFragmentation	启用 ICMP 数据包的不分段 (DF, Do not Fragment) 标志。	布尔型	false	否
sourceAddressV4	要在 ICMP ping 数据包中使用的源 IPv4 地址。	字符串	无	是
sourceAddressV6	要在 ICMP ping 数据包中使用的源 IPv6 地址。	字符串	无	是
totalTimeoutSec	指定在发出下一次 ping 尝试命令或结束此过程之前，ping 应等待系统响应的的时间（以秒为单位）。	整型	5	否
virtualNetworkTag	要在发送 ping 数据包时使用的 VLAN ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
details	可与此节点进行通信的各个 IP 的列表，以及 ping 响应统计信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

```
    "duration": "00:00:00.244379",  
    "result": "Passed"  
  }  
}
```

发布以下版本后新增

11.1

CheckProposedNodeAdditions

您可以使用 `CheckProposedNodeAdditions` 方法来测试一组存储节点，以查看是否可以在不出错或不违反最佳实践的情况下将其添加到存储集群中。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
nodes	可随时添加到存储集群中的存储节点的存储 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	说明	类型
proposedClusterValid	指示建议的存储节点是否会构成有效的存储集群。可能值： <ul style="list-style-type: none">• true• false	布尔型

名称	说明	类型
proposedClusterErrors	<p>使用建议的存储节点创建存储集群时可能发生的错误。可能的错误代码：</p> <ul style="list-style-type: none"> • nodesNoCapacity: 节点没有任何可用容量。 • nodesTooLarge: 对于当前活动保护方案，节点占用的集群容量份额太多。 • nodesConnectFailed: 无法连接到节点以查询硬件配置。 • nodesQueryFailed: 无法查询节点以获取硬件配置。 • nodesClusterMember: 节点的 IP 地址在集群中已被使用。 • nonFipsNodeCapable: 在启用 FIPS 140-2 驱动器加密功能后，无法将不支持 FIPS 的节点添加到存储集群中。 • nonFipsDrivesCapable: 在启用 FIPS 140-2 驱动器加密功能后，无法将具有不支持 FIPS 的驱动器的节点添加到集群中。 	字符串数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.0

CreateClusterSupportBundle

您可以在管理节点上使用 `CreateClusterSupportBundle` 从集群中的所有节点中收集支持包。创建支持包时，此支持包将以 `.tar.gz` 文件的形式存储在节点上。您只能在管理节点上运行此方法；在存储节点上运行将不起作用。

参数

注：您必须在管理节点上调用此方法。例如：

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

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>allowIncomplete</code>	如果无法从一个或多个节点收集支持包，则允许脚本继续运行。	布尔型	无	否
<code>bundleName</code>	所创建每个支持包的唯一名称。如果未提供任何名称，则使用“supportbundle”和节点名称作为文件名。	字符串	无	否
<code>extraArgs</code>	此参数将传递到 <code>sf_make_support_bundle</code> 脚本。只有在 NetApp 支持部门要求时，才应使用此参数。	字符串	无	否
<code>mvip</code>	集群的 MVIP。将从集群中的所有节点收集支持包。如果未指定 <code>nodes</code> 参数，则需要此参数。	字符串	无	是
<code>nodes</code>	要从中收集支持包的节点的 IP 地址。使用 <code>nodes</code> 或 <code>mvip</code> 指定要从中收集支持包的节点，但不能同时使用这两个参数。如果未指定 <code>mvip</code> ，则需要此参数。	字符串数组	无	是
<code>password</code>	集群管理员密码。 注： 输入密码时，将以文本格式显示此密码。	字符串	无	是
<code>username</code>	集群管理员用户名。	字符串	无	是

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

```
    },
    "id": 1
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id":1,
  "result":{
    "details":{
      "bundleName":"clusterbundle",
      "extraArgs":"",
      "files":[
        "/tmp/supportbundles/clusterbundle.cl-4SD5.tar"
      ],
      "output":"timeout -s KILL 1790s /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"
  }
}
```

发布以下版本后新增

9.6

CreateSupportBundle

您可以使用 `CreateSupportBundle` 在节点的目录下创建支持包文件。创建之后，该支持包将以 `tar.gz` 文件的形式存储在节点上。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
bundleName	支持包的唯一名称。如果未提供任何名称，则使用“supportbundle”和节点名称作为文件名。	字符串	无	否
extraArgs	此参数将传递到 <code>sf_make_support_bundle</code> 脚本。只有在 NetApp 支持部门要求时，才应使用此参数。	字符串	无	否
timeoutSec	支持包脚本运行的秒数。	整型	1500	否

返回值

此方法具有以下返回值：

名称	说明	类型
details	支持包的详细信息。可能值： <ul style="list-style-type: none"> bundleName: 在 API 方法 CreateSupportBundle 中指定的名称。如果未指定任何名称，则使用“supportbundle”。 extraArgs: 通过此方法传递的参数。 files: 系统创建的支持包文件的列表。 output: 创建支持包的脚本的命令行输出。 timeoutSec: 支持包脚本在停止前运行的秒数。 url: 指向所创建的支持包的 URL。 	JSON 对象
duration	创建支持包所使用的时间，格式为 HH:MM:SS.ssssss。	字符串
result	支持包操作的成功或失败状态。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateSupportBundle",
  "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": {
      "bundleName": "supportbundle",
      "extraArgs": "",
      "files": [
        "supportbundle.SF-D220.tar.gz"
      ],
      "output": "timeout -s KILL 1500s /sf/scripts/sf_make_support_bundle --quiet /tmp/solidfire-dtemp.bYRag7/supportbundle timeout -s KILL 1500s pigz --verbose --fast \" /tmp/solidfire-dtemp.bYRag7/supportbundle.SF-D220.tar to /tmp/solidfire-dtemp.bYRag7/supportbundle.SF-D220.tar.gz Moved '/tmp/solidfire-dtemp.bYRag7/supportbundle.SF-D220.tar.gz' to /tmp/supportbundles",
      "timeoutSec": 1500,
      "url": [
        "https://192.168.130.155:442/config/supportbundles/supportbundle.SF-D220.tar.gz"
      ]
    },
    "duration": "00:00:21.166692",
    "result": "Passed"
  }
}
```

发布以下版本后新增

9.6

DeleteAllSupportBundles

您可以使用 `DeleteAllSupportBundles` 方法删除通过 `CreateSupportBundle` API 方法生成的所有支持包。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

DisableSsh

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
enabled	此节点的 SSH 服务状态。	布尔型

请求示例

此方法的请求类似于以下示例：

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

```
"id" : 1  
}
```

响应示例

此方法返回类似于以下示例的响应:

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

EnableSsh

您可以使用 `EnableSsh` 方法为单个节点启用 SSH 服务。此方法不会影响集群范围的 SSH 超时持续时间,也不会影响通过全局 SSH 超时为此节点禁用 SSH。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值:

名称	说明	类型
enabled	此节点的 SSH 服务状态。	布尔型

请求示例

此方法的请求类似于以下示例:

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

响应示例

此方法返回类似于以下示例的响应:

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

GetClusterConfig

您可以使用 `GetClusterConfig` API 方法返回有关节点与其集群通信所使用的集群配置的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值:

名称	说明	类型
集群	节点用于与集群通信的集群配置信息。	<i>cluster</i>

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

GetClusterState

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
集群	集群名称。	字符串

名称	说明	类型
state	<ul style="list-style-type: none"> • Available: 节点尚未配置集群名称。 • Pending: 节点正在等待添加到已命名的特定集群，可以进行添加。 • Active: 节点是集群的活动成员，不能添加到其他集群。 	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

GetConfig

您可以使用 `GetConfig` API 方法获取节点的所有配置信息。此 API 方法包含可通过 `GetClusterConfig` 和 `GetNetworkConfig` API 方法获得的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
config	集群的配置详细信息。此对象包含： <ul style="list-style-type: none"> • cluster: 集群信息，用于确定存储节点与其关联存储集群通信的方式。 • network (所有接口): 节点的每个网络接口的网络连接类型和当前设置。 	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetConfig",
```

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

响应示例

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

相关参考

[GetClusterConfig](#) on page 203

您可以使用 `GetClusterConfig` API 方法返回有关节点与其集群通信所使用的集群配置的信息。

[GetNetworkConfig](#) on page 217

您可以使用 `GetNetworkConfig` 方法显示节点的网络配置信息。

[GetConfig](#) on page 497

`GetConfig` 方法返回类似于以下示例的响应。鉴于篇幅限制，该响应仅包含集群中一个节点的信息。

GetDriveConfig

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
driveConfig	有关节点所连接的驱动器的信息。	<i>drive</i>

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应类似于以下示例。鉴于篇幅限制，此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  
}  
}
```

GetHardwareConfig

您可以使用 GetHardwareConfig 方法获取节点的硬件配置信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
hardwareConfig	硬件信息和当前设置列表。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应类似于以下示例。

```
{
  "id": 1,
  "result": {
    "hardwareConfig": {
      "biosRevision": "1.0",
      "biosVendor": [
        "NetApp",
        "SolidFire"
      ],
      "biosVersion": "1.1.2",
      "blockDriveSizeBytes": 300069052416,
      "blockDrives": [
        "/dev/slot0",
        "/dev/slot1",
        "/dev/slot2",
        "/dev/slot3",
        "/dev/slot4",
        "/dev/slot5",
        "/dev/slot6",
        "/dev/slot7",
        "/dev/slot8",
        "/dev/slot9"
      ],
      "blockServiceFormat": "Standard",
      "bmcFirmwareRevision": "1.6",
      "bmcIpmiVersion": "2.0",
      "chassisType": "R620",
      "cpuCores": 6,
      "cpuCoresEnabled": 6,
      "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
      "cpuThreads": 12,
      "driveSizeBytesInternal": 400088457216,
      "fibreChannelFirmwareRevision": "",
      "fibreChannelModel": "",
      "fibreChannelPorts": {},
      "idracVersion": "1.06.06",
      "ignoreFirmware": [],
      "memoryGB": 72,
      "memoryMhz": 1333,
      "networkDriver": [
        "bnx2x"
      ],
      "nicPortMap": {
        "PortA": "eth2",
        "PortB": "eth3",
        "PortC": "eth0",
        "PortD": "eth1"
      },
      "nodeType": "SF3010",
      "numCpu": 2,
      "numDrives": 10,
      "numDrivesInternal": 1,
      "nvramTempMonitorEnable": false,
      "rootDrive": "/dev/sdimm0",
      "scsiBusExternalDriver": "mpt3sas",
      "scsiBusInternalDriver": "ahci",
    }
  }
}
```

```

        "sliceDriveSizeBytes": 299988156416,
        "sliceDrives": [
            "/dev/sdimm0p4"
        ],
        "slotOffset": 0,
        "solidfireDefaults": {
            "bufferCacheGB": 12,
            "configuredIops": 50000,
            "cpuDmaLatency": -1,
            "driveWriteThroughputMBPerSleep": 10,
            "maxDriveWriteThroughputMBPerSec": 175,
            "maxIncomingSliceSyncs": 10,
            "postCallbackThreadCount": 8,
            "sCacheFileCapacity": 100000000,
            "sliceFileLogFileCapacity": 5000000000
        }
    }
}
}
}

```

发布以下版本后新增

9.6

GetHardwareInfo

您可以使用 `GetHardwareInfo` 方法获取单个节点的硬件信息和状态。硬件信息通常包含制造商、供应商、版本、驱动器和其他相关的标识信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
force	将此“force”参数设置为 true，可在集群中的所有节点上运行。	布尔型	false	否

返回值

此方法具有以下返回值：

名称	说明	类型
hardwareInfo	节点的硬件信息。	hardwareInfo

请求示例

此方法的请求类似于以下示例：

```

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

```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "hardwareInfo": {
      "bus": {
        "core_DMI:0200": {
          "description": "Motherboard",
          "physid": "0",
          "product": "0A47AA",
          "serial": "..AB123456C12354.",
          "version": "C07"
        }
      }
    },
    "driveHardware": [
      {
        "canonicalName": "sdh",
        "connected": true,
        "dev": 2160,
        "devPath": "/dev/disk/by-path/pci-0000:41:00.0-sas-0x500056b37789abf0-lun-0",
        "driveEncryptionCapability": "fips",
        "driveType": "Block",
        "lifeRemainingPercent": 92,
        "lifetimeReadBytes": 175436696911872,
        "lifetimeWriteBytes": 81941097349120,
        "name": "scsi-SATA_INTEL_SSDSC2BB3BTWL12345686300AAA",
        "path": "/dev/sdh",
        "pathLink": "/dev/disk/by-path/pci-0000:41:00.0-sas-0x500056b37789abf0-lun-0",
        "powerOnHours": 17246,
        "product": "INTEL SSDAA2AA300A4",
        "reallocatedSectors": 0,
        "reserveCapacityPercent": 100,
        "scsiCompatId": "scsi-SATA_INTEL_SSDSC2BB3BTWL12345686300AAA",
        "scsiState": "Running",
        "securityAtMaximum": false,
        "securityEnabled": false,
        "securityFrozen": false,
        "securityLocked": false,
        "securitySupported": true,
        "serial": "AAAA33710886300AAA",
        "size": 300069052416,
        "slot": 1,
        "smartSsdWriteCapable": false,
        "uuid": "aeal78b9-c336-6bab-a61d-87b615e8120c",
        "vendor": "Intel",
        "version": "D2010370"
      },
      ...
    ]
  }
}
```

发布以下版本后新增

9.6

GetIpmiConfig

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

参数

此方法具有以下输入参数：

名称	说明	类型
chassisType	用于显示每个节点机箱类型的信息。可能值： <ul style="list-style-type: none">all: 返回每种机箱类型的传感器信息。{chassis type}: 返回指定机箱类型的传感器信息。	字符串

返回值

此方法具有以下返回值：

名称	说明	类型
sensorName	已找到的传感器的名称。	字符串
uniqueSensorID	传感器的唯一标识符。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "nodes": [
      {
        "nodeID": 1,
        "result": {
          "ipmiConfig": {
            "C220M4": [
              {
                "sensorName": "Fan1A RPM",
                "uniqueSensorID": "29.1:0xf"
              },
              {
                "sensorName": "Fan1B RPM",
                "uniqueSensorID": "29.1:0x10"
              },
              {
                "sensorName": "Fan2A RPM",
                "uniqueSensorID": "29.2:0x11"
              }
            ]
          }
        }
      }
    ]
  }
}
```

```
    },  
    {  
      "sensorName": "Fan2B RPM",  
      "uniqueSensorID": "29.2:0x12"  
    },  
    {  
      "sensorName": "Fan3A RPM",  
      "uniqueSensorID": "29.3:0x13"  
    },  
    {  
      "sensorName": "Fan3B RPM",  
      "uniqueSensorID": "29.3:0x14"  
    },  
    {  
      "sensorName": "Fan4A RPM",  
      "uniqueSensorID": "29.4:0x15"  
    },  
    {  
      "sensorName": "Fan4B RPM",  
      "uniqueSensorID": "29.4:0x16"  
    },  
    {  
      "sensorName": "Fan5A RPM",  
      "uniqueSensorID": "29.5:0x17"  
    },  
    {  
      "sensorName": "Fan5B RPM",  
      "uniqueSensorID": "29.5:0x18"  
    },  
    {  
      "sensorName": "Fan6A RPM",  
      "uniqueSensorID": "29.6:0x19"  
    },  
    {  
      "sensorName": "Fan6B RPM",  
      "uniqueSensorID": "29.6:0x1a"  
    },  
    {  
      "sensorName": "Exhaust Temp",  
      "uniqueSensorID": "7.1:0x1"  
    },  
    {  
      "sensorName": "Inlet Temp",  
      "uniqueSensorID": "7.1:0x4"  
    },  
    {  
      "sensorName": "PS1",  
      "uniqueSensorID": "10.1:0x26"  
    },  
    {  
      "sensorName": "PS2",  
      "uniqueSensorID": "10.2:0x2c"  
    }  
  ],  
  "R620": [  
    {  
      "sensorName": "Fan1A RPM",  
      "uniqueSensorID": "7.1:0x30"  
    },  
    {
```

```
"sensorName": "Fan1B RPM",  
"uniqueSensorID": "7.1:0x31"  
},  
{  
  "sensorName": "Fan2A RPM",  
  "uniqueSensorID": "7.1:0x32"  
},  
{  
  "sensorName": "Fan2B RPM",  
  "uniqueSensorID": "7.1:0x33"  
},  
{  
  "sensorName": "Fan3A RPM",  
  "uniqueSensorID": "7.1:0x34"  
},  
{  
  "sensorName": "Fan3B RPM",  
  "uniqueSensorID": "7.1:0x35"  
},  
{  
  "sensorName": "Fan4A RPM",  
  "uniqueSensorID": "7.1:0x36"  
},  
{  
  "sensorName": "Fan4B RPM",  
  "uniqueSensorID": "7.1:0x37"  
},  
{  
  "sensorName": "Fan5A RPM",  
  "uniqueSensorID": "7.1:0x38"  
},  
{  
  "sensorName": "Fan5B RPM",  
  "uniqueSensorID": "7.1:0x39"  
},  
{  
  "sensorName": "Fan6A RPM",  
  "uniqueSensorID": "7.1:0x3a"  
},  
{  
  "sensorName": "Fan6B RPM",  
  "uniqueSensorID": "7.1:0x3b"  
},  
{  
  "sensorName": "Fan7A RPM",  
  "uniqueSensorID": "7.1:0x3c"  
},  
{  
  "sensorName": "Fan7B RPM",  
  "uniqueSensorID": "7.1:0x3d"  
},  
{  
  "sensorName": "Exhaust Temp",  
  "uniqueSensorID": "7.1:0x1"  
},  
{  
  "sensorName": "Inlet Temp",  
  "uniqueSensorID": "7.1:0x4"  
},  
{
```

```
        "sensorName": "PS1",  
        "uniqueSensorID": "10.1:0x62"  
      },  
      {  
        "sensorName": "PS2",  
        "uniqueSensorID": "10.2:0x63"  
      }  
    ],  
  }  
}
```

发布以下版本后新增

9.6

GetIpmiInfo

您可以使用 `GetIpmiInfo` 方法报告系统所监控的节点风扇、进气和排气温度及电源的详细传感器（对象）信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
sensors	来自节点内每个传感器的详细信息。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此 API 方法返回的响应较长，因此本文档特意删除了部分响应内容。剩余部分为系统为确保节点运行性能最佳而监控的部分硬件信息。

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

NODE

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

```
    }  
  }  
}
```

发布以下版本后新增

9.6

GetNetworkConfig

您可以使用 `GetNetworkConfig` 方法显示节点的网络配置信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
network	节点的每个网络接口的网络连接类型和当前设置。	<i>network</i> (所有接口)

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

相关参考

[GetNetworkConfig](#) on page 532

`GetNetworkConfig` 方法返回类似于以下示例的响应。

GetNetworkInterface

您可以使用 `GetNetworkInterface` 方法获取有关某个节点上的网络接口的信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
interface	每个节点上要获取其相关信息的接口名称。可能值： <ul style="list-style-type: none"> Bond1G Bond10G 	字符串	无	否
force	将此参数设置为 true，可在集群中的所有节点上运行。	布尔型	false	否

返回值

此方法具有以下返回值：

名称	说明	类型
nodes	一个对象数组，用于说明存储集群中每个存储节点上的接口。此数组中的每个对象都包含以下项： <ul style="list-style-type: none"> nodeID: (整型) 存储集群中使用此接口信息的存储节点的 ID。 result: (<i>networkInterface</i>) 此存储节点的接口配置信息。 	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "nodes": [
      {
        "nodeID": 1,
        "result": {
          "interface": {
            "address": "10.117.64.32",
            "addressV6": ":::",
            "broadcast": "10.117.79.255",
            "macAddress": "90:b1:1c:42:e0:1e",
            "mtu": 1500,
            "name": "Bond1G",
            "namespace": false,
            "netmask": "255.255.240.0",
            "status": "UpAndRunning",
            "type": "BondMaster",
            "virtualNetworkTag": 0
          }
        }
      }
    ]
  }
}
```

```
    }  
  },  
{  
  "nodeID": 2,  
  "result": {  
    "interface": {  
      "address": "10.117.64.35",  
      "addressV6": "::",  
      "broadcast": "10.117.79.255",  
      "macAddress": "d4:ae:52:7a:ae:23",  
      "mtu": 1500,  
      "name": "Bond1G",  
      "namespace": false,  
      "netmask": "255.255.240.0",  
      "status": "UpAndRunning",  
      "type": "BondMaster",  
      "virtualNetworkTag": 0  
    }  
  }  
},  
{  
  "nodeID": 3,  
  "result": {  
    "interface": {  
      "address": "10.117.64.39",  
      "addressV6": "::",  
      "broadcast": "10.117.79.255",  
      "macAddress": "c8:1f:66:f0:9d:17",  
      "mtu": 1500,  
      "name": "Bond1G",  
      "namespace": false,  
      "netmask": "255.255.240.0",  
      "status": "UpAndRunning",  
      "type": "BondMaster",  
      "virtualNetworkTag": 0  
    }  
  }  
},  
{  
  "nodeID": 4,  
  "result": {  
    "interface": {  
      "address": "10.117.64.107",  
      "addressV6": "::",  
      "broadcast": "10.117.79.255",  
      "macAddress": "b8:ca:3a:f5:24:f8",  
      "mtu": 1500,  
      "name": "Bond1G",  
      "namespace": false,  
      "netmask": "255.255.240.0",  
      "status": "UpAndRunning",  
      "type": "BondMaster",  
      "virtualNetworkTag": 0  
    }  
  }  
}  
]  
}
```

发布以下版本后新增

9.6

GetNodeActiveTlsCiphers

您可以在某个节点上使用 `GetNodeActiveTlsCiphers` 方法来获取此节点当前已接受的 TLS 密码列表。可以在管理节点和存储节点上使用此方法。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>mandatoryCiphers</code>	此节点上的必需 TLS 密码套件列表。这些密码在此节点上始终处于活动状态。	字符串
<code>supplementalCiphers</code>	此节点上的补充 TLS 密码套件列表。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

GetNodeFipsDrivesReport

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
fipsDrives	一个 JSON 对象，包含此节点的 FIPS 140-2 功能支持状态。可能值： <ul style="list-style-type: none">• None：节点不支持 FIPS。• Partial：节点支持 FIPS，但此节点中并非所有驱动器都为 FIPS 驱动器。• Ready：节点支持 FIPS，并且此节点中的所有驱动器均为 FIPS 驱动器（或者不存在任何驱动器）。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.5

GetNodeSSLCertificate

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

参数

注：您必须在管理节点上调用此方法。例如：

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



```
C1473ACD5C1402C1402C1402C1402C1402C1402C1473ACD5C1402C1402C1473AFC5AFC5A1402C1402C1473A140
2C1402C1402C29CF5C29CF5C1473AFC
  "notAfter": "2027-03-06T22:50:26Z",
  "notBefore": "2017-03-08T22:50:26Z",
  "serial": "CC1B221598E37FF3",
  "sha1Fingerprint": "1D:70:7A:6F:18:8A:CD:29:50:C7:95:B1:DD:5E:63:21:F4:FA:6E:
21",
    "subject": "/C=US/ST=NV/L=Denver/O=NetApp/emailAddress=test@netapptest.org"
  }
}
}
```

GetNodeSupportedTlsCiphers

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>mandatoryCiphers</code>	此节点上的必需 TLS 密码套件列表。这些密码在此节点上始终处于活动状态。	字符串
<code>defaultSupplementalCiphers</code>	此节点上的默认补充 TLS 密码套件列表。运行 <code>ResetNodeSupplementalTlsCiphers</code> API 方法时，这些补充密码将还原为此列表。	字符串
<code>supportedSupplementalCiphers</code>	可用的补充 TLS 密码套件列表，您可以使用 <code>SetNodeSupplementalTlsCiphers</code> API 方法进行配置。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id" : 1,
  "result" : {
    "defaultSupplementalCiphers": [
      "DHE-RSA-AES128-SHA256",
      "DHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES128-SHA256",
      "ECDHE-RSA-AES128-GCM-SHA256"
    ],

```

```
    "mandatoryCiphers": [
      "DHE-RSA-AES256-SHA256",
      "DHE-RSA-AES256-GCM-SHA384",
      "ECDHE-RSA-AES256-SHA384",
      "ECDHE-RSA-AES256-GCM-SHA384"
    ],
    "supportedSupplementalCiphers": [
      "DHE-RSA-AES128-SHA256",
      "DHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES128-SHA256",
      "ECDHE-RSA-AES128-GCM-SHA256",
      "DHE-RSA-AES256-SHA",
      "ECDHE-RSA-AES256-SHA",
      "DHE-RSA-CAMELLIA256-SHA",
      "DHE-RSA-AES128-SHA",
      "ECDHE-RSA-AES128-SHA",
      "DHE-RSA-CAMELLIA128-SHA"
    ]
  }
}
```

GetPendingOperation

您可以使用 `GetPendingOperation` 方法检测节点上当前正在执行的操作。当操作完成时，此方法还可以用于进行回报。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
pending	可能值： <ul style="list-style-type: none">• true: 操作仍在进行。• false: 操作不再进行。	布尔型
operation	正在进行或已完成的操作名称。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

```
        "pendingOperation" : {  
            "pending" : "true",  
            "operation" : "TestDrivesInternal",  
        }  
    }  
}
```

发布以下版本后新增

9.6

GetSshInfo

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
result	此节点的 SSH 服务状态。	布尔型

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

ListNetworkInterfaces

您可以使用 `ListNetworkInterfaces` 方法列出有关节点上每个网络接口的信息。此 API 方法适用于单个节点；要访问单个节点，需要进行用户 ID 和密码身份验证。但是，如果参数 `force` 在此方法调用中的值为 `true`，则可以对集群使用此方法。如果对集群使用此参数，则会列出所有接口。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
force	可能值： <ul style="list-style-type: none"> • true: 返回有关集群中所有网络接口的信息。 • false: 不返回信息。 	布尔型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
interfaces	存储节点（如果 force=true，则为整个存储集群）的每个网络接口的配置信息列表。	<i>networkInterface</i> 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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


返回值

此方法具有以下返回值：

名称	说明	类型
driveHardware	返回的节点驱动器硬件信息。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[EnableEncryptionAtRest](#) on page 278

您可以使用 `EnableEncryptionAtRest` 方法在集群上启用高级加密标准 (Advanced Encryption Standard, AES) 256 位空闲加密，以便集群可以管理用于每个节点上的驱动器的加密密钥。默认情况下不会启用此功能。

ListTests

您可以使用 `ListTests` 方法列出可对节点运行的测试。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
tests	列出可对节点执行的测试。	字符串数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

ListUtilities

您可以使用 `ListUtilities` 方法列出可对节点运行的操作。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
utilities	列出当前可对节点运行的实用程序。	字符串数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

RemoveNodeSSLCertificate

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

参数

注：您必须在管理节点上调用此方法。例如：

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

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

ResetDrives

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

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
drives	列出要重置的设备名称（非驱动器 ID）。	字符串	无	是
force	设置为 true，可重置驱动器。	布尔型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
details	正在重置的驱动器的详细信息。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

ResetNode

您可以使用 `ResetNode` 方法将节点重置为出厂设置。调用此方法时，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。但此操作会保留节点的网络设置。加入集群的节点不能重置为出厂设置。

参数

`ResetNode` API 仅适用于处于“Available”状态的节点。而不适用于集群中的“Active”节点或处于“Pending”状态的节点。



警告：此方法将清除节点上的所有客户数据。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
build	用于指定要将节点重置到的远程 Element 软件映像的 URL。	URL	无	否
force	设置为 true，可重置节点。	布尔型	无	是
options	用于输入运行重置操作的规范。如果需要，NetApp 支持部门将提供详细信息。	JSON 对象	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": null,
  "result": {
    "rtfiInfo": {
      "build": "file:///sf/rtfi/image/filesystem.squashfs",
      "generation": "9",
      "options": {
        "edebug": "",
        "sf_auto": "0",
        "sf_bond_mode": "ActivePassive",
        "sf_check_hardware": "0",
        "sf_disable_otpw": "0",
        "sf_fa_host": "",
        "sf_hostname": "SF-FA18",
        "sf_inplace": "1",
        "sf_inplace_die_action": "kexec",
        "sf_inplace_safe": "0",
        "sf_keep_cluster_config": "0",
        "sf_keep_data": "0",
        "sf_keep_hostname": "0",
        "sf_keep_network_config": "0",
        "sf_keep_paths": "\\ /var/log/hardware.xml\\",
        "sf_max_archives": "5",
        "sf_nvram_size": "",
        "sf_oldroot": "",
        "sf_postinst_erase_root_drive": "0",
        "sf_root_drive": "",
        "sf_rtfi_cleanup_state": "",
        "sf_secure_erase": "1",
        "sf_secure_erase_retries": "5",
        "sf_slice_size": "",
        "sf_ssh_key": "1",
        "sf_ssh_root": "1",
        "sf_start_rtfti": "1",
        "sf_status_httpserver": "1",
        "sf_status_httpserver_stop_delay": "5m",
        "sf_status_inject_failure": "",
        "sf_status_json": "0",
        "sf_support_host": "sfsupport.solidfire.com",
        "sf_test_hardware": "0",
        "sf_upgrade": "0",
        "sf_upgrade_firmware": "0",
        "sf_upload_logs_url": ""
      }
    }
  }
}
```

```
    },  
    "statusUrlAll": "http://192.168.130.20/status/all.json",  
    "statusUrlCurrent": "http://192.168.130.20/status/current.json"  
  }  
}
```

发布以下版本后新增

9.6

ResetNodeSupplementalTlsCiphers

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

参数

注: 您必须在管理节点上调用此方法。例如:

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

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例:

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

响应示例

此方法返回类似于以下示例的响应:

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

RestartNetworking

您可以使用 `RestartNetworking` 方法在节点上重新启动网络服务。

参数



警告: 此方法将重新启动节点上的所有网络服务, 导致网络连接暂时丢失。

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
force	设置为 true, 可在节点上重新启动网络服务。	布尔型	无	是

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

RestartServices

您可以使用 `RestartServices` 方法在节点上重新启动服务。

参数



警告：此方法会导致节点服务暂时中断。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
force	设置为 true，可在节点上重新启动服务。	布尔型	无	是
service	要重新启动的服务名称。	字符串	无	否
action	要对服务执行的操作（启动、停止、重新启动）。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
details	服务重新启动过程的输出，包括错误（如有）。	JSON 对象
duration	重新启动节点服务所需的时间（以秒为单位）。	字符串

名称	说明	类型
result	重新启动的结果。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

SetClusterConfig

您可以使用 SetClusterConfig 方法设置节点与其关联集群进行通信所使用的配置。要显示节点的当前集群接口设置，请运行 GetClusterConfig API 方法。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
cluster	在此方法调用期间应更改的配置属性。只需要将要更改的字段通过此参数以成员形式添加到此方法中。	<i>cluster</i>	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
集群	节点与集群通信所使用的配置信息。	<i>cluster</i>

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

SetConfig

您可以使用 `SetConfig` 方法设置节点的网络和集群信息。此方法可在一个 API 方法中提供 `SetClusterConfig` 和 `SetNetworkConfig` 这两种方法中提供的相同设置。您只需在此方法中提供要更改的字段即可。

参数



警告：更改节点上的 `bond-mode` 会导致暂时断开网络连接。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
cluster	集群信息，用于确定存储节点与其关联存储集群通信的方式。	<i>cluster</i>	无	否
network	节点的每个网络接口的网络连接类型和当前设置。	<i>network (所有接口)</i>	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
config	节点的新配置和当前配置。此对象包含： <ul style="list-style-type: none"> <i>cluster</i>：集群信息，用于确定存储节点与其关联存储集群通信的方式。 <i>network (所有接口)</i>：节点的每个网络接口的网络连接类型和当前设置。 	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应与 GetConfig 方法的返回值相同。使用 SetConfig 时，系统会显示对象的所有字段以及更新值。

发布以下版本后新增

9.6

相关参考

[SetClusterConfig](#) on page 236

您可以使用 SetClusterConfig 方法设置节点与其关联集群进行通信所使用的配置。要显示节点的当前集群接口设置，请运行 GetClusterConfig API 方法。

[SetNetworkConfig](#) on page 239

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

[GetConfig](#) on page 497

GetConfig 方法返回类似于以下示例的响应。鉴于篇幅限制，该响应仅包含集群中一个节点的信息。

SetNetworkConfig

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

参数



警告: 更改节点上的 bond-mode 会导致暂时断开网络连接。

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
network	包含要修改的节点网络设置的对象。只需要将要更改的字段通过此参数以属性形式添加到此方法中。	<i>network</i> (所有接口)	无	否

返回值

此方法具有以下返回值:

名称	说明	类型
network	节点的新网络配置和当前网络配置。	<i>network</i> (所有接口)

请求示例

此方法的请求类似于以下示例:

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

响应示例

此方法的响应与 GetNetworkConfig 方法的响应相同。此方法会显示每个对象的所有成员，并显示任何已更改成员的新值。

发布以下版本后新增

9.6

相关参考

[GetNetworkConfig](#) on page 217

您可以使用 GetNetworkConfig 方法显示节点的网络配置信息。

[GetNetworkConfig](#) on page 532

GetNetworkConfig 方法返回类似于以下示例的响应。

SetNodeSSLCertificate

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

参数

注: 您必须在管理节点上调用此方法。例如:

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

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
certificate	PEM 编码的证书文本版本。	字符串	无	是
privateKey	PEM 编码的专用密钥文本。	字符串	无	是

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例:

```
{
  "method" : "SetNodeSSLCertificate",
  "params" : {
    "privateKey": "-----BEGIN RSA PRIVATE KEY-----\nMIIEowIBAAKCAQEASU
+28fnLKQNWEMR6akeDKuehSpS79odLGigI18qlCV/AUY5\nzLjqsTjBvTJVRv44yoCTgNrx36U7FHP4t6P/
Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP20HRYpZvY
+Uz7LYEFCmrgpGZQF3iOSIcBHtLKE5186JVT6j5dg\n6yjugQ0352y1c9HXHcn61b/
jy10DmVNUZ0caQwAmIS3Jmoyx+zj/Ya4WKq+2SqTA\nX7bX0F3wHHfXnZ1HnM8fET5N/9A
+K61S7dg9cyXu4afXcgKy14JiNBvqbbjhgJtE\n76yAy6rTHu0xM3jjdkcb9Y8miNzxFAcQ
+itawIDAQABoIBAH1j1IZr6/sltqVW\n00qVC/49dyNu
+KwVSq92ti9rFe7hBPueh9gklh78hP9QlitLkir3YK4GFsTFUMux\n7z1NRCxA/
4LrmLSkAjw2kRXdFv12bwZq0ua9NefGw9208D2OZvbuOxk7Put2p6se
\nfgNzSjf2SI5DIX3UMe5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7wnq\nt+Am5/
Vu1LI6rGiG6oHEW0oGSuHllesIyXXa2hqU+1+iF2iGRMTiXac4C8d1lNU\nWGIRCFJAmsAQ+hQm7pmtSKdEqumj/
PIoGXf0BoFVEWAiJIMEgnfuLZp8IelJQXn\nSFJbk2ECgYEA
+d5ooU4thZXylWHUzqomaxyzOruA1T53UeH69HiFTrLjvfwuaiqj\nlHzPlhms6hxexwzldzApgog/NOM
+2bAc0rn0dqvtV4doejtLDZKRqrNcf/cuN2QX
\njaCJC1CWau3sEHcckLOhWeY4HaPSoWq0GKLmKkKDChB4nWUYg3gSWQkCgYEA9zuN\nHW8GPS+yjixeKXmkK00x/
vVxzR+J5HH5znaIHss48THyhZxPLr+v30Hy2h0yAlBS\nny5Ja6wsomb0mVe4NxVtVawg2E9vVvTa1UC
+TNmFBBuLRPfjcnjDerrSuQ51YY+M\nC9MJtXGfhp//
G0bZwsRzXz0BsUJb15tppaZIs9MCGyAJricpkKjM0x1Z1jdvXsos
\nPilnbho4qLngrzuUuxKXEPenzBxUOqCpwQgdZLYYw788TCVVIVXLEYem2s07dDA\nnDT0
```


名称	说明	类型	默认值	必需
supplementalCiphers	使用 OpenSSL 命名方案的补充密码套件名称。使用密码套件名称时不区分大小写。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
mandatoryCiphers	此节点上的必需 TLS 密码套件列表。这些密码在此节点上始终处于活动状态。	字符串
supplementalCiphers	此节点上的补充 TLS 密码套件列表。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

Shutdown

您可以使用 `Shutdown` 方法重新启动或关闭尚未添加到集群中的节点。要使用此方法，请登录到待定节点的 MIP，并输入 `shutdown` 方法以及下表中的 `restart` 或 `halt` 选项。

参数



警告: 此方法将重新启动节点上的所有网络服务，导致网络连接暂时丢失。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
option	为关闭节点而采取的操作： <ul style="list-style-type: none">• <code>restart</code>：重新启动节点。• <code>halt</code>：完全关闭节点。	字符串	无	是

注: 您需要使用端口 443 访问单个节点上的 API 服务。请参见以下 HTTPS API 请求示例，将 `<MIP>` 替换为单个节点的管理 IP 地址，并将 `<API version>` 替换为此节点上运行的 Element 软件版本：

```
https://<MIP>:443/json-rpc/<API version>/?method=Shutdown&option=halt
```

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": "Shutdown",  
  "params": {  
    "option": "restart"  
  },  
  "id": 1  
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

TestConnectEnsemble

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

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
ensemble	用于连接测试的集合节点集群 IP 地址的列表，以英文逗号分隔。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
details	返回的对象： <ul style="list-style-type: none"><code>nodes</code>：（对象）列出测试中的每个集合节点以及测试结果。<code>duration</code>：（字符串）运行测试所需的时间。<code>result</code>：（字符串）整体测试结果。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

TestConnectMvip

您可以使用 `TestConnectMvip` 方法测试与存储集群的管理连接。该测试会对 `MVIP` 执行 `ping` 操作，并执行简单的 API 方法来验证连接。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>mvip</code>	您可以传递此值以测试其他 <code>MVIP</code> 的管理连接。测试目标集群的连接时，您无需使用该值。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>details</code>	有关测试操作的信息（JSON 对象）： <ul style="list-style-type: none"><code>connected</code>: 指示此测试是否可以连接到 <code>MVIP</code>（布尔值）<code>mvip</code>: 用于测试的 <code>MVIP</code>（字符串）<code>pingBytes</code>: 56 字节和 1500 字节 <code>ping</code> 测试的详细信息（对象）<ul style="list-style-type: none"><code>56</code>: 56 字节 <code>ping</code> 测试的结果（JSON 对象）：<ul style="list-style-type: none"><code>individualResponseTimes</code>: 列出每个集合节点的响应时间（字符串数组）<code>individualStatus</code>: 列出每个集合节点的 <code>ping</code> 状态（布尔值数组）<code>responseTime</code>: 平均 <code>ping</code> 响应时间（字符串）<code>successful</code>: 指示 <code>ping</code> 测试是否成功（布尔值）<code>1500</code>: 1500 字节 <code>ping</code> 测试的结果（JSON 对象）：<ul style="list-style-type: none"><code>individualResponseTimes</code>: 列出每个集合节点的响应时间（字符串数组）<code>individualStatus</code>: 列出每个集合节点的 <code>ping</code> 状态（布尔值数组）<code>responseTime</code>: 平均 <code>ping</code> 响应时间（字符串）<code>successful</code>: <code>ping</code> 测试是否成功（布尔值）<code>duration</code>: 运行测试所需的时间长度（字符串）<code>result</code>: 整体测试结果（字符串）	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

```
    "id":1  
  }
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id": 1,  
  "result": {  
    "details": {  
      "connected": true,  
      "mvip": "172.27.62.50",  
      "pingBytes": {  
        "1500": {  
          "individualResponseTimes": [  
            "00:00:00.000250",  
            "00:00:00.000206",  
            "00:00:00.000200",  
            "00:00:00.000199",  
            "00:00:00.000199"  
          ],  
          "individualStatus": [  
            true,  
            true,  
            true,  
            true,  
            true  
          ],  
          "responseTime": "00:00:00.000211",  
          "successful": true  
        },  
        "56": {  
          "individualResponseTimes": [  
            "00:00:00.000217",  
            "00:00:00.000122",  
            "00:00:00.000117",  
            "00:00:00.000119",  
            "00:00:00.000121"  
          ],  
          "individualStatus": [  
            true,  
            true,  
            true,  
            true,  
            true  
          ],  
          "responseTime": "00:00:00.000139",  
          "successful": true  
        }  
      }  
    },  
    "duration": "00:00:00.271244",  
    "result": "Passed"  
  }  
}
```

发布以下版本后新增

9.6

TestConnectSvip

您可以使用 `TestConnectSvip` 方法测试与存储集群的存储连接。此测试将使用 ICMP 数据包对 SVIP 执行 ping 操作，如果测试成功，它将作为 iSCSI 启动程序而连接。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
svip	您可以传递此值以测试其他 SVIP 的管理连接。测试目标集群的连接时，您无需使用该值。	字符串	无	否

返回值

此方法具有以下返回值：

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

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "details": {
      "connected": true,
      "pingBytes": {
        "56": {
          "individualResponseTimes": [
            "00:00:00.000152",
            "00:00:00.000132",
            "00:00:00.000119",
            "00:00:00.000114",
            "00:00:00.000112"
          ],
          "individualStatus": [
            true,
            true,
            true,
            true,
            true
          ],
          "responseTime": "00:00:00.000126",
          "successful": true
        },
        "9000": {
          "individualResponseTimes": [
            "00:00:00.000295",
            "00:00:00.000257",
            "00:00:00.000172",
            "00:00:00.000172",
            "00:00:00.000267"
          ],
          "individualStatus": [
            true,
            true,
            true,
            true,
            true
          ],
          "responseTime": "00:00:00.000233",
          "successful": true
        }
      },
      "svip": "172.27.62.50"
    },
    "duration": "00:00:00.421907",
    "result": "Passed"
  }
}
```

发布以下版本后新增

9.6

TestDrives

您可以使用 `TestDrives` 方法对节点上的所有驱动器运行硬件验证。此方法可检测到驱动器上的硬件故障，并会在验证测试结果中报告所有故障。

参数

`TestDrives` 方法仅适用于集群中的非“Active”节点。

注：该测试需要约 10 分钟。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
force	设置为 true，可测试节点上的驱动器。	布尔型	无	是
minutes	指定测试运行的分钟数。	整型	10	否

返回值

此方法具有以下返回值：

名称	说明	类型
details	有关测试操作成功与否的信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法将返回一个表，其中包含节点中每个驱动器的测试结果。

发布以下版本后新增

9.6

TestHardwareConfig

您可以使用 `TestHardwareConfig` 方法对节点执行硬件测试。测试选项包括验证硬件配置、固件版本以及验证所有驱动器是否都存在。

参数

注：这些测试不用于检测硬件故障。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
clean	使用干净的缓存启动硬件配置测试。 可能值： <ul style="list-style-type: none"> • true: 删除缓存的测试结果文件并重新运行测试。 • false: 检索缓存的测试结果。 	布尔型	false	否
force	要成功重置节点，此方法中必须包含 force 参数。	布尔型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
details	硬件配置详细信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

相关参考

[TestHardwareConfig](#) on page 558

TestHardwareConfig 方法返回类似于以下示例的响应。

TestLocateCluster

您可以使用 TestLocateCluster 方法验证节点能否找到集群配置中指定的集群。输出会验证是否已创建集群，同时会列出集群集合中的节点。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
details	有关测试操作成功与否的信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

TestLocalConnectivity

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
details	本地活动集群中每个节点的单个 ping 响应时间。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的请求类似于以下示例：

```
{  
  "id": null,  
  "result": {  
    "details": {  
      "10.26.86.17": {  
        individualResponseTimes: [  
          "00:00:00.006868",  
          "00:00:00.005933",  
          "00:00:00.006655",  
          "00:00:00.006584",  
          "00:00:00.006334"  
        ],  
        individualStatus: [  
          true,  
          true,  
          true,  
          true,  
          true  
        ],  
        responseTime: "00:00:00.006475",  
        successful: true  
      },  
      "10.26.86.18": {  
        individualResponseTimes: [  
          "00:00:00.006201",  
          "00:00:00.006187",  
          "00:00:00.005990",  
          "00:00:00.006029",  
          "00:00:00.005917"  
        ],  
        individualStatus: [  
          true,  
          true,  
          true,  
          true,  
          true  
        ],  
        responseTime: "00:00:00.006065",  
        successful: true  
      },  
      "10.26.86.19": {  
        individualResponseTimes: [  

```

```
        "00:00:00.005988",
        "00:00:00.006948",
        "00:00:00.005981",
        "00:00:00.005964",
        "00:00:00.005942"
    ],
    individualStatus: [
        "true",
        "true",
        true,
        true,
        true
    ],
    responseTime: "00:00:00.006165",
    successful: true,
},
    "10.26.86.20": {
    individualResponseTimes: [
        "00:00:00.005926",
        "00:00:00.006072",
        "00:00:00.005675",
        "00:00:00.009904",
        "00:00:00.006225"
    ],
        "individualStatus": [
            true,
            true,
            true,
            true,
            true
        ],
        responseTime: "00:00:00.006760",
        successful: true
    }
},
    "duration": "00:00:00.595982",
    "result": "Passed"
}
}
```

发布以下版本后新增

9.6

TestNetworkConfig

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

参数

在 UI 或 TUI 中使用 `SetNetworkConfig` 方法配置节点时，系统会验证并存储此配置。`TestNetworkConfig` API 测试将使用存储的配置执行验证后逻辑操作。例如，如果发生断电或网络故障，您可以使用此 API 方法来确保节点正以最新存储的网络配置运行。这可验证配置无错误且正在使用最新配置。

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

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
details	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所发现的所有错误。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例 1

如果未检测到任何错误，则不会返回任何响应。

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

响应示例 2

MTU 不匹配示例。

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

响应示例 3

缺少静态路由的示例。

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

```

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

```

发布以下版本后新增

9.6

相关参考

[SetNetworkConfig](#) on page 239

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

TestPing

您可以使用 `TestPing` 方法通过使用 ICMP 数据包的 1 G 和 10 G 接口测试与集群中所有节点的网络连接。该测试将根据网络配置中的 MTU 设置对每个数据包使用大小合适的 MTU。`TestPing` 不会创建临时 VLAN 接口。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
attempts	指定系统应重复测试 ping 的次数。	整型	5	否
hosts	指定要执行 ping 操作的设备的地址或主机名的列表，以英文逗号分隔。如果未指定任何主机、则会使用 ping 方法 ping 存储集群中的主机。	字符串	无	否
interface	用于发送 ping 的现有（基本）接口。可能值： <ul style="list-style-type: none"> Bond10G: 从绑定 10G 接口发送 ping。 Bond1G: 从绑定 1G 接口发送 ping。 	字符串	无	否
packetSize	指定要在发送给每个 IP 的 ICMP 数据包中发送的字节数。该字节数必须小于在网络配置中指定的最大 MTU。	整型	无	否

名称	说明	类型	默认值	必需
pingTimeoutMsec	指定等待各 ping 响应的时间（毫秒）。	整型	500 毫秒	否
prohibitFragmentation	启用 ICMP 数据包的不分段 (DF, Do not Fragment) 标志。	布尔型	false	否
sourceAddressV4	要在 ICMP ping 数据包中使用的源 IPv4 地址。	字符串	无	否
sourceAddressV6	要在 ICMP ping 数据包中使用的源 IPv6 地址。	字符串	无	否
totalTimeoutSec	指定在发出下一次 ping 尝试命令或结束此过程之前，ping 应等待系统响应的的时间（以秒为单位）。	整型	5	否
virtualNetworkTag	要在发送 ping 数据包时使用的 VLAN ID。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
details	可与此节点进行通信的各个 IP 的列表，以及 ping 响应统计信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "details": {
      "192.168.0.1": {
        "individualResponseCodes": [
          "Success",
          "Success",
          "Success",
          "Success",
          "Success"
        ],
        "individualResponseTimes": [
```

```
        "00:00:00.000304",
        "00:00:00.000123",
        "00:00:00.000116",
        "00:00:00.000113",
        "00:00:00.000111"
    ],
    "individualStatus": [
        true,
        true,
        true,
        true,
        true
    ],
    "interface": "Bond1G",
    "responseTime": "00:00:00.000154",
    "sourceAddressV4": "192.168.0.5",
    "successful": true
    }
},
"duration": "00:00:00.001747",
"result": "Passed"
}
}
```

发布以下版本后新增

5.0

TestRemoteConnectivity

您可以使用 `TestRemoteConnectivity` 方法对远程集群的每个节点执行 ping 操作并检查远程集合数据库连接。为了通过此方法返回有用的结果，您必须使用配对集群。如果远程数据库连接失败，则系统响应会列出异常。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
details	每个节点的单个 ping 响应时间。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": null,
  "result": {
    "details": {
      "1": {
        "details": {
          "10.26.86.17": {
            "individualResponseTimes": [
              "00:00:00.006868",
              "00:00:00.005933",
              "00:00:00.006655",
              "00:00:00.006584",
              "00:00:00.006334"
            ],
            "individualStatus": [
              "true",
              "true",
              "true",
              "true",
              "true"
            ],
            "responseTime": "00:00:00.006475",
            "successful": true
          },
          "10.26.86.18": {
            "individualResponseTimes": [
              "00:00:00.006201",
              "00:00:00.006187",
              "00:00:00.005990",
              "00:00:00.006029",
              "00:00:00.005917"
            ],
            "individualStatus": [
              "true",
              "true",
              "true",
              "true",
              "true"
            ],
            "responseTime": "00:00:00.006065",
            "successful": true
          },
          "10.26.86.19": {
            "individualResponseTimes": [
              "00:00:00.005988",
              "00:00:00.006948",
              "00:00:00.005981",
              "00:00:00.005964",
              "00:00:00.005942"
            ],
            "individualStatus": [
              "true",
              "true",
              "true",
              "true",
              "true"
            ]
          }
        }
      }
    }
  }
}
```

```
    ],  
    "responseTime": "00:00:00.006165",  
    "successful": true,  
  },  
  "10.26.86.20": {  
    "individualResponseTimes": [  
      "00:00:00.005926",  
      "00:00:00.006072",  
      "00:00:00.005675",  
      "00:00:00.009904",  
      "00:00:00.006225"  
    ],  
    "individualStatus": [  
      "true",  
      "true",  
      "true",  
      "true",  
      "true"  
    ],  
    "responseTime": "00:00:00.006760",  
    "successful": true  
  }  
},  
"successful": true  
}  
},  
"duration": "00:00:00.595982",  
"result": "Passed"  
}  
}
```

发布以下版本后新增

9.6

复制 API 方法

借助复制 API 方法，您可以将两个集群连接起来，以实现持续数据保护 (Continuous Data Protection, CDP)。当您将两个集群连接起来后，一个集群中的活动卷便可持续复制到另一个集群，从而实现数据恢复。通过配对卷以进行复制，您可以保护数据免受可能导致其无法访问的事件影响。

集群配对操作顺序

必须先在一对运行 Element 软件的存储集群之间建立连接，然后才能使用远程复制。

可使用下面这组 API 方法建立集群连接：

- ***StartClusterPairing***:
此 API 方法会创建并返回一个用于建立集群对的配对密钥。该密钥已经过编码，包含用于在集群之间建立通信的信息。一个集群最多可与另外四个集群进行配对。但是，必须为每个集群配对生成一个新密钥。*StartClusterPairing* 方法在每次调用时都会生成一个新密钥。在 *CompleteClusterPairing* 方法中使用每个唯一密钥即可与另外一个集群配对。
注：出于安全原因，不要通过电子邮件将配对密钥发送给其他用户。该密钥包含一个用户名和密码。
- ***CompleteClusterPairing***:
此方法将使用通过 *StartClusterPairing* API 方法创建的配对密钥来创建集群对。向目标发出带有 `clusterPairingKey` 参数的 *CompleteClusterPairing* API 方法。发起调用的集群为创建该密钥的集群。

相关参考

[StartClusterPairing on page 268](#)

您可以使用 `StartClusterPairing` 方法从用来与其他集群配对的集群创建编码密钥。使用此 API 方法创建的密钥将在 `CompleteClusterPairing` 方法中用于建立集群配对。一个集群最多可以与四个其他集群配对使用。

[CompleteClusterPairing on page 261](#)

`CompleteClusterPairing` 方法是集群配对过程中的第二个步骤。要完成集群配对过程，需将此方法与从 `StartClusterPairing` 方法收到的编码密钥结合使用。

卷配对操作顺序

必须先两个对应集群之间创建集群对，之后才能为卷配对。

可使用下面这组 API 方法建立集群连接：

- ***StartVolumePairing***:
此 API 方法会创建并返回一个用于创建卷对的卷配对密钥。该密钥包含用于在卷之间建立通信的信息。
- ***CompleteVolumePairing***:
此方法将使用通过 *StartVolumePairing* API 方法创建的配对密钥来创建卷对。向目标卷发出带有 `volumeID` 和 `volumePairingKey` 参数的 *CompleteVolumePairing* API 方法。

只能将其中的一个配对卷标识为复制目标卷。使用 *ModifyVolumePair* API 方法可以通过标识目标卷来确定卷的数据复制方向。数据会从源卷复制到目标卷。

相关参考

[StartVolumePairing on page 269](#)

您可以使用 `StartVolumePairing` 方法从用来与其他卷配对的卷创建编码密钥。此方法创建的密钥将在 `CompleteVolumePairing` 方法中用于建立卷配对。

[Complete Volume Pairing on page 262](#)

使用 `CompleteVolumePairing` 可以完成两个卷的配对过程。

[Modify Volume Pair on page 266](#)

您可以使用 `ModifyVolumePair` 方法暂停或重新启动一对卷之间的复制。应在源卷（具有读取/写入访问权限的卷）上设置此方法。

配对集群支持的复制模式

配对集群支持下列复制模式：

- 数据的异步复制：数据会异步发送到复制目标卷。系统不会等待发送确认消息即会写入数据。
- 数据的同步复制：数据会同步发送到复制目标卷。一旦系统确认从主机发送的 I/O 操作，就会向该主机发送系统确认消息，并将数据发送至复制目标卷。
- 数据的仅快照复制：只有卷快照才会复制到目标集群。

CompleteClusterPairing

`CompleteClusterPairing` 方法是集群配对过程中的第二个步骤。要完成集群配对过程，需将此方法与从 `StartClusterPairing` 方法收到的编码密钥结合使用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>clusterPairingKey</code>	一个字符串，包含从 StartClusterPairing API 方法返回的字符。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
<code>clusterPairID</code>	集群对的唯一标识符。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CompleteClusterPairing",
  "params": {
    "clusterPairingKey" :
    "7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223a2231636561313333
    6322d346338662d343631612d626537322d373435363661393533643266222c22636c7573746572556e6971756
    54944223a2278736d36222c226d766970223a223139322e3136382e3133392e313232222c226e616d65223a224
    175746f54657374322d6330755222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222
    c22727063436f6e6e656374696f6e4944223a3931333134323634392c22757365726e616d65223a225f5f53465
    f706169725f50597a796647704c7246564432444a42227d"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[StartClusterPairing](#) on page 268

您可以使用 `StartClusterPairing` 方法从用来与其他集群配对的集群创建编码密钥。使用此 API 方法创建的密钥将在 `CompleteClusterPairing` 方法中用于建立集群配对。一个集群最多可以与四个其他集群配对使用。

CompleteVolumePairing

使用 `CompleteVolumePairing` 可以完成两个卷的配对过程。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	将完成卷配对的卷的 ID。	整型	无	是
volumePairingKey	StartVolumePairing API 方法返回的密钥。	字符串	无	是

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": "CompleteVolumePairing",
  "params": {
    "volumeID" : 12,
    "volumePairingKey" :
    "7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223a2231636561313333
    6322d346338662d343631612d626537322d373435363661393533643266222c22636c75737465725556e6971756
    54944223a2278736d36222c226d766970223a223139322e3136382e31333392e313232222c226e616d65223a224
    175746f54657374322d6330755222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222
    c22727063436f6e6e656374696f6e4944223a3931333134323634392c22757365726e616d65223a225f5f53465
    f706169725f50597a796647704c7246564432444a42227d"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

相关参考

[Start VolumePairing](#) on page 269

您可以使用 `StartVolumePairing` 方法从用来与其他卷配对的卷创建编码密钥。此方法创建的密钥将在 `CompleteVolumePairing` 方法中用于建立卷配对。

ListClusterPairs

您可以使用 `ListClusterPairs` 方法列出与当前集群配对的所有集群。此方法将返回有关活动和特定集群配对的信息，例如有关当前配对以及集群配对连接性和延迟性（以毫秒为单位）的统计信息。

参数

此方法没有任何输入参数：

返回值

此方法具有以下返回值：

名称	说明	类型
<code>clusterPairs</code>	有关每个配对集群的信息。	<code>clusterPair</code> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListClusterPairs",
  "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": {
    "clusterPairs": [
      {
        "clusterName": "cluster2",
        "clusterPairID": 3,
        "clusterPairUUID": "9866fbeb-c2f8-4df3-beb9-58a5c4e49c9b",
        "clusterUUID": 5487,
        "latency": 1,
        "mvip": "172.1.1.5",

```

```
    "status": "Connected",
    "version": "8.0.0.1361"
  },
  {
    "clusterName": "cluster3",
    "clusterPairID": 2,
    "clusterPairUUID": "8132a699-ce82-41e0-b406-fb914f976042",
    "clusterUUID": 1383,
    "latency": 1,
    "mvip": "172.1.1.6",
    "status": "Connected",
    "version": "8.0.0.1361"
  }
]
}
```

发布以下版本后新增

9.6

ListActivePairedVolumes

您可以使用 `ListActivePairedVolumes` 方法列出与某个卷配对的所有活动卷。此方法将返回有关活动和待定配对的卷的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
volumes	配对卷的卷信息。	<i>volumePair</i> 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应类似于以下示例：

```
{
  "id": 1,
  "result": {
    "volumes": [
      {
        "access": "readWrite",
        "accountID": 1,
        "attributes": {},
        "blockSize": 4096,
        "createTime": "2016-06-24T15:21:59Z",
```

```
"deleteTime": "",
"enable512e": true,
"iqn": "iqn.2010-01.com.solidfire:0oto.bk.24",
"name": "BK",
"purgeTime": "",
"qos": {
  "burstIOPS": 15000,
  "burstTime": 60,
  "curve": {
    "4096": 100,
    "8192": 160,
    "16384": 270,
    "32768": 500,
    "65536": 1000,
    "131072": 1950,
    "262144": 3900,
    "524288": 7600,
    "1048576": 15000
  },
  "maxIOPS": 15000,
  "minIOPS": 50
},
"scsiEUIDeviceID": "306f746f00000018f47acc0100000000",
"scsiNAADeviceID": "6f47acc100000000306f746f00000018",
"sliceCount": 1,
"status": "active",
"totalSize": 10737418240,
"virtualVolumeID": null,
"volumeAccessGroups": [],
"volumeID": 24,
"volumePairs": [
  {
    "clusterPairID": 2,
    "remoteReplication": {
      "mode": "Async",
      "pauseLimit": 3145728000,
      "remoteServiceID": 14,
      "resumeDetails": "",
      "snapshotReplication": {
        "state": "Idle",
        "stateDetails": ""
      },
      "state": "Active",
      "stateDetails": ""
    },
    "remoteSliceID": 8,
    "remoteVolumeID": 8,
    "remoteVolumeName": "PairingDoc",
    "volumePairUUID": "229fcbf3-2d35-4625-865a-d04bb9455cef"
  }
]
}
}
}
}
```

发布以下版本后新增

9.6

ModifyVolumePair

您可以使用 `ModifyVolumePair` 方法暂停或重新启动一对卷之间的复制。应在源卷（具有读取/写入访问权限的卷）上设置此方法。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>volumeID</code>	要修改的卷的标识号。	整型	无	是
<code>pausedManual</code>	可以在源（读取/写入）卷上暂停或重新启动远程复制。可能值： <ul style="list-style-type: none"><code>true</code>：暂停卷复制。<code>false</code>：重新启动卷复制。 如果未指定任何值，则复制过程不会发生变化。	布尔型	无	否
<code>mode</code>	卷复制模式。可能值： <ul style="list-style-type: none"><code>Async</code>：写入操作在本地完成后确认。集群不会等待写入复制到目标集群。<code>Sync</code>：将数据存储在本地和远程集群上后，源将确认写入。<code>SnapshotsOnly</code>：仅复制在源集群上创建的快照。而不会复制从源卷写入的活动数据。	字符串	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyVolumePair",
  "params": {
    "pausedManual": false,
    "volumeID": 5,
    "mode": "sync"
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

RemoveClusterPair

您可以使用 `RemoveClusterPair` 方法关闭在两个配对集群之间打开的连接。

参数

注: 在删除集群对之前，您必须先使用 `RemoveVolumePair` API 方法删除集群的所有卷配对。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
clusterPairID	用于将两个集群配对的唯一标识符。	整型	无	是

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": "RemoveClusterPair",
  "params": {
    "clusterPairID": 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": {}
}
```

发布以下版本后新增

9.6

RemoveVolumePair

您可以使用 `RemoveVolumePair` 方法删除两个卷之间的远程配对。应对配对的源卷和目标卷同时使用此方法。当您删除卷配对信息时，相关卷将不再发生数据复制。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	要停止复制过程的卷的 ID。	整型	无	是

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": "RemoveVolumePair",
  "params": {
    "volumeID": 5
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

StartClusterPairing

您可以使用 `StartClusterPairing` 方法从用来与其他集群配对的集群创建编码密钥。使用此 API 方法创建的密钥将在 `CompleteClusterPairing` 方法中用于建立集群配对。一个集群最多可以与四个其他集群配对使用。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
clusterPairingKey	一个字符串，由 <i>CompleteClusterPairing</i> API 方法使用。	字符串
clusterPairID	集群对的唯一标识符。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "StartClusterPairing",
  "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": {
    "clusterPairID": 1,
    "clusterPairingKey":
"7b22636c7573746572506169724944223a312c22636c75737465725061697255554944223a223163656131333
6322d346338662d343631612d626537322d373435363661393533643266222c22636c7573746572556e6971756
54944223a2278736d36222c226d766970223a223139322e3136382e3133392e313232222c226e616d65223a224
175746f54657374322d6330755222c2270617373776f7264223a22695e59686f20492d64774d7d4c67614b222
c22727063436f6e6e656374696f6e4944223a3931333134323634392c22757365726e616d65223a225f5f53465
f706169725f50597a796647704c7246564432444a42227d"
  }
}
```

发布以下版本后新增

9.6

相关参考

[CompleteClusterPairing](#) on page 261

CompleteClusterPairing 方法是集群配对过程中的第二个步骤。要完成集群配对过程，需将此方法与从 StartClusterPairing 方法收到的编码密钥结合使用。

StartVolumePairing

您可以使用 StartVolumePairing 方法从用来与其他卷配对的卷创建编码密钥。此方法创建的密钥将在 CompleteVolumePairing 方法中用于建立卷配对。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
mode	要启动配对过程的卷的模式。您只能对源卷设置此模式。可能值： <ul style="list-style-type: none">Async：写入操作在本地完成后确认。集群不会等待写入复制到目标集群。（如果未指定 mode 参数，将使用默认值。）Sync：将数据存储在本地和远程集群上后，源将确认写入。SnapshotsOnly：仅复制在源集群上创建的快照。而不会复制从源卷写入的活动数据。	字符串	无	否
volumeID	要启动配对过程的卷的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
volumePairingKey	一个字符串，由 Complete Volume Pairing API 方法使用。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "StartVolumePairing",
  "params": {
    "mode": "Async",
    "volumeID" : 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" : {
    "volumePairingKey" :
    "7b226d766970223a223139322e3136382e3133392e313232222c22766f6c756d654944223a312c22766f6c756
    d654e616d65223a2254657374222c22766f6c756d655061697255554944223a2236393632346663622d3230326
    52d343332352d613536392d656339633635356337623561227d"
  }
}
```

发布以下版本后新增

9.6

相关参考

[Complete Volume Pairing](#) on page 262

使用 CompleteVolumePairing 可以完成两个卷的配对过程。

安全 API 方法

您可以将 Element 软件与外部安全相关服务（例如外部密钥管理服务器）集成在一起。您可以通过这些安全相关方法配置 Element 安全功能，例如用于空闲加密的外部密钥管理。

AddKeyServerToProviderKmip

您可以使用 AddKeyServerToProviderKmip 方法将密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器分配给指定的密钥提供程序。分配期间会访问此服务器以验证是否运行正常。如果已将指定密钥服务器分配给指定密钥提供程序，则不会执行任何操作，也不会返回任何错误。您可以使用 RemoveKeyServerFromProviderKmip 方法删除此分配。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyProviderID	要将密钥服务器分配到的密钥提供程序的 ID。	整型	无	是
keyServerID	要分配的密钥服务器的 ID。	整型	无	是

返回值

此方法没有返回值。只要没有返回错误，此分配就会视为成功。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "AddKeyServerToProviderKmip",
  "params": {
    "keyProviderID": 1,
    "keyServerID": 15
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.7

CreateKeyProviderKmip

您可以使用 `CreateKeyProviderKmip` 方法创建具有指定名称的密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥提供程序。密钥提供程序定义了一种用于检索身份验证密钥的机制，可从此密钥提供程序检索此密钥。在创建新的 KMIP 密钥提供程序时，不会为此密钥提供程序分配任何 KMIP 密钥服务器。要创建 KMIP 密钥服务器，请使用 `CreateKeyServerKmip` 方法。要将此密钥服务器分配给提供程序，请参见 `AddKeyServerToProviderKmip`。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>keyProviderName</code>	要与创建的 KMIP 密钥提供程序关联的名称。此名称仅用于显示，不需要唯一。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
<code>kmipKeyProvider</code>	一个对象，其中包含有关新创建的密钥提供程序的详细信息。	<i>KeyProviderKmip</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateKeyProviderKmip",
  "params": {
    "keyProviderName": "ProviderName",
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "kmipKeyProvider": {
      "keyProviderName": "ProviderName",
      "keyProviderIsActive": true,
      "kmipCapabilities": "SSL",
      "keyServerIDs": [
        15
      ],
      "keyProviderID": 1
    }
  }
}
```

发布以下版本后新增

11.7

CreateKeyServerKmip

您可以使用 `CreateKeyServerKmip` 方法创建具有指定属性的密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器。创建期间不会访问此服务器；使用此方法之前，不需要事先存在此服务器。对于集群模式密钥服务器配置，必须在 `kmipKeyServerHostnames` 参数中提供所有服务器节点的主机名或 IP 地址。您可以使用 `TestKeyServerKmip` 方法测试密钥服务器。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>kmipCaCertificate</code>	外部密钥服务器根 CA 的公共密钥证书。在 TLS 通信中会使用此证书来验证外部密钥服务器提供的证书。对于各个服务器使用不同 CA 的密钥服务器集群，应提供一个串联字符串来列出所有这些 CA 的根证书。	字符串	无	是
<code>kmipClientCertificate</code>	SolidFire KMIP 客户端使用的 PEM 格式 Base64 编码 PKCS#10 X.509 证书。	字符串	无	是
<code>kmipKeyServerHostnames</code>	与此 KMIP 密钥服务器关联的主机名或 IP 地址数组。只有当密钥服务器采用集群模式配置时，才需要提供多个主机名或 IP 地址。	字符串数组	无	是
<code>kmipKeyServerName</code>	此 KMIP 密钥服务器的名称。此名称仅用于显示，不需要唯一。	字符串	无	是
<code>kmipKeyServerPort</code>	与此 KMIP 密钥服务器关联的端口号（通常为 5696）。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
kmipKeyServer	一个对象，其中包含有关新创建的密钥服务器的详细信息。	KeyServerKmip

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateKeyServerKmip",
  "params": {
    "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
    "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
    "kmipKeyServerHostnames" : ["server1.hostname.com", "server2.hostname.com"],
    "kmipKeyServerName" : "keyserverName",
    "kmipKeyServerPort" : 5696
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "kmipKeyServer": {
      "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
      "kmipKeyServerHostnames": [
        "server1.hostname.com", "server2.hostname.com"
      ],
      "keyProviderID": 1,
      "kmipKeyServerName": "keyserverName",
      "keyServerID": 1,
      "kmipKeyServerPort": 1,
      "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
      "kmipAssignedProviderIsActive": true
    }
  }
}
```

发布以下版本后新增

11.7

CreatePublicPrivateKeyPair

您可以使用 `CreatePublicPrivateKeyPair` 方法创建公共和专用 SSL 密钥。您可以使用这些密钥生成证书签名请求。每个存储集群只能有一个正在使用的密钥对。在使用此方法替换现有密钥之前，请确保这些密钥不再由任何提供程序使用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
commonName	X.509 可分辨名称 Common Name 字段（CN）。	字符串	无	否
country	X.509 标识名 Country 字段（C）。	字符串	无	否
emailAddress	X.509 可分辨名称 Email Address 字段（邮件）。	字符串	无	否
locality	X.509 标识名 Locality Name 字段（L）。	字符串	无	否
organization	X.509 可分辨名称 Organization Name 字段（O）。	字符串	无	否
organizationalUnit	X.509 可分辨名称 Organizational Unit Name 字段（OU）。	字符串	无	否
state	X.509 标识名 State 或 Province Name 字段（ST 或 SP 或 S）。	字符串	无	否

返回值

此方法没有返回值。如果未发生错误，密钥创建将视为成功。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreatePublicPrivateKeyPair",
  "params": {
    "commonName": "Name",
    "country": "US",
    "emailAddress": "email@domain.com"
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.7

DeleteKeyProviderKmip

您可以使用 `DeleteKeyProviderKmip` 方法删除指定的非活动密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥提供程序。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyProviderID	要删除的密钥提供程序的 ID。	整型	无	是

返回值

此方法没有返回值。只要没有发生错误，此删除操作就会视为成功。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "DeleteKeyProviderKmip",
  "params": {
    "keyProviderID": "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": {}
}
```

发布以下版本后新增

11.7

DeleteKeyServerKmip

您可以使用 `DeleteKeyServerKmip` 方法删除现有密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器。只要此密钥服务器不是分配给其提供程序的最后一个密钥服务器且此提供程序未提供当前正在使用的密钥，您就可以删除此密钥服务器。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyServerID	要删除的 KMIP 密钥服务器的 ID。	整型	无	是

返回值

此方法没有返回值。如果没有发生错误，此删除操作就会视为成功。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "DeleteKeyServerKmip",
  "params": {
    "keyServerID": 15
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.7

DisableEncryptionAtRest

您可以使用 `DisableEncryptionAtRest` 方法删除先前通过 `EnableEncryptionAtRest` 方法应用于集群的加密。此禁用方法是异步的，将在禁用加密之前返回响应。可以使用 `GetClusterInfo` 方法对系统进行轮询，以查看进程完成的时间。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "DisableEncryptionAtRest",
  "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" : {}
}
```

发布以下版本后新增

9.6

相关参考

[GetClusterInfo](#) on page 119

您可以使用 `GetClusterInfo` 方法返回有关集群的配置信息。

EnableEncryptionAtRest

您可以使用 `EnableEncryptionAtRest` 方法在集群上启用高级加密标准 (Advanced Encryption Standard, AES) 256 位空闲加密，以便集群可以管理用于每个节点上的驱动器的加密密钥。默认情况下不会启用此功能。

如果启用空闲加密，集群会自动在内部管理用于每个节点上的驱动器的加密密钥。节点不会存储用于解锁驱动器的密钥，而且密钥永不通过网络传递。需要有两个加入集群的节点，这样才能访问用于在驱动器上禁用加密的密钥。加密管理不影响集群的性能或效率。如果通过 API 从集群中删除已启用加密的驱动器或节点，则会禁用空闲加密，并且无法安全擦除数据。可以使用 `SecureEraseDrives` API 方法安全擦除数据。

如果指定了 `keyProviderID`，则会根据密钥提供程序的类型生成和检索密码。对于 KMIP 密钥提供程序，通常使用密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器来完成此操作。执行此操作后，指定的提供程序将被视为处于活动状态且无法删除，除非使用 `DisableEncryptionAtRest` 方法禁用空闲加密。

注：如果您的节点类型的型号以 `-NE` 结尾，则 `EnableEncryptionAtRest` 方法调用将失败并显示响应 “Encryption not allowed.Cluster detected non-encryptable node”。

注：仅当集群正在运行并且处于正常状态时，才可启用或禁用加密。您可以根据需要决定启用或禁用加密。

注：此进程是异步的，将在启用加密之前返回响应。可以使用 `GetClusterInfo` 方法对系统进行轮询，以查看进程完成的时间。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>keyProviderID</code>	要使用的 KMIP 密钥提供程序的 ID。	整型	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回的响应类似于 `EnableEncryptionAtRest` 方法中的以下示例。没有任何结果可供报告。

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

```
    "result": {}  
  }
```

在集群上启用空闲加密后，`GetClusterInfo` 返回的结果会将空闲加密状态（“`encryptionAtRestState`”）显示为“`enabling`”。完全启用空闲加密后，返回的状态将变为“`enabled`”。

```
{  
  "id": 1,  
  "result": {  
    "clusterInfo": {  
      "attributes": { },  
      "encryptionAtRestState": "enabling",  
      "ensemble": [  
        "10.10.5.94",  
        "10.10.5.107",  
        "10.10.5.108"  
      ],  
      "mvip": "192.168.138.209",  
      "mvipNodeID": 1,  
      "name": "Marshall",  
      "repCount": 2,  
      "svip": "10.10.7.209",  
      "svipNodeID": 1,  
      "uniqueID": "91dt"  
    }  
  }  
}
```

发布以下版本后新增

9.6

相关参考

[SecureEraseDrives](#) on page 162

您可以使用 `SecureEraseDrives` 方法从状态为“`available`”的驱动器中删除所有残留数据。如果要更换服务寿命即将结束且包含敏感数据的驱动器，您可以使用此方法。此方法会使用安全擦除单元命令将预先确定的模式写入驱动器中，并重置驱动器上的加密密钥。此异步方法可能需要几分钟的时间才能完成。

[GetClusterInfo](#) on page 119

您可以使用 `GetClusterInfo` 方法返回有关集群的配置信息。

GetClientCertificateSignRequest

您可以使用 `GetClientCertificateSignRequest` 方法生成一个证书签名请求，证书颁发机构可对此证书签名请求进行签名，以便为集群生成客户端证书。要建立信任关系以与外部服务进行交互，需要使用签名证书。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>clientCertificateSignRequest</code>	PEM 格式 Base64 编码 PKCS#10 X.509 客户端证书签名请求。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetClientCertificateSignRequest",
  "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": {
    "clientCertificateSignRequest":
    "MIIBYjCCATMCAQAwgYkxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEWpDYWxpZm9ybm..."
  }
}
```

发布以下版本后新增

11.7

GetKeyProviderKmip

您可以使用 `GetKeyProviderKmip` 方法检索有关指定密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥提供程序的信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyProviderID	要返回的 KMIP 密钥提供程序对象的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
kmipKeyProvider	一个对象，其中包含有关请求的密钥提供程序的详细信息。	KeyProviderKmip

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetKeyProviderKmip",
  "params": {
    "keyProviderID": 15
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "kmipKeyProvider": {
      "keyProviderID": 15,
      "kmipCapabilities": "SSL",
      "keyProviderIsActive": true,
      "keyServerIDs": [
        1
      ],
      "keyProviderName": "ProviderName"
    }
  }
}
```

发布以下版本后新增

11.7

GetKeyServerKmip

您可以使用 `GetKeyServerKmip` 方法返回有关指定密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器的信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyServerID	要返回其相关信息的 KMIP 密钥服务器的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
kmipKeyServer	一个对象，其中包含有关请求的密钥服务器的详细信息。	KeyServerKmip

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetKeyServerKmip",
  "params": {
    "keyServerID": 15
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{
  "id": 1,
  "result": {
    "kmipKeyServer": {
      "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
      "kmipKeyServerHostnames": [
        "server1.hostname.com", "server2.hostname.com"
      ],
      "keyProviderID": 1,
      "kmipKeyServerName": "keyserverName",
      "keyServerID": 15,
      "kmipKeyServerPort": 1,
      "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
      "kmipAssignedProviderIsActive": true
    }
  }
}
```

发布以下版本后新增

11.7

ListKeyProvidersKmip

您可以使用 `ListKeyProvidersKmip` 方法检索所有现有密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥提供程序的列表。您可以通过指定其他参数对此列表进行筛选。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
keyProviderIsActive	<p>根据是否处于活动状态筛选返回的 KMIP 密钥服务器对象。可能值：</p> <ul style="list-style-type: none"> • true: 仅返回处于活动状态的 KMIP 密钥提供程序（提供当前正在使用的密钥）。 • false: 仅返回处于非活动状态的 KMIP 密钥提供程序（未提供任何密钥且可删除）。 <p>如果省略此参数，则不会根据是否处于活动状态对返回的 KMIP 密钥提供程序进行筛选。</p>	布尔型	无	否
kmipKeyProviderHasServerAssigned	<p>根据是否已分配 KMIP 密钥服务器对返回的 KMIP 密钥提供程序进行筛选。可能值：</p> <ul style="list-style-type: none"> • true: 仅返回已分配 KMIP 密钥服务器的 KMIP 密钥提供程序。 • false: 仅返回未分配 KMIP 密钥服务器的 KMIP 密钥提供程序。 <p>如果省略此参数，则不会根据是否已分配 KMIP 密钥服务器对返回的 KMIP 密钥提供程序进行筛选。</p>	布尔型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
kmipKeyProviders	已创建的 KMIP 密钥提供程序的列表。	<i>KeyProviderKmp</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListKeyProvidersKmip",
  "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": {
    "kmipKeyProviders": [
      {
        "keyProviderID": 15,
        "kmipCapabilities": "SSL",
        "keyProviderIsActive": true,
        "keyServerIDs": [
          1
        ],
        "keyProviderName": "KeyProvider1"
      }
    ]
  }
}
```

发布以下版本后新增

11.7

ListKeyServersKmip

您可以使用 `ListKeyServersKmip` 方法列出已创建的所有密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器。您可以通过指定其他参数对结果进行筛选。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyProviderID	如果指定此参数，则此方法仅返回分配给指定 KMIP 密钥提供程序的 KMIP 密钥服务器。如果省略此参数，则不会根据是否已分配给指定的 KMIP 密钥提供程序对返回的 KMIP 密钥服务器进行筛选。	整型	无	否

名称	说明	类型	默认值	必需
kmpAssignedProviderIsActive	<p>根据是否处于活动状态筛选返回的 KMP 密钥服务器对象。可能值：</p> <ul style="list-style-type: none"> • true: 仅返回处于活动状态的 KMP 密钥服务器（提供当前正在使用的密钥）。 • false: 仅返回处于非活动状态的 KMP 密钥服务器（未提供任何密钥且可删除）。 <p>如果省略此参数，则不会根据是否处于活动状态对返回的 KMP 密钥服务器进行筛选。</p>	布尔型	无	否
kmpHasProviderAssigned	<p>根据是否已分配 KMP 密钥提供程序对返回的 KMP 密钥服务器进行筛选。可能值：</p> <ul style="list-style-type: none"> • true: 仅返回已分配 KMP 密钥提供程序的 KMP 密钥服务器。 • false: 仅返回未分配 KMP 密钥提供程序的 KMP 密钥服务器。 <p>如果省略此参数，则不会根据是否已分配 KMP 密钥提供程序对返回的 KMP 密钥服务器进行筛选。</p>	布尔型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
kmpKeyServers	已创建的 KMP 密钥服务器的完整列表。	<i>KeyServerKmp</i> 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "kmipKeyServers": [
    {
      "kmipKeyServerName": "keyserverName",
      "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
      "keyServerID": 15,
      "kmipAssignedProviderIsActive": true,
      "kmipKeyServerPort": 5696,
      "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
      "kmipKeyServerHostnames": [
        "server1.hostname.com", "server2.hostname.com"
      ],
      "keyProviderID": 1
    }
  ]
}
```

发布以下版本后新增

11.7

ModifyKeyServerKmip

您可以使用 `ModifyKeyServerKmip` 方法将现有密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器修改为指定属性。虽然唯一必需的参数是 `keyServerID`，但如果某个请求仅包含 `keyServerID`，则不会执行任何操作，也不会返回任何错误。您指定的任何其他参数将替换具有指定 `keyServerID` 的密钥服务器的现有值。在此操作过程中会访问此密钥服务器以确保其运行正常。您可以通过 `kmipKeyServerHostnames` 参数提供多个主机名或 IP 地址，但前提是密钥服务器采用集群模式配置。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>keyServerID</code>	要修改的 KMIP 密钥服务器的 ID。	整型	无	是

名称	说明	类型	默认值	必需
kmipCaCertificate	外部密钥服务器根 CA 的公共密钥证书。在 TLS 通信中会使用此证书来验证外部密钥服务器提供的证书。对于各个服务器使用不同 CA 的密钥服务器集群，应提供一个串联字符串来列出所有这些 CA 的根证书。	字符串	无	否
kmipClientCertificate	SolidFire KMIP 客户端使用的 PEM 格式 Base64 编码 PKCS#10 X.509 证书。	字符串	无	否
kmipKeyServerHostnames	与此 KMIP 密钥服务器关联的主机名或 IP 地址数组。只有当密钥服务器采用集群模式配置时，才需要提供多个主机名或 IP 地址。	字符串数组	无	否
kmipKeyServerName	此 KMIP 密钥服务器的名称。此名称仅用于显示，不需要唯一。	字符串	无	否
kmipKeyServerPort	与此 KMIP 密钥服务器关联的端口号（通常为 5696）。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
kmipKeyServer	一个对象，其中包含有关新修改的密钥服务器的详细信息。	KeyServerKmip

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyKeyServerKmip",
  "params": {
    "keyServerID": 15
    "kmipCaCertificate": "CPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
    "kmipClientCertificate": "kirWmnWXbj9T/UWZYB2oK0z5...",
    "kmipKeyServerHostnames" : ["server1.hostname.com", "server2.hostname.com"],
    "kmipKeyServerName" : "keyserverName",
    "kmipKeyServerPort" : 5696
  },
}
```

```
"id": 1  
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id": 1,  
  "result":  
    {  
      "kmipKeyServer": {  
        "kmipCaCertificate": "CPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",  
        "kmipKeyServerHostnames": [  
          "server1.hostname.com", "server2.hostname.com"  
        ],  
        "keyProviderID": 1,  
        "kmipKeyServerName": "keyserverName",  
        "keyServerID": 1  
        "kmipKeyServerPort": 1,  
        "kmipClientCertificate": "kirWmnWXbj9T/UWZYB2oK0z5...",  
        "kmipAssignedProviderIsActive": true  
      }  
    }  
}
```

发布以下版本后新增

11.7

RemoveKeyServerFromProviderKmpip

您可以使用 `RemoveKeyServerFromProviderKmpip` 方法将指定的密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器从为其分配的提供程序中取消分配。只要此密钥服务器不是最后一个密钥服务器且其提供程序未处于活动状态 (提供当前正在使用的密钥), 您就可以将此密钥服务器从其提供程序中取消分配。如果没有将指定的密钥服务器分配给提供程序, 则不会执行任何操作, 也不会返回任何错误。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
keyServerID	要取消分配的 KMIP 密钥服务器的 ID。	整型	无	是

返回值

此方法没有返回值。只要没有返回错误, 此删除就会视为成功。

请求示例

此方法的请求类似于以下示例:

```
{  
  "method": "RemoveKeyServerFromProviderKmpip",  
  "params": {  
    "keyServerID": 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":
    {}
}
```

发布以下版本后新增

11.7

TestKeyProviderKmip

您可以使用 `TestKeyProviderKmip` 方法测试指定的密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥提供程序是否可访问且运行正常。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyProviderID	要测试的密钥提供程序的 ID。	整型	无	是

返回值

此方法没有返回值。只要没有返回错误，此测试就会视为成功。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "TestKeyProviderKmip",
  "params": {
    "keyProviderID": 15
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.7

TestKeyServerKmip

您可以使用 `TestKeyServerKmip` 方法测试指定的密钥管理互操作性协议 (Key Management Interoperability Protocol, KMIP) 密钥服务器是否可访问且运行正常。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
keyServerID	要测试的 KMIP 密钥服务器的 ID。	整型	无	是

返回值

此方法没有返回值。如果没有返回错误，此测试就会视为成功。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "TestKeyServerKmip",
  "params": {
    "keyServerID": 15
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.7

SnapMirror API 方法

Element Web UI 可通过 SnapMirror API 方法管理使用远程 ONTAP 系统镜像的快照。这些方法仅供 Element Web UI 使用。如果您需要通过 API 访问 SnapMirror 功能，请使用 ONTAP API。本文不提供 SnapMirror API 方法的请求和返回示例。

AbortSnapMirrorRelationship

Element 软件 Web UI 可通过 AbortSnapMirrorRelationship 方法停止已启动但尚未完成的 SnapMirror 传输。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
destinationVolume	SnapMirror 关系中的目标卷。	snapMirrorVolumeInfo	无	是
clearCheckpoint	确定是否清除重新启动检查点。可能值： <ul style="list-style-type: none"> true false 	布尔型	false	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	包含有关已中止的 SnapMirror 关系的信息的对象。	snapMirrorRelationship

发布以下版本后新增

10.1

BreakSnapMirrorRelationship

Element Web UI 可通过 BreakSnapMirrorRelationship 方法中断 SnapMirror 关系。中断 SnapMirror 关系后，目标卷将设为独立的读写卷，之后，便可与源卷分离。可以使用 ResyncSnapMirrorRelationship API 方法重新建立此关系。此方法要求 ONTAP 集群可用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是

名称	说明	类型	默认值	必需
destinationVolume	SnapMirror 关系中的目标卷。	<i>snapMirrorVolumeInfo</i>	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	包含有关已中断的 SnapMirror 关系的对象的信息。	<i>snapMirrorRelationship</i>

发布以下版本后新增

10.1

相关参考

[BreakSnapMirrorVolume](#) on page 292

Element Web UI 可通过 `BreakSnapMirrorVolume` 方法中断 ONTAP 源容器和 Element 目标卷之间的 SnapMirror 关系。如果在向 Element 卷复制数据时 ONTAP 系统变得不可用，则中断 Element SnapMirror 卷会很有用。通过此功能，存储管理员可以控制 Element SnapMirror 卷、中断其与远程 ONTAP 系统的关系以及将此卷还原为先前的快照。

BreakSnapMirrorVolume

Element Web UI 可通过 `BreakSnapMirrorVolume` 方法中断 ONTAP 源容器和 Element 目标卷之间的 SnapMirror 关系。如果在向 Element 卷复制数据时 ONTAP 系统变得不可用，则中断 Element SnapMirror 卷会很有用。通过此功能，存储管理员可以控制 Element SnapMirror 卷、中断其与远程 ONTAP 系统的关系以及将此卷还原为先前的快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	要执行中断操作的卷。卷访问模式必须为 <code>snapMirrorTarget</code> 。	整型	无	是
snapshotID	将卷回滚到此 ID 所标识的快照。默认行为是，回滚到最新快照。	整型	无	否
preserve	保留比 <code>snapshotID</code> 标识的快照更新的所有快照。可能值： <ul style="list-style-type: none"> <code>true</code>：保留比 <code>snapshotID</code> 更新的快照。 <code>false</code>：不保留比 <code>snapshotID</code> 更新的快照。 如果为 <code>false</code> ，则删除比 <code>snapshotID</code> 更新的所有快照。	布尔型	<code>false</code>	否
access	生成的卷访问模式。可能值： <ul style="list-style-type: none"> <code>readWrite</code> <code>readOnly</code> <code>locked</code> 	字符串	<code>readWrite</code>	否

返回值

此方法没有返回值。

发布以下版本后新增

10.0

相关参考

[BreakSnapMirrorRelationship](#) on page 291

Element Web UI 可通过 `BreakSnapMirrorRelationship` 方法中断 SnapMirror 关系。中断 SnapMirror 关系后，目标卷将设为独立的读写卷，之后，便可与源卷分离。可以使用 `ResyncSnapMirrorRelationship` API 方法重新建立此关系。此方法要求 ONTAP 集群可用。

CreateSnapMirrorEndpoint

Element Web UI 可通过 `CreateSnapMirrorEndpoint` 方法创建与远程 SnapMirror 端点的关系。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
managementIP	远程 SnapMirror 端点的管理 IP 地址。	字符串	无	是
username	ONTAP 系统的管理用户名。	字符串	无	是
password	ONTAP 系统的管理密码。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorEndpoint	新创建的 SnapMirror 端点。	snapMirrorEndpoint

发布以下版本后新增

10.0

CreateSnapMirrorEndpointUnmanaged

Element 软件存储系统可通过 `CreateSnapMirrorEndpointUnmanaged` 方法使远程非受管 SnapMirror 端点能够与 Element 存储集群进行通信。不能使用 Element SnapMirror API 来管理非受管端点。必须使用 ONTAP 管理软件或 API 对其进行管理。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
clusterName	端点的名称。	字符串	无	是
ipAddresses	应与此 Element 存储集群通信的 ONTAP 存储系统集群的 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	说明	类型
snapMirrorEndpoint	新创建的 SnapMirror 端点。	snapMirrorEndpoint

发布以下版本后新增

10.3

CreateSnapMirrorRelationship

Element Web UI 可通过 CreateSnapMirrorRelationship 方法在源端点和目标端点之间创建 SnapMirror 扩展数据保护关系。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
sourceVolume	关系中的源卷。	snapMirrorVolumeInfo	无	是
destinationVolume	关系中的目标卷。	snapMirrorVolumeInfo	无	是
relationshipType	关系的类型。在运行 Element 软件的存储系统上，此值始终为“extended_data_protection”。	字符串	无	否
policyName	指定关系的 ONTAP SnapMirror 策略名称。如果不指定，则默认策略名称为 MirrorLatest。	字符串	无	否
scheduleName	ONTAP 系统上用于更新 SnapMirror 关系的原有 cron 计划的名称。如果不指定计划，则不会计划 snapMirror 更新，因此必须手动执行更新。	字符串	无	否
maxTransferRate	指定卷之间的最大数据传输速率（以 KB/秒为单位）。默认值 0 表示无限制，允许 SnapMirror 关系充分利用可用的网络带宽。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	有关新创建的 SnapMirror 关系的信息。	snapMirrorRelationship

发布以下版本后新增

10.1

CreateSnapMirrorVolume

Element Web UI 可通过 CreateSnapMirrorVolume 方法在远程 ONTAP 系统上创建卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
vserver	SVM 的名称。	字符串	无	是
name	目标 ONTAP 卷名称。	字符串	无	是
type	卷类型。可能值： <ul style="list-style-type: none"> rw: 读写卷 ls: 负载共享卷 dp: 数据保护卷 如果不提供类型，则默认类型为 dp。	字符串	无	否
aggregate	要在其中创建卷的 ONTAP 聚合。可以通过 ListSnapMirrorAggregates 获取有关可用 ONTAP 聚合的信息。	字符串	无	是
size	卷的大小（以字节为单位）。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorVolume	有关 SnapMirror 卷的信息。	snapMirrorVolume

发布以下版本后新增

10.1

DeleteSnapMirrorEndpoints

Element Web UI 可通过 `DeleteSnapMirrorEndpoints` 从系统中删除一个或多个 SnapMirror 端点。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointIDs	要删除的 SnapMirror 端点的 ID 数组	整型数组	无	是

返回值

此方法没有返回值。

发布以下版本后新增

10.0

DeleteSnapMirrorRelationships

Element Web UI 可通过 `DeleteSnapMirrorRelationships` 方法删除源端点与目标端点之间的一个或多个 SnapMirror 关系。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
destinationVolumes	SnapMirror 关系中的一个或多个目标卷。	<i>snapMirrorVolumeInfo</i> 数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
result	如果删除操作成功，此对象将包含一条成功消息。如果操作失败，此对象将包含一条错误消息。	JSON 对象

发布以下版本后新增

10.1

GetOntapVersionInfo

Element Web UI 可通过 `GetOntapVersionInfo` 从 SnapMirror 关系中的 ONTAP 集群获取有关 API 版本支持的信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	如果提供此参数，则系统将列出具有指定 snapMirrorEndpointID 的端点的版本信息。如果不提供此参数，则系统将列出所有已知 SnapMirror 端点的版本信息。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
ontapVersionInfo	ONTAP 端点的软件版本信息。	<i>ontapVersionInfo</i> 数组

发布以下版本后新增

10.1

GetSnapMirrorClusterIdentity

Element 软件 Web UI 可通过 `GetSnapMirrorClusterIdentity` 获取有关 ONTAP 集群的标识信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	如果提供此参数，则系统将列出具有指定 snapMirrorEndpointID 的端点的集群标识。如果不提供此参数，则系统将列出所有已知 SnapMirror 端点的集群标识。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorClusterIdentity	列出 SnapMirror 端点的集群标识。	<i>snapMirrorClusterIdentity</i> 数组

发布以下版本后新增

10.1

InitializeSnapMirrorRelationship

Element 软件 Web UI 可通过 `InitializeSnapMirrorRelationship` 方法在集群之间执行初始基线传输以初始化 SnapMirror 关系中的目标卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	远程 ONTAP 系统的 ID。	整型	无	是
destinationVolume	SnapMirror 关系中的目标卷。	snapMirrorVolumeInfo	无	是
maxTransferRate	指定卷之间的最大数据传输速率（以 KB/秒为单位）。默认值 0 表示无限制，允许 SnapMirror 关系充分利用可用的网络带宽。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	有关已初始化的 SnapMirror 关系的信息。	snapMirrorRelationship

发布以下版本后新增

10.1

ListSnapMirrorAggregates

Element 软件 Web UI 可通过 `ListSnapMirrorAggregates` 方法列出远程 ONTAP 系统上可用的所有 SnapMirror 聚合。聚合表示一组物理存储资源。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	仅返回与指定端点 ID 关联的聚合。如果不提供端点 ID，则系统将列出所有已知 SnapMirror 端点中的聚合。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorAggregates	列出 ONTAP 存储系统上的可用聚合。	snapMirrorAggregate 数组

发布以下版本后新增

10.1

ListSnapMirrorEndpoints

Element 软件 Web UI 可通过 `ListSnapMirrorEndpoints` 方法列出正在与 Element 存储集群通信的所有 SnapMirror 端点。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointIDs	仅返回与这些 ID 关联的对象。如果不提供 ID 或此数组为空，则此方法将返回所有 SnapMirror 端点 ID。	整型数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorEndpoints	列出现有 SnapMirror 端点。	snapMirrorEndpoint 数组

发布以下版本后新增

10.0

ListSnapMirrorLuns

Element 软件 Web UI 可通过 `ListSnapMirrorLuns` 方法列出远程 ONTAP 集群上的 SnapMirror 关系的 LUN 信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	仅列出与指定端点 ID 关联的 LUN 信息。	整型	无	是
destinationVolume	SnapMirror 关系中的目标卷。	snapMirrorVolumeInfo	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorLunInfos	包含有关 SnapMirror LUN 的信息的对象列表。	snapMirrorLunInfo 数组

发布以下版本后新增

10.1

ListSnapMirrorNetworkInterfaces

Element 软件 Web UI 可通过 `ListSnapMirrorNetworkInterfaces` 方法列出远程 ONTAP 系统上的所有可用 SnapMirror 接口。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	仅返回与指定端点 ID 关联的网络接口。如果不提供端点 ID，则系统将列出所有已知 SnapMirror 端点中的接口。	整型	无	否
interfaceRole	仅列出为指定角色提供服务的网络接口。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorNetworkInterfaces	列出 ONTAP 远程存储系统上的可用 SnapMirror 网络接口。	snapMirrorNetworkInterface 数组

发布以下版本后新增

10.1

ListSnapMirrorNodes

Element 软件 Web UI 可通过 `ListSnapMirrorNodes` 方法获取远程 ONTAP 集群中的节点列表。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	如果提供此参数，则系统将列出具有指定 snapMirrorEndpointID 的端点的节点。如果不提供此参数，则系统将列出所有已知 SnapMirror 端点的节点。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorNodes	列出 ONTAP 集群中的节点。	snapMirrorNode 数组

发布以下版本后新增

10.1

ListSnapMirrorPolicies

Element 软件 Web UI 可通过 `ListSnapMirrorPolicies` 方法列出远程 ONTAP 系统上的所有 SnapMirror 策略。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	仅列出与指定端点 ID 关联的策略。如果不提供端点 ID，则系统将列出所有已知 SnapMirror 端点中的策略。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorPolicies	列出 ONTAP 存储系统上的 SnapMirror 策略。	<i>snapMirrorPolicy</i> 数组

发布以下版本后新增

10.1

ListSnapMirrorSchedules

Element 软件 Web UI 可通过 `ListSnapMirrorSchedules` 方法获取远程 ONTAP 集群上可用的计划列表。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	如果提供此参数，则系统将列出具有指定 SnapMirror 端点 ID 的端点的计划。如果不提供此参数，则系统将列出所有已知 SnapMirror 端点的计划。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorSchedules	列出远程 ONTAP 集群上的 SnapMirror 计划。	snapMirrorJobScheduleCronInfo 数组

发布以下版本后新增

10.1

ListSnapMirrorRelationships

Element 软件 Web UI 可通过 `ListSnapMirrorRelationships` 方法列出 Element 存储集群中的一个或所有 SnapMirror 关系。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	仅列出与指定端点 ID 关联的关系。如果不提供端点 ID，则系统将列出所有已知 SnapMirror 端点中的关系。	整型	无	否
destinationVolume	列出与指定目标卷关联的关系。	snapMirrorVolumeInfo	无	否
sourceVolume	列出与指定源卷关联的关系。	snapMirrorVolumeInfo	无	否
vserver	列出指定 SVM 上的关系。	字符串	无	否
relationshipID	列出与指定关系 ID 关联的关系。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationships	包含有关 SnapMirror 关系的信息的对象列表。	snapMirrorRelationship 数组

发布以下版本后新增

10.1

ListSnapMirrorVolumes

Element 软件 Web UI 可通过 `ListSnapMirrorVolumes` 方法列出远程 ONTAP 系统上可用的所有 SnapMirror 卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	仅列出与指定端点 ID 关联的卷。如果不提供端点 ID，则系统将列出所有已知 SnapMirror 端点中的卷。	整型	无	否
vserver	列出指定 SVM 上托管的卷。SVM 的类型必须为“data”。	字符串	无	否
name	仅列出具有指定名称的 ONTAP 卷。	字符串	无	否
type	仅列出指定类型的 ONTAP 卷。可能值： <ul style="list-style-type: none"> rw: 读写卷 ls: 负载共享卷 dp: 数据保护卷 	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorVolumes	列出 ONTAP 存储系统上可用的 SnapMirror 卷。	snapMirrorVolume 数组

发布以下版本后新增

10.1

ListSnapMirrorVservers

Element 软件 Web UI 可通过 ListSnapMirrorVservers 方法列出远程 ONTAP 系统上可用的所有 SnapMirror SVM。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	仅列出与指定端点 ID 关联的 SVM。如果不提供端点 ID，则系统将列出所有已知 SnapMirror 端点中的 SVM。	整型	无	否

名称	说明	类型	默认值	必需
vserverType	仅列出指定类型的 SVM。可能值： <ul style="list-style-type: none"> admin data node system 	字符串	无	否
vserverName	仅列出具有指定名称的 SVM。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorVservers	列出 ONTAP 存储系统上可用的 SnapMirror SVM。	snapMirrorVserver 数组

发布以下版本后新增

10.1

ModifySnapMirrorEndpoint

Element 软件 Web UI 可通过 `ModifySnapMirrorEndpoint` 方法更改 SnapMirror 端点的名称和管理属性。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
SnapMirrorEndpointID	要修改的 SnapMirror 端点。	整型	无	是
managementIP	ONTAP 系统的新管理 IP 地址。	字符串	无	否
username	ONTAP 系统的新管理用户名。	字符串	无	否
password	ONTAP 系统的新管理密码。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorEndpoint	有关已修改的 SnapMirror 端点的信息。	snapMirrorEndpoint

发布以下版本后新增

10.0

ModifySnapMirrorEndpoint (非受管)

Element 软件可通过此版本的 `ModifySnapMirrorEndpoint` 方法修改非受管 `SnapMirror` 端点的存储集群名称或 IP 地址属性。不能使用 Element `SnapMirror` API 来管理非受管端点。必须使用 ONTAP 管理软件或 API 对其进行管理。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>SnapMirrorEndpointID</code>	要修改的 <code>SnapMirror</code> 端点。	整型	无	是
<code>clusterName</code>	端点的新名称。	字符串	无	否
<code>ipAddresses</code>	应与此 Element 存储集群通信的 ONTAP 存储系统集群的新 IP 地址列表。	字符串数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>snapMirrorEndpoint</code>	有关已修改的 <code>SnapMirror</code> 端点的信息。	<i><code>snapMirrorEndpoint</code></i>

发布以下版本后新增

10.3

ModifySnapMirrorRelationship

您可以使用 `ModifySnapMirrorRelationship` 更改执行计划快照的间隔。您还可以使用此方法删除或暂停计划。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>destinationVolume</code>	<code>SnapMirror</code> 关系中的目标卷。	<i><code>snapMirrorVolumeInfo</code></i>	无	是
<code>maxTransferRate</code>	指定卷之间的最大数据传输速率（以 KB/秒为单位）。默认值 0 表示无限制，允许 <code>SnapMirror</code> 关系充分利用可用的网络带宽。	整型	无	否
<code>policyName</code>	指定关系的 ONTAP <code>SnapMirror</code> 策略名称。	字符串	无	否

名称	说明	类型	默认值	必需
scheduleName	ONTAP 系统上用于更新 SnapMirror 关系的原有 cron 计划的名称。	字符串	无	否
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	包含已修改的 SnapMirror 关系属性的对象。	<i>snapMirrorRelationship</i>

发布以下版本后新增

10.1

UpdateSnapMirrorRelationship

Element 软件 Web UI 可通过 `UpdateSnapMirrorRelationship` 方法使 SnapMirror 关系中的目标卷成为源卷的最新镜像。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
destinationVolume	SnapMirror 关系中的目标卷。	<i>snapMirrorVolumeinfo</i>	无	是
maxTransferRate	指定卷之间的最大数据传输速率（以 KB/秒为单位）。默认值 0 表示无限制，允许 SnapMirror 关系充分利用可用的网络带宽。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	包含有关已更新的 SnapMirror 关系的信息的对象。	<i>snapMirrorRelationship</i>

发布以下版本后新增

10.1

QuiesceSnapMirrorRelationship

Element 软件 Web UI 可通过 `QuiesceSnapMirrorRelationship` 方法为 `SnapMirror` 关系禁用未来数据传输。如果正在进行传输，则此关系状态将变为“quiescing”，直到传输完成为止。如果当前传输被中止，它将不会重新启动。可以通过 `ResumeSnapMirrorRelationship` API 方法为此关系重新启用数据传输。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
destinationVolume	SnapMirror 关系中的目标卷。	snapMirrorVolumeinfo	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	包含有关已暂停的 SnapMirror 关系的信息的对象。	snapMirrorRelationship

发布以下版本后新增

10.1

ResumeSnapMirrorRelationship

Element 软件 Web UI 可通过 `ResumeSnapMirrorRelationship` 方法为已暂停的 `SnapMirror` 关系启用未来传输。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
destinationVolume	SnapMirror 关系中的目标卷。	snapMirrorVolumeinfo	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	包含有关已恢复的 SnapMirror 关系的信息的对象。	snapMirrorRelationship

发布以下版本后新增

10.1

ResyncSnapMirrorRelationship

Element 软件 Web UI 可通过 `ResyncSnapMirrorRelationship` 方法在源端点和目标端点之间建立或重新建立镜像关系。重新同步某个关系时，系统会删除目标卷上比通用 Snapshot 副本更新的快照，然后将目标卷挂载为数据保护卷，并将通用 Snapshot 副本作为导出的 Snapshot 副本。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapMirrorEndpointID	与 Element 存储集群通信的远程 ONTAP 存储系统的端点 ID。	整型	无	是
destinationVolume	SnapMirror 关系中的目标卷。	<i>snapMirrorVolumeInfo</i>	无	是
maxTransferRate	指定卷之间的最大数据传输速率（以 KB/秒为单位）。默认值 0 表示无限制，允许 SnapMirror 关系充分利用可用的网络带宽。	整型	无	否
sourceVolume	SnapMirror 关系中的源卷。	<i>snapMirrorVolumeInfo</i>	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapMirrorRelationship	包含有关已重新同步的 SnapMirror 关系的信息的对象。	<i>snapMirrorRelationship</i>

发布以下版本后新增

10.1

系统配置 API 方法

使用系统配置 API 方法可以获取和设置应用于集群内所有节点的配置值。

DisableBMCColdReset

您可以使用 `DisableBmcColdReset` 该方法禁用后台任务、该后台任务会定期为集群中的所有节点重置底板管理控制器（BMC）。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>cbcmResetDurationMinutes</code>	返回复位间隔之间的时间。命令完成后间隔应始终为 0。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "DisableBMCColdReset" ,
  "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": {
    "CBMCRestDurationMinutes" : 0
  }
}
```

发布以下版本后新增

12.0

DisableClusterSsh

您可以使用 `DisableClusterSsh` 方法为整个存储集群禁用 SSH 服务。在向此存储集群添加节点时，新节点将继承此集群范围的设置。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
result	一个 JSON 对象，其中包含存储集群的 SSH 服务状态、禁用 SSH 之前的剩余时间以及每个节点的 SSH 服务状态。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "DisableClusterSsh",
  "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,
    "timeRemaining": "00:43:21",
    "nodes": [
      {
        "nodeID": 1,
        "enabled": true
      },
      {
        "nodeID": 2,
        "enabled": true
      },
      {
        "nodeID": 3,
        "enabled": false
      },
      {
        "nodeID": 4,
        "enabled": false
      }
    ]
  }
}
```

发布以下版本后新增

10.3

DisableSnmp

您可以使用 `DisableSnmp` 方法在集群节点上禁用 SNMP。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

启用 BMCColdReset

您可以使用 `EnableBmcColdReset` 该方法启用后台任务、该后台任务会定期为集群中的所有节点重置底板管理控制器（BMC）。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
timeout	BMC 重置操作之间的时间（以分钟为单位）。	整型	20160 分钟	否

返回值

此方法具有以下返回值：

名称	说明	类型
cbcmResetDurationMinutes	返回复位间隔之间的时间。命令完成后间隔应始终为 0。	整型

请求示例

此方法的请求类似于以下示例：

```
{  
  "方法" : "EnableBMCColdReset",  
  "params": {  
    "超时" : 36000  
  },  
  "id": 1  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "CBMCResetDurationMinutes" : 36000
  }
}
```

发布以下版本后新增

12.0

EnableClusterSsh

您可以使用 `EnableClusterSsh` 方法在存储集群中的所有节点上启用 SSH 服务。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
duration	SSH 服务保持启用状态的时间量。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
result	一个 JSON 对象，其中包含存储集群的 SSH 服务状态、禁用 SSH 之前的剩余时间以及每个节点的 SSH 服务状态。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "EnableClusterSsh",
  "params": {
    "duration" : "02:00:00.00"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "enabled": true,
    "timeRemaining": "00:43:21",
    "nodes": [
      {
        "nodeID": 1,
```

```

        "enabled": true
    },
    {
        "nodeID": 2,
        "enabled": true
    },
    {
        "nodeID": 3,
        "enabled": false
    },
    {
        "nodeID": 4,
        "enabled": false
    } ]
    }
}

```

发布以下版本后新增

10.3

EnableSnmpp

您可以使用 `EnableSnmpp` 方法在集群节点上启用 **SNMP**。在启用 **SNMP** 时，此操作将应用于集群中的所有节点，而且所传递的值将整体替换在先前任何 `EnableSnmpp` 调用中设置的所有值。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snmpV3Enabled	如果设置为 <code>true</code> ，将会在集群中的每个节点上启用 SNMP v3 。如果设置为 <code>false</code> ，则会启用 SNMP v2 。	布尔型	<code>false</code>	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": "EnableSnmpp",
  "params": {
    "snmpV3Enabled" : "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" : {}
}

```

发布以下版本后新增

9.6

GetBinAssignmentProperties

可以使用GetBinAssignmentProperties方法检索数据库中的容器分配属性。

参数

此方法具有无输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
属性	详细说明数据库中所有当前容器分配的属性。	BinAssignmentProperties 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "GetBinAssignmentProperties" ,
  "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": {
    "属性" : {
      "算法运行时间" : 1105 、
      "ArerplicasValid" : true 、
      "二进制计数" : 65536 、
      "IsBalanced" : 正确、
      "稳定" : 正确、
      "Iswell耦合" : 错误、
      "布局" : [
        {
          "保护域名" : "1" 、
          "services": [
            {
              "预算" : 7281 、
              "ServiceID" : 16.
            },
            {
              "预算" : 7281 、
              "ServiceID" : 19.
            },
            {
              "预算" : 7281 、
            }
          ]
        }
      ]
    }
  }
}
```

```
        "ServiceID" : 24
      }
    ]
  },
  {
    "保护域名 ":"2" 、
    "services": [
      {
        "预算 " : 7281 、
        "ServiceID" : 17
      },
      {
        "预算 " : 7281 、
        "ServiceID" : 20
      },
      {
        "预算 " : 7281 、
        "ServiceID" : 22.
      }
    ]
  },
  {
    "保护域名 ":"3" 、
    "services": [
      {
        "预算 " : 7281 、
        "ServiceID" : 18
      },
      {
        "预算 " : 7281 、
        "ServiceID" : 21.
      },
      {
        "预算 " : 7281 、
        "ServiceID" : 23
      }
    ]
  }
],
"数字交换 " : 0 、
"NumUpdateingbins" : 0 、
"protectionDomainType": "node",
"原因 " : " 最终 " 、
"复制计数 " : 2 、
"请求重新平衡 " : 错误、
"服务转换容量 " : [] 、
"发布时间 " : "2020-04-02T18:34:07.807681Z" 、
"ValidSchemes" : []
}
}
}
```

发布以下版本后新增

12.0

GetClusterSshInfo

您可以使用 `GetClusterSshInfo` 方法查询整个存储集群的 SSH 服务状态。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
result	一个 JSON 对象，其中包含存储集群的 SSH 服务状态、禁用 SSH 之前的剩余时间以及每个节点的 SSH 服务状态。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetClusterSshInfo",
  "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",
    "timeRemaining" : "00:43:21",
    "nodes": [
      {
        "nodeID": 1,
        "enabled": true
      },
      {
        "nodeID": 2,
        "enabled": true
      },
      {
        "nodeID": 3,
        "enabled": false
      },
      {
        "nodeID": 4,
        "enabled": false
      }
    ]
  }
}
```

发布以下版本后新增

10.3

GetClusterStructure

您可以使用 `GetClusterStructure` 方法备份当前存储集群配置信息。如果在运行此方法时更改了存储集群配置，则此配置备份的内容将不可预测。您可以将此数据保存到一个文本文件中，然后在发生灾难时，在其他集群上或同一集群上还原此数据。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
result	一个 JSON 对象，其中包含当前存储集群配置信息。	<i>clusterStructure</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetClusterStructure",
  "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" : <clusterStructure object containing configuration information>
}
```

发布以下版本后新增

10.3

GetFipsReport

您可以使用 `GetFipsReport` 方法查看存储集群中所有节点的 FIPS 140-2 加密功能支持状态。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
result	一个 JSON 对象，其中包含每个节点的 FIPS 140-2 功能支持状态，以及未响应查询的每个节点的错误信息。	<i>fipsReport</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetFipsReport",
  "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,
        "fipsDrives": "None",
        "httpsEnabled": true
      },
      {
        "nodeID": 3,
        "fipsDrives": "None",
        "httpsEnabled": true
      }
    ],
    "errorNodes": [
      {
        "nodeID": 2,
        "error": {
          "message": "The RPC timed out.",
          "name": "xRpcTimeout"
        }
      }
    ]
  }
}
```

发布以下版本后新增

10.3

GetLldpConfig

您可以使用 `GetLldpConfig` 方法获取存储集群中每个节点的链路层发现协议 (Link Layer Discovery Protocol, LLDP) 配置。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
lldpConfig	有关存储集群 LLDP 配置的信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "GetLldpConfig",  
  "id" : 1  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{  
  "id": null,  
  "result": {  
    "lldpConfig": {  
      "enableLldp": true,  
      "enableMed": false,  
      "enableOtherProtocols": true  
    }  
  }  
}
```

GetLldpInfo

您可以使用 `GetLldpInfo` 方法获取存储集群中每个节点或单个存储节点的链路层发现协议 (Link Layer Discovery Protocol, LLDP) 配置。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
lldpInfo	有关存储集群中每个节点的机箱、接口和相邻 LLDP 设置的信息。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "GetLldpInfo",  
  "id" : 1  
}
```

响应示例

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

11.0

相关参考

[GetLldpInfo](#) on page 509

GetLldpInfo 方法返回类似于以下示例的响应。

GetNodeFipsDrivesReport

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

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
fipsDrives	一个 JSON 对象，包含此节点的 FIPS 140-2 功能支持状态。可能值： <ul style="list-style-type: none">• None：节点不支持 FIPS。• Partial：节点支持 FIPS，但此节点中并非所有驱动器都为 FIPS 驱动器。• Ready：节点支持 FIPS，并且此节点中的所有驱动器均为 FIPS 驱动器（或者不存在任何驱动器）。	字符串

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

11.5

GetNtpInfo

您可以使用 GetNtpInfo 方法获取当前网络时间协议 (Network Time Protocol, NTP) 配置信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
servers	列出 NTP 服务器。	字符串数组
broadcastclient	指出集群中的节点是否正在侦听广播 NTP 消息。可能值： <ul style="list-style-type: none"> • true • false 	布尔型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetNtpInfo",
  "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" : {
    "broadcastclient" : false,
    "servers" : [ "us.pool.ntp.org" ]
  }
}
```

发布以下版本后新增

9.6

GetNvramInfo

您可以使用 GetNvramInfo 方法从每个节点获取 NVRAM 卡的相关信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
force	要在集群中的所有节点上成功运行，此方法中必须包含 force 参数。	布尔型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
nvramInfo	在 NVRAM 卡上检测到的事件和错误的数组。	JSON 对象

请求示例

此方法的请求类似于以下示例：

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

响应示例

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

相关参考

[GetNvramInfo](#) on page 542

GetNvramInfo 方法返回类似于以下示例的响应。

GetProtectionDomainLayout

您可以使用GetProtectionDomainLayout该方法返回集群的所有保护域信息、包括每个节点所在的机箱和自定义保护域。

参数

此方法具有无输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
域保护布局	节点列表、每个节点及其关联的保护域。	NodeProtectionDomains 对象的 JSON 列表。

请求示例

此方法的请求类似于以下示例：

```
{
  "方法" : "GetProtectionDomainLayout",
  "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": {
    "ProtectionDomainLayout" : [
      {
```

```
    "nodeID": 1,
    "ProtectionDomains": [
      {
        "保护域名": "Qtfcr2914008d"、
        "ProtectionDomainType": "Chassis"
      },
      {
        "保护域名": "机架 1"、
        "ProtectionDomainType": "Custom"
      }
    ]
  },
  {
    "nodeID": 2,
    "ProtectionDomains": [
      {
        "保护域名": "Qtfcr291500ea"、
        "ProtectionDomainType": "Chassis"
      },
      {
        "保护域名": "机架 1"、
        "ProtectionDomainType": "Custom"
      }
    ]
  },
  {
    "nodeID": 3,
    "ProtectionDomains": [
      {
        "保护域名": "Qtfcr291500 c3"、
        "ProtectionDomainType": "Chassis"
      },
      {
        "保护域名": "rack-2"、
        "ProtectionDomainType": "Custom"
      }
    ]
  },
  {
    "nodeID": 4,
    "ProtectionDomains": [
      {
        "保护域名": "Qtfcr291400e6"、
        "ProtectionDomainType": "Chassis"
      },
      {
        "保护域名": "rack-2"、
        "ProtectionDomainType": "Custom"
      }
    ]
  }
]
```

发布以下版本后新增

12.0

GetRemoteLoggingHosts

您可以使用 `GetRemoteLoggingHosts` 方法获取日志服务器的当前列表。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>remoteHosts</code>	有关配置为接收转发的日志记录信息的主机的 IP 地址列表和端口信息。	<i>loggingServer</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 3386609,
  "method": "GetRemoteLoggingHosts",
  "params": {}
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 3386609,
  "result": {
    "remoteHosts": [
      {
        "host": "172.16.1.20",
        "port": 10514
      },
      {
        "host": "172.16.1.25"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

相关参考

[SetRemoteLoggingHosts](#) on page 339

您可以使用 `SetRemoteLoggingHosts` 方法配置从存储集群节点到集中式日志服务器的远程日志记录。远程日志记录是借助默认端口 514 通过 TCP 执行的。此 API 并不添加到现有日志记录主机中，而是将现有值替换为此 API 方法所指定的新值。您可以使用 `GetRemoteLoggingHosts` 确定当前的日志记录主机，然后使用 `SetRemoteLoggingHosts` 设置所需的当前和新日志记录主机列表。

GetSnmPACL

您可以使用 `GetSnmPACL` 方法获取对集群节点的当前 SNMP 访问权限。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
networks	列出网络及其对正在集群节点上运行的 SNMP 服务器所具有的访问类型。如果禁用 SNMP v3，则显示此值。	<i>network</i> 数组
usmUsers	列出用户及其对正在集群节点上运行的 SNMP 服务器所具有的访问类型。如果启用 SNMP v3，则显示此值。	<i>usmUser</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "GetSnmPACL",  
  "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" : {  
    "usmUsers" : [  
      {  
        "name": "jdoe",  
        "access": "rouser",  
        "secLevel": "priv",  
        "password": "mypassword",  
        "passphrase": "mypassphrase",  
      }  
    ]  
  }  
}
```

发布以下版本后新增

9.6

GetSnmplInfo

您可以使用 `GetSnmplInfo` 方法获取当前简单网络管理协议 (Simple Network Management Protocol, SNMP) 配置信息。

参数

注: `GetSnmplInfo` 对于高于 Element 8.0 的版本已弃用。 `GetSnmplState` 和 `SetSnmplACL` 方法将取代 `GetSnmplInfo` 方法。

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
networks	列出网络和为 SNMP 启用的访问类型。 注: <code>networks</code> 仅当已禁用 SNMP v3 时才会显示。	<i>network</i>
enabled	指示集群中的节点是否针对 SNMP 进行配置。可能值： <ul style="list-style-type: none">• true• false	布尔型
snmpV3Enabled	如果集群中的节点针对 SNMP v3 进行配置。可能值： <ul style="list-style-type: none">• true• false	布尔型
usmUsers	如果启用 SNMP v3，将从集群返回 SNMP 的用户访问参数列表，而非 <code>networks</code> 参数。	<i>usmUser</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetSnmplInfo",
  "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,
    "networks" : [
      {
        "access" : "rosys",
        "cidr" : 0,

```

```
        "community" : "public",  
        "network" : "localhost"  
    }  
  ]  
}  
}
```

发布以下版本后新增

9.6

相关参考

[GetSnmState](#) on page 327

您可以使用 `GetSnmState` 方法获取 SNMP 功能的当前状态。

[SetSnmACL](#) on page 340

您可以使用 `SetSnmACL` 方法配置对集群节点的 SNMP 访问权限。使用此接口设置的值将应用于集群中的所有节点，而且所传递的值将替换在先前任何 `SetSnmACL` 调用中设置的所有值。另请注意，使用此接口设置的值将替换使用 `SetSnmInfo` 设置的所有 `network` 或 `usmUsers` 值。

GetSnmState

您可以使用 `GetSnmState` 方法获取 SNMP 功能的当前状态。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
enabled	可能值： <ul style="list-style-type: none">• true• false 默认值为 false。如果集群中的节点针对 SNMP 进行配置则返回 true。	布尔型
snmpV3Enabled	可能值： <ul style="list-style-type: none">• true• false 默认值为 false。如果集群中的节点针对 SNMP v3 进行配置则返回 true。	布尔型

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "GetSnmState",  
  "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,
    "snmpV3Enabled": false
  }
}
```

发布以下版本后新增

9.6

相关参考

[SetSnmpACL](#) on page 340

您可以使用 `SetSnmpACL` 方法配置对集群节点的 SNMP 访问权限。使用此接口设置的值将应用于集群中的所有节点，而且所传递的值将替换在先前任何 `SetSnmpACL` 调用中设置的所有值。另请注意，使用此接口设置的值将替换使用 `SetSnmpInfo` 设置的所有 `network` 或 `usmUsers` 值。

GetSnmpTrapInfo

您可以使用 `GetSnmpTrapInfo` 方法获取当前 SNMP 陷阱配置信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
trapRecipients	要接收集群生成的陷阱的主机列表。	<i>snmpTrapRecipient</i> 数组
clusterFaultTrapsEnabled	值 <code>true</code> 表示 <code>solidFireClusterFaultNotification</code> 配置为在记录集群故障后发送给一系列陷阱接收方。	布尔型
clusterFaultResolvedTrapsEnabled	值 <code>true</code> 表示 <code>solidFireClusterFaultResolvedNotification</code> 配置为在解决集群故障后发送给一系列陷阱接收方。	布尔型
clusterEventTrapsEnabled	值 <code>true</code> 表示 <code>solidFireClusterEventNotification</code> 配置为在记录集群事件后发送给一系列陷阱接收方。	布尔型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetSnmpTrapInfo"
  "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": {  
    "clusterEventTrapsEnabled": true,  
    "clusterFaultResolvedTrapsEnabled": true,  
    "clusterFaultTrapsEnabled": true,  
    "trapRecipients": [  
      {  
        "community": "public",  
        "host": "192.168.151.60",  
        "port": 162  
      },  
      {  
        "community": "solidfireAlerts",  
        "host": "NetworkMonitor",  
        "port": 162  
      },  
      {  
        "community": "wakeup",  
        "host": "PhoneHomeAlerter",  
        "port": 1008  
      }  
    ]  
  }  
}
```

发布以下版本后新增

9.6

GetSSLCertificate

您可以使用 `GetSSLCertificate` 方法检索集群存储节点上当前处于活动状态的 SSL 证书。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值:

名称	说明	类型
certificate	PEM 编码的完整证书文本。	字符串
details	证书的解码信息。	JSON 对象

请求示例

此方法的请求类似于以下示例:

```
{  
  "method" : "GetSSLCertificate",  
  "params" : {},  
}
```


ListProtectionDomainLevels

您可以使用 `ListProtectionDomainLevels` 方法列出存储集群的容错和故障恢复能力级别。容错级别表示集群在发生故障时可以继续读取和写入数据的能力，而故障恢复能力级别则表示存储集群自动从一个或多个故障中自行修复的能力。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>protectionDomainLevels</code>	不同保护域级别的列表，其中每个级别均会提供存储集群的容错和故障恢复能力信息。	<i>protectionDomainLevel</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListProtectionDomainLevels",
  "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": {
    "protectionDomainLevels": [
      {
        "protectionDomainType": "node",
        "resiliency": {
          "protectionSchemeResiliencies": [
            {
              "protectionScheme": "doubleHelix",
              "sustainableFailuresForBlockData": 0,
              "sustainableFailuresForMetadata": 1
            }
          ],
          "singleFailureThresholdBytesForBlockData": 0,
          "sustainableFailuresForEnsemble": 1
        },
        "tolerance": {
          "protectionSchemeTolerances": [
            {
              "protectionScheme": "doubleHelix",
              "sustainableFailuresForBlockData": 0,
              "sustainableFailuresForMetadata": 1
            }
          ],
          "sustainableFailuresForEnsemble": 1
        }
      }
    ]
  }
},
```

```
{
  "protectionDomainType": "chassis",
  "resiliency": {
    "protectionSchemeResiliencies": [
      {
        "protectionScheme": "doubleHelix",
        "sustainableFailuresForBlockData": 0,
        "sustainableFailuresForMetadata": 1
      }
    ],
    "singleFailureThresholdBytesForBlockData": 0,
    "sustainableFailuresForEnsemble": 1
  },
  "tolerance": {
    "protectionSchemeTolerances": [
      {
        "protectionScheme": "doubleHelix",
        "sustainableFailuresForBlockData": 0,
        "sustainableFailuresForMetadata": 1
      }
    ],
    "sustainableFailuresForEnsemble": 1
  }
}
]
```

发布以下版本后新增

11.0

RemoveSSLCertificate

您可以使用 `RemoveSSLCertificate` 方法为集群中的存储节点删除用户 SSL 证书和专用密钥。删除此证书和专用密钥后，存储节点将配置为使用默认证书和专用密钥。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method" : "RemoveSSLCertificate",
  "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" : {}  
  }
```

发布以下版本后新增

10.0

ResetNetworkConfig

您可以使用 `ResetNetworkConfig` 方法帮助解决单个节点的网络配置问题。此方法会将单个节点的网络配置重置为出厂默认设置。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法不会返回响应。

发布以下版本后新增

11.0

ResetSupplementalTlsCiphers

您可以使用 `ResetSupplementalTlsCiphers` 方法将补充 TLS 密码列表还原为默认值。可对整个集群使用此方法。

参数

此方法没有任何输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "ResetSupplementalTlsCiphers",  
  "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" : {}  
}
```

发布以下版本后新增

11.3

SetClusterStructure

您可以使用 `SetClusterStructure` 方法从备份中还原存储集群配置信息。调用此方法时，您可以通过 `params` 参数传递包含要还原的配置信息的 `clusterStructure` 对象。

参数

此方法具有以下输入参数：

名称	说明	类型
<code>params</code>	一个 JSON 对象，其中包含当前存储集群配置信息。	<i>clusterStructure</i>

返回值

此方法具有以下返回值：

名称	说明	类型
<code>result</code>	异步结果句柄。	<code>asyncHandle</code>

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "SetClusterStructure",  
  "params": <insert clusterStructure object here>,  
  "id" : 1  
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

10.3

SetLldpConfig

您可以使用 `SetLldpConfig` 方法为存储集群配置链路层发现协议 (Link Layer Discovery Protocol, LLDP) 设置。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>enableOtherProtocols</code>	自动使用其他发现协议 — CDP、FDP、EDP 和 SONMP。	布尔型	<code>true</code>	否
<code>enableMed</code>	启用介质端点发现 (LLDP-MED)。	布尔型	<code>false</code>	否
<code>enableLldp</code>	启用或禁用 LLDP。	布尔型	<code>true</code>	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>lldpConfig</code>	有关当前存储集群 LLDP 配置的信息，包括新更改的设置。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 3920,
  "method": "SetLldpConfig",
  "params": {
    "lldpConfig": {
      "enableMed": true
    }
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 3920,
  "result": {
    "lldpConfig": {
      "enableLldp": true,
      "enableMed": true,
      "enableOtherProtocols": true
    }
  }
}
```

SetNtpInfo

您可以使用 `SetNtpInfo` 方法在集群节点上配置 NTP。使用此接口设置的值将应用于集群中的所有节点。如果 NTP 广播服务器定期在网络上广播时间信息，则可以选择将节点配置为广播客户端。

参数

注：请确保使用网络内部的 NTP 服务器，而不是使用安装默认值。

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
servers	要添加到每个节点 NTP 配置中的 NTP 服务器列表。	字符串数组	无	是
broadcastclient	使集群中的每个节点成为广播客户端。	布尔型	false	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": "SetNtpInfo",
  "params": {
    "servers" : [
      "ntpserver1.example.org",
      "ntpserver2.example.org",
      "ntpserver3.example.org"
    ],
    "broadcastclient" : 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" : {}
}
```

发布以下版本后新增

9.6

SetProtectionDomainLayout

您可以使用 `SetProtectionDomainLayout` 该方法将节点分配给自定义保护域。

必须为集群中的所有活动节点提供信息、并且不能为非活动节点提供任何信息。必须将给定机箱中的所有节点分配给同一个自定义保护域。必须为所有节点提供相同的

protectionDomainType。不应包括非自定义域保护类型、例如节点和机箱。如果提供了其中一个，则会忽略自定义保护域并返回相应的错误。

注：以下配置不支持自定义保护域：

- 包含共享机箱的存储集群
- 双节点存储集群

该方法在使用这些配置的存储集群上使用时返回错误。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
protectionDomainLayout	每个节点的保护域信息。	<i>NodeProtectionDomains</i> 对象的 JSON 列表。	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
域保护布局	节点列表、每个节点及其关联的保护域。	<i>NodeProtectionDomains</i> 对象的 JSON 列表。

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 1,
  "方法": "SetProtectionDomainLayout",
  "params": {
    "ProtectionDomainLayout": [
      {
        "nodeID": 1,
        "ProtectionDomains": [
          {
            "保护域名": "机架 1",
            "ProtectionDomainType": "Custom"
          }
        ]
      }
    ],
    {
      "nodeID": 2,
      "ProtectionDomains": [
        {
          "保护域名": "机架 1",
          "ProtectionDomainType": "Custom"
        }
      ]
    },
    {
      "nodeID": 3,
      "ProtectionDomains": [
```

```
{
  {
    "保护域名" : "rack-2"、
    "ProtectionDomainType" : "Custom"
  }
],
},
{
  "nodeID": 4,
  "ProtectionDomains" : [
    {
      "保护域名" : "rack-2"、
      "ProtectionDomainType" : "Custom"
    }
  ]
}
]
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{
  "id": 1,
  "result": {
    "ProtectionDomainLayout" : [
      {
        "nodeID": 1,
        "ProtectionDomains" : [
          {
            "保护域名" : "Qtfcr2914008d"、
            "ProtectionDomainType" : "Chassis"
          },
          {
            "保护域名" : "机架 1"、
            "ProtectionDomainType" : "Custom"
          }
        ]
      },
      {
        "nodeID": 2,
        "ProtectionDomains" : [
          {
            "保护域名" : "Qtfcr291500ea"、
            "ProtectionDomainType" : "Chassis"
          },
          {
            "保护域名" : "机架 1"、
            "ProtectionDomainType" : "Custom"
          }
        ]
      },
      {
        "nodeID": 3,
        "ProtectionDomains" : [
          {
            "保护域名" : "Qtfcr291500 c3"、
            "ProtectionDomainType" : "Chassis"
          }
        ]
      }
    ]
  }
}
```



```

        },
        {
          "host": "172.16.1.25"
        }
      ]
    }
  }
}

```

响应示例

此方法返回类似于以下示例的响应：

```

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

```

发布以下版本后新增

9.6

相关参考

[GetRemoteLoggingHosts](#) on page 324

您可以使用 `GetRemoteLoggingHosts` 方法获取日志服务器的当前列表。

SetSnmpACL

您可以使用 `SetSnmpACL` 方法配置对集群节点的 SNMP 访问权限。使用此接口设置的值将应用于集群中的所有节点，而且所传递的值将替换在先前任何 `SetSnmpACL` 调用中设置的所有值。另请注意，使用此接口设置的值将替换使用 `SetSnmpInfo` 设置的所有 `network` 或 `usmUsers` 值。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>networks</code>	列出网络及其对正在集群节点上运行的 SNMP 服务器所具有的访问类型。有关可能的 <code>networks</code> 值，请参见 SNMP <code>network</code> 对象。如果禁用 SNMP v3，则需要此参数。	<i>network</i>	无	否
<code>usmUsers</code>	列出用户及其对正在集群节点上运行的 SNMP 服务器所具有的访问类型。如果启用 SNMP v3，则需要此参数。	<i>usmUser</i>	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "SetSnmpACL",
  "params": {

```

```

    "usmUsers" : [
      {
        "name": "jdoe",
        "access": "rouser",
        "secLevel": "priv",
        "password": "mypassword",
        "passphrase": "mypassphrase",
      }
    ]
  },
  "id" : 1
}

```

响应示例

此方法返回类似于以下示例的响应:

```

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

```

发布以下版本后新增

9.6

相关参考

[SetSnmplInfo](#) on page 341

您可以使用 `SetSnmplInfo` 方法在集群节点上配置 SNMP 版本 2 和版本 3。使用此接口设置的值将应用于集群中的所有节点，而且所传递的值将替换在先前任意 `SetSnmplInfo` 调用中设置的所有值。

SetSnmplInfo

您可以使用 `SetSnmplInfo` 方法在集群节点上配置 SNMP 版本 2 和版本 3。使用此接口设置的值将应用于集群中的所有节点，而且所传递的值将替换在先前任意 `SetSnmplInfo` 调用中设置的所有值。

参数

注: `SetSnmplInfo` 对于 Element 6.0 及更高版本已弃用。请改用 [EnableSnmpl](#) 和 [SetSnmplACL](#) 方法。

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
networks	列出网络及其对正在集群节点上运行的 SNMP 服务器所具有的访问类型。有关可能的值，请参见 SNMP network 对象。仅 SNMP v2 需要此参数。	network 数组	无	否
enabled	如果设置为 <code>true</code> ，则系统将会在集群中的每个节点上启用 SNMP。	布尔型	<code>false</code>	否

名称	说明	类型	默认值	必需
snmpV3Enabled	如果设置为 true, 则系统将会在集群中的每个节点上启用 SNMP v3。	布尔型	false	否
usmUsers	如果启用了 SNMP v3, 则必须传递此值, 而非 networks 参数。仅 SNMP v3 需要此参数。	<i>usmUser</i>	无	否

返回值

此方法没有返回值。

已启用 **SNMP v3** 的请求示例

此方法的请求类似于以下示例:

```
{
  "method": "SetSnmpInfo",
  "params": {
    "enabled": true,
    "snmpV3Enabled": true,
    "usmUsers": [
      {
        "name": "user1",
        "access": "rouser",
        "secLevel": "auth",
        "password": "namex1",
        "passphrase": "yourpassphrase"
      }
    ]
  },
  "id": 1
}
```

已启用 **SNMP v2** 的请求示例

此方法的请求类似于以下示例:

```
{
  "method": "SetSnmpInfo",
  "params": {
    "enabled": true,
    "snmpV3Enabled": false,
    "networks": [
      {
        "community": "public",
        "access": "ro",
        "network": "localhost",
      }
    ]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

SetSnmptTrapInfo

您可以使用 `SetSnmptTrapInfo` 方法启用和禁用集群 SNMP 通知（陷阱）的生成以及指定一组接收通知的主机。每次调用 `SetSnmptTrapInfo` 方法时传递的值都将替换先前调用中设置的所有值。

参数

此方法具有以下输入参数：

名称	说明	类型
trapRecipients	要接收存储集群生成的陷阱的主机列表。如果启用了任何一个陷阱类型，则至少需要一个对象。仅当任何布尔型参数设置为 <code>true</code> 时，才需要此参数。（无默认值。非必需。）	<i>snmpTrapRecipient</i> 数组
clusterFaultTrapsEnabled	如果设置为 <code>true</code> ，则在记录集群故障后，系统会向所配置的一系列陷阱接收方发送相应的集群故障通知。（默认值： <code>false</code> 。非必需。）	布尔型
clusterFaultResolvedTrapsEnabled	如果设置为 <code>true</code> ，则在解决集群故障后，系统会向所配置的一系列陷阱接收方发送相应的集群故障已解决通知。（默认值： <code>false</code> 。非必需。）	布尔型
clusterEventTrapsEnabled	如果设置为 <code>true</code> ，则在记录集群事件后，系统会向所配置的一系列陷阱接收方发送相应的集群事件通知。（默认值： <code>false</code> 。非必需。）	布尔型

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": "SetSnmptTrapInfo",
"params": {
  "clusterFaultTrapsEnabled": true,
  "clusterFaultResolvedTrapsEnabled": true,
  "clusterEventTrapsEnabled": true,
}
```

```

    "trapRecipients":[
    {
      "host":"192.30.0.10",
      "port":162,
      "community":"public"
    }
  ],
  "id":1
}

```

响应示例

此方法返回类似于以下示例的响应:

```

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

```

发布以下版本后新增

9.6

SetSSLCertificate

您可以使用 SetSSLCertificate 方法为集群中的存储节点设置用户 SSL 证书和专用密钥。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
certificate	PEM 编码的证书文本版本。	字符串	无	是
privateKey	PEM 编码的专用密钥文本。	字符串	无	是

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" : "SetSSLCertificate",
  "params" : {
    "privateKey": "-----BEGIN RSA PRIVATE KEY-----\nMIIIEowIBAAKCAQE8U
+28fnLQNWEMR6akeDKuehSpS79odLGigI18qlCV/AUY5\nzLjqsTjBvTJVRv44yoCTgNrx36U7FHP4t6P/
Si0aYr4ovxl5wDpEM3Qyy5JPB7Je\nlOB6AD7fmiTweP20HRYpZvY
+Uz7LYEFCmrpgGZQF3iOSIcBHTLKE5186JVT6j5dg\n6yJUGQO352y1c9HXHcn61b/
jy10DmVNUZ0caQwAmIS3Jmoyx+zj/Ya4WKq+2SqTA\nX7bX0F3wHHfXnZlHnM8fET5N/9A
+K6lS7dg9cyXu4afXcgKy14JiNBvqbBjhgJtE\n76yAy6rTHu0xM3jjdkcb9Y8miNzxF+ACq
+itawIDAQABAoIBAH1j1IZr6/sltqVW\n00qVC/49dyNu
+KWVSq92ti9rFe7hBPueh9gklh78hP9QlitLkir3YK4GFsTFUMux\n7z1NRCxA/
4LrmLSkAjjW2kRXDFv12bwZq0ua9NefGw9208D2OZvbuOxk7Put2p6se
\nfgNzSjf2SI5DIX3Ume5dDN5FByu52CJ9mI4U16ngbWln2wc4nsxJg0aAEkzB7wnq\nt+Am5/

```


SnmpSendTestTraps

SnmpSendTestTraps 可用于测试集群的 SNMP 功能。此方法可指示集群将测试 SNMP 陷阱发送至当前配置的 SNMP 管理器。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
status	测试的状态。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "SnmpSendTestTraps",
  "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": {
    "status": "complete"
  }
}
```

发布以下版本后新增

9.6

TestAddressAvailability

您可以使用 TestAddressAvailability 方法检查某个 IP 地址是否正被存储集群中的某个接口使用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
interface	目标网络接口（例如 eth0、绑定 10G 等）。	字符串	无	是
address	目标接口上要扫描的 IP 地址。	字符串	无	是
virtualNetworkTag	目标 VLAN ID。	整型	无	否
timeout	测试目标地址的超时值（以秒为单位）。	整型	5	否

返回值

此方法具有以下返回值：

名称	说明	类型
address	已测试的 IP 地址。	字符串
available	如果请求的 IP 地址正在使用中，则为 True；否则为 False。	布尔型

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "TestAddressAvailability",  
  "params": {  
    "interface": "Bond10G",  
    "address": "10.0.0.1",  
    "virtualNetworkTag": 1234  
  }  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{  
  "id": 1,  
  "result": {  
    "address": "10.0.0.1",  
    "available": true  
  }  
}
```

发布以下版本后新增

11.0

多租户网络 API 方法

通过 Element 存储集群中的多租户网络，可以在没有第 3 层路由的情况下使不同逻辑网络中多个客户端之间的流量连接到一个 Element 存储集群。

使用 VLAN 标记可以在网络堆栈中隔离与存储集群的连接。

设置多租户虚拟网络的前提条件

- 您必须已确定要分配给存储节点上虚拟网络的客户端网络 IP 地址块。
- 您必须已确定要用作所有存储通信的端点的客户端存储网络 IP (SVIP) 地址。

虚拟网络操作顺序

1. 使用 `AddVirtualNetwork` 方法批量配置输入的 IP 地址。
添加虚拟网络后，集群将自动执行以下步骤：
 - 每个存储节点创建一个虚拟网络接口。
 - 为每个存储节点分配一个可使用虚拟 SVIP 路由到的 VLAN 地址。
 - 节点重新启动时，VLAN IP 地址持久保留在每个节点上。
2. 分配了虚拟网络接口和 VLAN 地址后，您可以为虚拟 SVIP 分配客户端网络流量。

相关概念

[虚拟网络命名约定](#) on page 348

NetApp Element 存储系统会使用单调递增的数字作为系统中所有对象的唯一标识符。

虚拟网络命名约定

NetApp Element 存储系统会使用单调递增的数字作为系统中所有对象的唯一标识符。

创建新卷时，新的卷 ID 将以 1 为增量。此约定适用于运行 Element 软件的存储集群中的虚拟网络。您在 Element 集群中创建的第一个虚拟网络的 `VirtualNetworkID` 为 1。此 ID 与 VLAN 标记编号不同。

您可以如 API 方法中所述，将 `VirtualNetworkID` 和 `VirtualNetworkTag`（VLAN 标记）互换使用。

AddVirtualNetwork

您可以使用 `AddVirtualNetwork` 方法将新的虚拟网络添加到集群配置中。

添加虚拟网络时，系统将为每个节点创建一个接口，且每个接口都需要一个虚拟网络 IP 地址。在此 API 方法的参数中指定的 IP 地址数量必须大于或等于集群中的节点数。系统将批量配置虚拟网络地址，并自动将其分配给各个节点。您无需手动将虚拟网络地址分配给节点。

注：`AddVirtualNetwork` 方法仅用于创建新的虚拟网络。如果要更改现有虚拟网络，请使用 [ModifyVirtualNetwork](#) 方法。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
addressBlocks	要包含在虚拟网络中的唯一 IP 地址范围。此对象所需的成员： <ul style="list-style-type: none"> start: IP 地址范围的起点。（字符串） size: 要包含在块中的 IP 地址数。（整型） 	JSON 对象数组	无	是
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
gateway	虚拟网络的网关的 IP 地址。只有当 namespace 参数设置为 true 时，此参数才有效。	字符串	无	否
name	用户为新虚拟网络定义的名称。	字符串	无	是
namespace	设置为 true 时，通过创建并配置命名空间及其所包含的虚拟网络，可启用可路由存储 VLAN 功能。	布尔型	无	否
netmask	所创建虚拟网络的唯一网络掩码。	字符串	无	是
svip	所创建虚拟网络的唯一存储 IP 地址。	字符串	无	是
virtualNetworkTag	唯一虚拟网络 (VLAN) 标记。支持的值为 1 到 4094。	整型	无	是

注: 如果将 namespace 设置为 false，则虚拟网络参数必须是每个虚拟网络特有的。

返回值

此方法具有以下返回值：

名称	说明	类型
virtualNetworkID	新虚拟网络的虚拟网络 ID。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "AddVirtualNetwork",
  "params": {
    "virtualNetworkTag": 2010,
    "name": "network1",
```

```
    "addressBlocks" : [
      { "start": "192.86.5.1", "size": 10 },
      { "start": "192.86.5.50", "size": 20 }
    ],
    "netmask" : "255.255.192.0",
    "gateway" : "10.0.1.254",
    "svip" : "192.86.5.200",
    "attributes" : {}
    "namespace" : 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": {
    "virtualNetworkID": 5
  }
}
```

发布以下版本后新增

9.6

ModifyVirtualNetwork

您可以使用 `ModifyVirtualNetwork` 方法更改现有虚拟网络的属性。

使用此方法可以添加或删除地址块、更改网络掩码或者修改虚拟网络的名称或说明。此外，还可以使用它来启用或禁用命名空间，以及在虚拟网络启用了命名空间的情况下添加或删除网关。

注: 此方法需要 `virtualNetworkID` 或 `virtualNetworkTag` 作为参数，但不能两者都用。



警告: 通过更改 `namespace` 参数来启用或禁用现有虚拟网络的可路由存储 VLAN 功能会中断此虚拟网络处理的所有流量。最好在计划维护时段更改 `namespace` 参数。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
<code>virtualNetworkID</code>	要修改的虚拟网络的唯一标识符。这是由集群分配的虚拟网络 ID。	整型	无	否
<code>virtualNetworkTag</code>	用于标识要修改的虚拟网络的网络标记。	整型	无	否

名称	说明	类型	默认值	必需
addressBlocks	要为此虚拟网络设置的新地址块。此参数可能包含要为现有对象添加的新地址块，也可能会省略需要删除的未用地址块。或者，您也可以增加或减小现有地址块的大小。您只能增加虚拟网络对象的起始地址块的大小，但不能减小它。此对象所需的成员： <ul style="list-style-type: none"> start: IP 地址范围的起点。（字符串） size: 要包含在块中的 IP 地址数。（整型） 	JSON 对象	无	否
gateway	虚拟网络的网关的 IP 地址。只有当 namespace 参数设置为 true 时，此参数才有效。	字符串	无	否
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
name	虚拟网络的新名称。	字符串	无	否
namespace	如果设置为 true，则可以通过重新创建虚拟网络并配置要包含该虚拟网络的命名空间来启用可路由存储 VLAN 功能。如果设置为 false，则可以禁用虚拟网络的 VRF 功能。如果更改此值，则会中断通过此虚拟网络运行的流量。	布尔型	无	否
netmask	此虚拟网络的新网络掩码。	字符串	无	否
svip	此虚拟网络的存储虚拟 IP 地址。您不能更改虚拟网络的 SVIP，而是必须创建新的虚拟网络才能使用其他 SVIP 地址。	字符串	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyVirtualNetwork",
  "params": {
    "virtualNetworkID": 2,
    "name": "ESX-VLAN-3112",
    "addressBlocks": [
      {
        "start": "10.1.112.1",
        "size": 20
      }
    ]
  }
}
```

```

    },
    {
      "start": "10.1.112.100",
      "size": 20
    }
  ],
  "netmask": "255.255.255.0",
  "gateway": "10.0.1.254",
  "svip": "10.1.112.200",
  "attributes": {}
},
"id":1
}

```

响应示例

此方法返回类似于以下示例的响应：

```

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

```

发布以下版本后新增

9.6

ListVirtualNetworks

您可以使用 `ListVirtualNetworks` 方法列出为集群配置的所有虚拟网络。

您可以使用此方法确认集群中的虚拟网络设置。此方法没有必需的参数。但是，要筛选结果，您可以传递一个或多个 `virtualNetworkID` 或 `virtualNetworkTag` 值。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>virtualNetworkID</code>	用于筛选单个虚拟网络列表的网络 ID。	整型	无	否
<code>virtualNetworkTag</code>	用于筛选单个虚拟网络列表的网络标记。	整型	无	否
<code>virtualNetworkIDs</code>	要包含在列表中的网络 ID。	整型数组	无	否
<code>virtualNetworkTags</code>	要包含在列表中的网络标记。	整型数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>virtualNetworks</code>	包含虚拟网络 IP 地址的对象。	<i>virtualNetwork</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVirtualNetworks",
  "params": {
    "virtualNetworkIDs": [5,6]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "virtualNetworks": [
      {
        "addressBlocks": [
          {
            "available": "11000000",
            "size": 8,
            "start": "10.26.250.207"
          }
        ],
        "attributes": null,
        "gateway": "10.26.250.254",
        "name": "2250",
        "namespace": false,
        "netmask": "255.255.255.0",
        "svip": "10.26.250.200",
        "virtualNetworkID": 2250
      },
      {
        "addressBlocks": [
          {
            "available": "11000000",
            "size": 8,
            "start": "10.26.241.207"
          }
        ],
        "attributes": null,
        "gateway": "10.26.241.254",
        "name": "2241",
        "namespace": false,
        "netmask": "255.255.255.0",
        "svip": "10.26.241.200",
        "virtualNetworkID": 2241
      },
      {
        "addressBlocks": [
          {
            "available": "11000000",
            "size": 8,
            "start": "10.26.240.207"
          }
        ]
      }
    ]
  }
}
```

```
    "attributes": null,  
    "gateway": "10.26.240.254",  
    "name": "2240",  
    "namespace": false,  
    "netmask": "255.255.255.0",  
    "svip": "10.26.240.200",  
    "virtualNetworkID": 2240  
  },  
  {  
  }  
]  
}
```

发布以下版本后新增

9.6

RemoveVirtualNetwork

您可以使用 `RemoveVirtualNetwork` 方法删除先前添加的虚拟网络。

注: 此方法需要 `virtualNetworkID` 或 `virtualNetworkTag` 作为参数，但不能两者都用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>virtualNetworkID</code>	标识要删除的虚拟网络的网络 ID。	整型	无	是
<code>virtualNetworkTag</code>	标识要删除的虚拟网络的网络标记。	整型	无	是

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": "RemoveVirtualNetwork",  
  "params": {  
    "virtualNetworkID": 5  
  }  
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

卷 API 方法

您可以通过 Element 软件卷 API 方法管理驻留在存储节点上的卷。您可以通过这些方法创建、修改、克隆和删除卷。此外，还可以使用卷 API 方法收集和显示卷的数据测量结果。

AddInitiatorsToVolumeAccessGroup

您可以使用 AddInitiatorsToVolumeAccessGroup 方法将启动程序添加到指定的卷访问组。

可接受的启动程序 IQN 格式为 iqn.yyyy-mm（其中 y 和 m 为数字）加上一段文本，该文本只能包含数字、小写字母字符、句点 (.)、冒号 (:) 或短划线 (-)。请参见以下示例：

```
iqn.2010-01.com.solidfire:17oi.solidfire-0.1
```

可接受的光纤通道启动程序 WWPN 格式为 Aa:bB:CC:dd:11:22:33:44 或 AabBCCdd11223344。请参见以下示例：

```
21:00:00:0e:1e:11:f1:81
```

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
initiators	列出要包括在卷访问组中的启动程序 ID 或名称（IQN 和 WWPN）。如果传递启动程序名称列表，但不存在相关启动程序，则会创建这些启动程序。如果传递启动程序 ID 列表，但不存在任何启动程序，则该方法将返回错误。 传递启动程序名称已弃用；您应尽可能使用启动程序 ID。	整型数组或字符串数组（已弃用）	[]	是
volumeAccessGroupID	要添加启动程序的卷访问组的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
volumeAccessGroup	包含新修改的卷访问组相关信息的对象。	<i>volumeAccessGroup</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 13171,
  "method": "AddInitiatorsToVolumeAccessGroup",
  "params": {
    "initiators": [116,117],
    "volumeAccessGroupID": 96
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 13171,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        327
      ],
      "initiatorIDs": [
        116,
        117
      ],
      "initiators": [
        "iqn.1993-08.org.debian:01:181324777",
        "iqn.1993-08.org.debian:01:181324888"
      ],
      "name": "northbanktest",
      "volumeAccessGroupID": 96,
      "volumes": [
        346
      ]
    }
  }
}
```

发布以下版本后新增

9.6

AddVolumesToVolumeAccessGroup

您可以使用 `AddVolumesToVolumeAccessGroup` 方法将卷添加到指定的卷访问组。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumes	要添加到卷访问组的卷 ID 列表。	整型数组	无	是
volumeAccessGroupID	向其添加卷的卷访问组的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
volumeAccessGroup	包含新修改的卷访问组相关信息的对象。	<i>volumeAccessGroup</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "AddVolumesToVolumeAccessGroup",
  "params": {
    "volumeAccessGroupID": 96,
    "volumes": [1,2]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        346
      ],
      "initiatorIDs": [
        116,
        117
      ],
      "initiators": [
        "iqn.1993-08.org.debian:01:181324777",
        "iqn.1993-08.org.debian:01:181324888"
      ],
      "name": "northbanktest",
      "volumeAccessGroupID": 96,
      "volumes": [
        1,
        2
      ]
    }
  }
}
```

发布以下版本后新增

9.6

CancelClone

您可以使用 `CancelClone` 方法停止正在执行的卷克隆或卷复制进程。取消组克隆操作时，系统会完成该操作并删除其关联的 `asyncHandle`。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
cloneID	正在运行的克隆进程的克隆 ID。	整型	无	是

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": "CancelClone",
  "params": {
    "cloneID" : 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" : {}
}
```

发布以下版本后新增

9.6

CancelGroupClone

您可以使用 `CancelGroupClone` 方法停止正在对一组卷运行的克隆进程。取消组克隆操作时，系统会完成该操作并删除其关联的 `asyncHandle`。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
groupCloneID	正在运行的克隆进程的克隆 ID。	整型	无	是

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": "CancelGroupClone",
  "params": {
    "cloneID" : 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" : {}
}
```

发布以下版本后新增

9.6

CloneMultipleVolumes

您可以使用 `CloneMultipleVolumes` 方法创建一组指定卷的克隆。同时克隆多个卷时，您可以为这组卷分配一组统一的特性。

在使用 `groupSnapshotID` 参数克隆组快照中的卷之前，您必须先使用 [CreateGroupSnapshot](#) API 方法或 Web UI 创建此组快照。克隆多个卷时，您可以选择使用 `groupSnapshotID`。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>access</code>	新卷的新默认访问方法（如果未被卷数组中传入的信息所覆盖）。	字符串	无	否
<code>enableSnapMirrorReplication</code>	确定卷是否可用于 <code>SnapMirror</code> 端点的复制。可能值： <ul style="list-style-type: none"><code>true</code><code>false</code>	布尔型	<code>false</code>	否
<code>groupSnapshotID</code>	要用作克隆基础的组快照的 ID。	整型	无	否
<code>newAccountID</code>	卷的新帐户 ID（如果未被卷数组中传入的信息所覆盖）。	整型	无	否

名称	说明	类型	默认值	必需
volumes	<p>为新卷指定的一组成员。成员：</p> <ul style="list-style-type: none"> • volumeID: (必需) • access: (可选) 可以是 readOnly、readWrite、locked 或 replicationTarget 之一。 • attributes: (可选) 名称-值对的列表, 采用 JSON 对象格式。 • name: (可选) 克隆的新名称。 • newAccountID: (可选) 新卷的帐户 ID。 • newSize: (可选) 卷的总大小 (以字节为单位)。大小将向上取整为最接近的 MB。 <p>如果未指定可选成员, 这些值将从源卷继承。</p>	JSON 对象数组	无	是 (volumeID)

返回值

此方法具有以下返回值：

名称	说明	类型
asyncHandle	从异步方法调用返回的值。	整型
groupCloneID	新组克隆的唯一 ID。	整型
members	源和目标卷对的卷 ID 列表。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CloneMultipleVolumes",
  "params": {
    "volumes": [
      {
        "volumeID": 5
        "name": "foxhill",
        "access": "readOnly"
      },
      {
        "volumeID": 18
      },
      {
        "volumeID": 20
      }
    ]
  }
},
```

```
    "id": 1  
  }
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id": 1,  
  "result": {  
    "asyncHandle": 12,  
    "groupCloneID": 4,  
    "members": [  
      {  
        "srcVolumeID": 5,  
        "volumeID": 29  
      },  
      {  
        "srcVolumeID": 18,  
        "volumeID": 30  
      },  
      {  
        "srcVolumeID": 20,  
        "volumeID": 31  
      }  
    ]  
  }  
}
```

发布以下版本后新增

9.6

CloneVolume

您可以使用 `CloneVolume` 方法创建卷的副本。此方法是异步方法，其完成所需的时间可能不同。

一旦发出 `CloneVolume` 请求，克隆过程就会开始；调用此 API 方法时，此过程会反映卷的状态。您可以使用 [GetAsyncResult](#) 方法来确定克隆过程何时完成以及何时可以连接新卷。您可以使用 [ListSyncJobs](#) 来查看创建克隆的进度。卷的初始属性和服务质量设置将从所克隆的卷继承。您可以通过 [ModifyVolume](#) 更改这些设置。

注：克隆的卷不会继承源卷的卷访问组成员资格。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
access	允许对新卷进行的访问。如果未指定值，则访问值不会更改。可能值： <ul style="list-style-type: none"> readOnly: (可选) 仅允许执行读取操作。 readWrite: (可选) 允许执行读取和写入。 locked: (可选) 不允许执行任何读取或写入。如果未指定，将使用所克隆卷的访问值。 replicationTarget: (可选) 将一个卷标识为一组配对卷的目标卷。如果卷未进行配对，则访问状态为 locked。 	字符串	无	否
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
enable512e	指定新卷是否应使用 512 字节扇区仿真。如果未指定、则使用正在克隆的卷的设置。	布尔型	设置原始卷	否
enableSnapMirrorReplication	确定卷是否可用于 SnapMirror 端点的复制。可能值： <ul style="list-style-type: none"> true false 	布尔型	false	否
name	新克隆卷的名称；长度必须是 1 到 64 个字符。	字符串	无	是
newAccountID	新卷的所有者的帐户 ID。如果未指定、则使用正在克隆的卷的所有者的帐户 ID。	整型	原始卷所有者的帐户 ID	否
newSize	卷的新大小（以字节为单位）。可大于或小于所克隆卷的大小。如果未指定，则卷大小不变。大小将向上取整为最接近的 MB。	整型	无	否
snapshotID	用作克隆源的快照的 ID。如果未提供 ID，则将使用当前活动卷。	整型	无	否
volumeID	要克隆的卷的卷 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
asyncHandle	用于获取操作结果的句柄值。	整型
cloneID	新克隆卷的克隆 ID。	整型
curve	应用于克隆的 QoS 曲线值。	JSON 对象
volume	包含新克隆卷的相关信息的对象。	<i>volume</i>

名称	说明	类型
volumeID	新克隆卷的卷 ID。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CloneVolume",
  "params": {
    "volumeID" : 5,
    "name" : "mysqldata-snapshot1",
    "access" : "readOnly"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "asyncHandle": 42,
    "cloneID": 37,
    "volume": {
      "access": "readOnly",
      "accountID": 1,
      "attributes": {},
      "blockSize": 4096,
      "createTime": "2016-03-31T22:26:03Z",
      "deleteTime": "",
      "enable512e": true,
      "iqn": "iqn.2010-01.com.solidfire:jyay.mysqldata-snapshot1.680",
      "name": "mysqldata-snapshot1",
      "purgeTime": "",
      "qos": {
        "burstIOPS": 100,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 100,
        "minIOPS": 50
      },
      "scsiEUIDeviceID": "6a796179000002a8f47acc0100000000",
      "scsiNAADeviceID": "6f47acc1000000006a796179000002a8",
      "sliceCount": 0,
      "status": "init",
      "totalSize": 1000341504,
    }
  }
}
```

```

        "virtualVolumeID": null,
        "volumeAccessGroups": [],
        "volumeID": 680,
        "volumePairs": []
    },
    "volumeID": 680
}
}

```

发布以下版本后新增

9.6

相关参考

[GetAsyncResult](#) on page 84

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

[ListSyncJobs](#) on page 396

您可以使用 `ListSyncJobs` 方法获取有关在 `Element` 存储集群上运行的同步作业的信息。此方法可返回有关分区、克隆、块和远程同步作业的信息。

[ModifyVolume](#) on page 416

您可以使用 `ModifyVolume` 方法修改现有卷上的设置。一次只能对一个卷进行修改，并且更改将立即生效。

CopyVolume

您可以使用 `CopyVolume` 方法将现有卷的数据内容覆盖为另一个卷（或快照）的数据内容。目标卷的属性（例如 IQN、QoS 设置、大小、帐户和卷访问组成员资格）不会更改。目标卷必须已存在且大小与源卷相同。

客户端最好在该操作开始之前卸载目标卷。如果在操作期间修改目标卷，所做的更改将会丢失。完成此操作所需的时间可能不同。您可以使用 [GetAsyncResult](#) 方法确定进程何时完成，并使用 [ListSyncJobs](#) 查看复制进度。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>dstVolumeID</code>	要覆盖的卷的卷 ID。	整型	无	是
<code>volumeID</code>	要从此读取卷的卷 ID。	整型	无	是
<code>snapshotID</code>	用作克隆源的快照的 ID。如果未提供 ID，则将使用当前活动卷。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>asyncHandle</code>	用于获取操作结果的句柄值。	整型
<code>cloneID</code>	新克隆卷的克隆 ID。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CopyVolume",
  "params": {
    "volumeID" : 3,
    "dstVolumeID" : 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": {
    "asyncHandle": 9,
    "cloneID": 5
  }
}
```

发布以下版本后新增

9.6

相关参考

[GetAsyncResult](#) on page 84

可以使用 `GetAsyncResult` 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 `GetAsyncResult` 轮询此方法返回的 `asyncHandle` 值。

[ListSyncJobs](#) on page 396

您可以使用 `ListSyncJobs` 方法获取有关在 `Element` 存储集群上运行的同步作业的信息。此方法可返回有关分区、克隆、块和远程同步作业的信息。

CreateQoSPolicy

您可以使用 `CreateQoSPolicy` 方法创建 `QoSPolicy` 对象，稍后可在创建或修改卷时对卷应用此对象。`QoS` 策略具有唯一的 ID、名称和 `QoS` 设置。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
name	QoS 策略的名称；例如 gold、platinum 或 silver。	字符串	无	是
qos	此策略表示的 QoS 设置。	<i>QoS</i>	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
qosPolicy	新创建的 QoSPolicy 对象。	<i>QoSPolicy</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 68,
  "method": "CreateQoSPolicy",
  "params": {
    "name": "bronze",
    "qos": {
      "minIOPS": 50,
      "maxIOPS": 15000,
      "burstIOPS": 15000
    }
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 68,
  "result": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15000,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 15000,
        "minIOPS": 50
      },
      "qosPolicyID": 2,
      "volumeIDs": []
    }
  }
}
```

发布以下版本后新增

10.0

CreateVolume

您可以使用 `CreateVolume` 方法在集群上创建新的空卷。创建完卷后，即可通过 iSCSI 连接该卷。

如果在创建卷时未指定 QoS 值，则该卷将使用默认值。您可以使用 `GetDefaultQoS` 方法来查看卷的默认值。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>accountID</code>	拥有此卷的帐户的 ID。	整型	无	是
<code>associateWithQoSPolicy</code>	将卷与指定的 QoS 策略相关联。可能值： <ul style="list-style-type: none"> <code>true</code>：将卷与 <code>QoSPolicyID</code> 参数中指定的 QoS 策略相关联。 <code>false</code>：不将卷与 <code>QoSPolicyID</code> 参数中指定的 QoS 策略相关联。如果为 <code>false</code>，无论是否在 <code>QoSPolicy</code> 参数中指定 QoS 策略，都将删除任何现有的策略关联。 	布尔型	<code>true</code>	否
<code>attributes</code>	名称-值对的列表，采用 JSON 对象格式。属性总大小必须小于 1000 B 或 1 KB，包括 JSON 格式的字符。	JSON 对象	无	否
<code>enable512e</code>	启用 512 字节的扇区模拟。可能值： <ul style="list-style-type: none"> <code>true</code>：卷提供 512 字节的扇区模拟。 <code>false</code>：不启用 512e 模拟。 	布尔型	无	是
<code>enableSnapMirrorReplication</code>	确定卷是否可用于 <code>SnapMirror</code> 端点的复制。可能值： <ul style="list-style-type: none"> <code>true</code> <code>false</code> 	布尔型	<code>false</code>	否
<code>name</code>	卷访问组的名称（可由用户指定）。此名称不必唯一，但建议使用唯一名称。长度必须是 1 到 64 个字符。	字符串	无	是

名称	说明	类型	默认值	必需
qos	该卷的初始服务质量设置。如果未指定，将使用默认值。可能值： <ul style="list-style-type: none"> • minIOPS • maxIOPS • burstIOPS 	QoS 对象	无	否
qosPolicyID	策略 ID，该策略中的 QoS 设置应该应用于指定卷。此参数与 qos 参数不能同时使用。	整型	无	否
totalSize	卷的总大小（以字节为单位）。大小将向上取整为最接近的 MB。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
volume	包含新创建卷的相关信息的对象。	<i>volume</i>
volumeID	新建卷的卷 ID。	整型
curve	此曲线是一组键-值对。这些键表示 I/O 大小（以字节为单位）。这些值表示在特定 I/O 大小情形下执行 IOP 的成本。此曲线相对于 100 IOPS 处的 4096 字节操作集计算而得。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateVolume",
  "params": {
    "name": "mysqldata",
    "accountID": 1,
    "totalSize": 107374182400,
    "enable512e": false,
    "attributes": {
      "name1": "value1",
      "name2": "value2",
      "name3": "value3"
    },
    "qos": {
      "minIOPS": 50,
      "maxIOPS": 500,
      "burstIOPS": 1500,
      "burstTime": 60
    }
  }
},
```

```
    "id": 1  
  }
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id": 1,  
  "result": {  
    "curve": {  
      "4096": 100,  
      "8192": 160,  
      "16384": 270,  
      "32768": 500,  
      "65536": 1000,  
      "131072": 1950,  
      "262144": 3900,  
      "524288": 7600,  
      "1048576": 15000  
    },  
    "volume": {  
      "access": "readWrite",  
      "accountID": 1,  
      "attributes": {  
        "name1": "value1",  
        "name2": "value2",  
        "name3": "value3"  
      },  
      "blockSize": 4096,  
      "createTime": "2016-03-31T22:20:22Z",  
      "deleteTime": "",  
      "enable512e": false,  
      "iqn": "iqn.2010-01.com.solidfire:mysqldata.677",  
      "name": "mysqldata",  
      "purgeTime": "",  
      "qos": {  
        "burstIOPS": 1500,  
        "burstTime": 60,  
        "curve": {  
          "4096": 100,  
          "8192": 160,  
          "16384": 270,  
          "32768": 500,  
          "65536": 1000,  
          "131072": 1950,  
          "262144": 3900,  
          "524288": 7600,  
          "1048576": 15000  
        },  
        "maxIOPS": 500,  
        "minIOPS": 50  
      },  
      "scsiEUIDeviceID": "6a796179000002a5f47acc0100000000",  
      "scsiNAADeviceID": "6f47acc1000000006a796179000002a5",  
      "sliceCount": 0,  
      "status": "active",  
      "totalSize": 107374182400,  
      "virtualVolumeID": null,  
      "volumeAccessGroups": [],  
      "volumeID": 677,  
    },  
  },  
}
```

```

        "volumePairs": []
      },
      "volumeID": 677
    }
  }
}

```

发布以下版本后新增

9.6

相关参考

[GetDefaultQoS](#) on page 383

您可以使用 `GetDefaultQoS` 方法获取新建卷的默认服务质量 (QoS) 值。

CreateVolumeAccessGroup

您可以使用 `CreateVolumeAccessGroup` 创建新的卷访问组。创建卷访问组时，您需要为其提供名称，还可以为其输入启动程序和卷。

添加到卷访问组中的任何启动程序 IQN 可以访问组中的任何卷，而无需 CHAP 身份验证。

注：克隆的卷不会继承源卷的卷访问组成员资格。

创建卷访问组时，请考虑以下事项：

- 一个卷访问组最多可以包含 64 个启动程序 IQN。
- 一个启动程序只能属于一个卷访问组。
- 一个卷访问组最多可以包含 2000 个卷。
- 每个卷访问组最多可以属于四个其他卷访问组。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
initiators	列出要包括在卷访问组中的启动程序 ID 或名称 (IQN 和 WWPN)。如果传递启动程序名称列表，但不存在相关启动程序，则会创建这些启动程序。如果传递启动程序 ID 列表，但不存在任何启动程序，则该方法将返回错误。传递启动程序名称已弃用；您应尽可能使用启动程序 ID。	整型数组或字符串数组 (已弃用)	[]	否
name	卷访问组的名称。此名称不必唯一，但建议使用唯一名称。长度必须是 1 到 64 个字符。	字符串	无	是
volumes	列出要包括在卷访问组中的卷 ID。	整型数组	[]	否
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	{}	否

返回值

此方法具有以下返回值：

名称	说明	类型
volumeAccessGroup	包含新创建的卷访问组的相关信息的对象。	<i>volumeAccessGroup</i>
volumeAccessGroupID	新创建的卷访问组的 ID。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateVolumeAccessGroup",
  "params": {
    "name": "myaccessgroup",
    "initiators": ["iqn.1993-08.org.debian: 01: a31b1d799d5c"],
    "volumes": [327],
    "attributes": {}
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": null,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [],
      "initiatorIDs": [
        95
      ],
      "initiators": [
        "iqn.1993-08.org.debian: 01: a31b1d799d5c"
      ],
      "name": "myaccessgroup",
      "volumeAccessGroupID": 96,
      "volumes": [
        327
      ]
    },
    "volumeAccessGroupID": 96
  }
}
```

发布以下版本后新增

9.6

相关参考

[GetAsyncResult](#) on page 84

可以使用 [GetAsyncResult](#) 检索异步方法调用的结果。部分方法调用需要运行一段时间，并且在系统发送初始响应时可能仍未完成。要获得此方法调用的状态或结果，请使用 [GetAsyncResult](#) 轮询此方法返回的 `asyncHandle` 值。

[ListSyncJobs](#) on page 396

您可以使用 `ListSyncJobs` 方法获取有关在 **Element** 存储集群上运行的同步作业的信息。此方法可返回有关分区、克隆、块和远程同步作业的信息。

[Modify Volume](#) on page 416

您可以使用 `ModifyVolume` 方法修改现有卷上的设置。一次只能对一个卷进行修改，并且更改将立即生效。

CreateBackupTarget

您可以使用 `CreateBackupTarget` 创建和存储备份目标信息、以便在每次创建备份时不需要重新输入。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
name	备份目标的名称。	字符串	无	是
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	是（但可能为空）

返回值

此方法具有以下返回值：

名称	说明	类型
backupTargetID	分配给新备份目标的唯一标识符。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateBackupTarget",
  "params": {
    "name": "mytargetbackup"
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

CreateInitiators

您可以使用 `CreateInitiators` 创建多个新启动程序 IQN 或全球通用端口名称 (World Wide Port Name, WWPN)，也可以为其分配别名和属性。使用 `CreateInitiators` 创建新启动程序时，也可以将这些启动程序添加到卷访问组中。

如果该操作无法创建该参数中提供的某个启动程序，则该方法将返回错误且不会创建任何启动程序（即无法创建一部分启动程序）。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
initiators	<p>包含每个新启动程序的特性的对象列表。对象：</p> <ul style="list-style-type: none"> • alias：（可选）要分配给此启动程序的友好名称。（字符串） • attributes：（可选）要分配给此启动程序的一组 JSON 属性。（JSON 对象） • chapUsername：（可选）此启动程序的唯一 CHAP 用户名。如果在创建期间未指定此名称，并且 requireChap 为 true，则默认为此启动程序名称 (IQN)。（字符串） • initiatorSecret：（可选）用于对此启动程序进行身份验证的 CHAP 密钥。如果在创建期间未指定此密钥，并且 requireChap 为 true，则默认为一个随机生成的密钥。（字符串） • name：（必需）要创建的启动程序的名称 (IQN 或 WWPN)。（字符串） • requireChap：（可选）如果在会话登录期间需要对此启动程序使用 CHAP，则为 True。（布尔值） • targetSecret：（可选）用于对目标进行身份验证的 CHAP 密钥（使用相互 CHAP 身份验证时）。如果在创建期间未指定此密钥，并且 requireChap 为 true，则默认为一个随机生成的密钥。（字符串） • volumeAccessGroupID：（可选）要将此新创建的启动程序添加到的卷访问组的 ID。（整型） 	JSON 对象数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
initiators	一个对象列表，用于描述新创建的启动程序。	<i>initiator</i> 数组

错误

此方法可能返回以下错误：

名称	说明
xInitiatorExists	如果所选的启动程序名称已存在，则返回此错误。

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 3291,
  "method": "CreateInitiators",
  "params": {
    "initiators": [
      {
        "name": "iqn.1993-08.org.debian:01:288170452",
        "alias": "example1"
      },
      {
        "name": "iqn.1993-08.org.debian:01:297817012",
        "alias": "example2"
      }
    ]
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 3291,
  "result": {
    "initiators": [
      {
        "alias": "example1",
        "attributes": {},
        "initiatorID": 145,
        "initiatorName": "iqn.1993-08.org.debian:01:288170452",
        "volumeAccessGroups": []
      },
      {
        "alias": "example2",
        "attributes": {},
        "initiatorID": 146,
        "initiatorName": "iqn.1993-08.org.debian:01:297817012",
        "volumeAccessGroups": []
      }
    ]
  }
}
```

发布以下版本后新增

9.6

相关参考

[ListInitiators](#) on page 393

您可以使用 `ListInitiators` 方法获取启动程序 IQN 或全球通用端口名称 (World Wide Port Name, WWPN) 的列表。

DeleteInitiators

您可以使用 `DeleteInitiators` 从系统（以及任何关联的卷或卷访问组）中删除一个或多个启动程序。

如果 `DeleteInitiators` 无法删除此参数中提供的启动程序之一，则系统将返回错误且不会删除任何启动程序（无法执行部分操作）。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
initiators	要删除的启动程序的 ID 数组。	整型数组	无	是

返回值

此方法没有返回值。

错误

此方法可能返回以下错误：

名称	说明
xInitiatorDoesNotExist	如果所选的启动程序名称不存在，则返回此错误。

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 5101,
  "method": "DeleteInitiators",
  "params": {
    "initiators": [
      145,
      147
    ]
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 5101,
  "result": {}
}
```

发布以下版本后新增

9.6

DeleteQoSPolicy

您可以使用 `DeleteQoSPolicy` 方法从系统中删除 QoS 策略。使用此策略创建或修改的所有卷的 QoS 设置不受影响。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
qosPolicyID	要删除的 QoS 策略的 ID。	整型	无	是

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：

```
{
  "id": 663,
  "method": "DeleteQoSPolicy",
  "params": {
    "qosPolicyID": 4
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 663,
  "result": {}
}
```

发布以下版本后新增

9.6

DeleteVolume

您可以使用 `DeleteVolume` 方法将活动卷标记为待删除。标记后的卷会在清理间隔过后被清除（永久删除）。

在请求删除某个卷之后，与该卷的所有活动 iSCSI 连接都会立即终止，在这种状态下，不再允许进行任何其他连接。标记的卷不会在目标发现请求中返回。

已标记为删除的卷的所有快照都不会受影响。这些快照会一直保留到相关卷从系统中清除为止。如果某个卷被标记为删除，但该卷正在执行批量卷读取或批量卷写入操作，则批量卷读取或写入操作将停止。

如果要删除的卷已与某个卷配对，则系统将暂停两个配对卷之间的复制过程，并且处于已删除状态的卷不会再发生任何数据传输。与已删除卷配对的远程卷将进入 `PausedMisconfigured` 状态，不会再向其发送数据，也不会再从已删除的卷发送数据。在清

除已删除的卷之前，您可以还原该卷，并继续传输数据。如果已删除卷已从系统中清除，其配对卷将进入 `StoppedMisconfigured` 状态，且卷配对状态也会被删除。已清除的卷将永久不可用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	要删除的卷的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
volume	包含已删除卷的相关信息的对象。	<i>volume</i>
volumeID	已删除卷的卷 ID。	整型
curve	此曲线是一组键-值对。这些键表示 I/O 大小（以字节为单位）。这些值表示在特定 I/O 大小情形下执行 IOP 的成本。此曲线相对于 100 IOPS 处的 4096 字节操作集计算而得。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "DeleteVolume",
  "params": {
    "volumeID" : 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": {
    "volume": {
      "access": "readWrite",
      "accountID": 1,
      "attributes": {
        "name1": "value1",
        "name2": "value2",
        "name3": "value3"
      },
    },
    "blockSize": 4096,
    "createTime": "2016-03-28T16:16:13Z",
    "deleteTime": "2016-03-31T22:59:42Z",
    "enable512e": true,
    "iqn": "iqn.2010-01.com.solidfire:jyay.1459181777648.5",
    "name": "1459181777648",
    "purgeTime": "2016-04-01T06:59:42Z",
    "qos": {
```

```
        "burstIOPS": 150,
        "burstTime": 60,
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "maxIOPS": 100,
        "minIOPS": 60
    },
    "scsiEUIDeviceID": "6a79617900000005f47acc0100000000",
    "scsiNAADeviceID": "6f47acc1000000006a79617900000005",
    "sliceCount": 1,
    "status": "deleted",
    "totalSize": 1000341504,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
        1
    ],
    "volumeID": 5,
    "volumePairs": []
}
}
```

发布以下版本后新增

9.6

DeleteVolumes

您可以使用 `DeleteVolumes` 方法将多个（最多 500 个）活动卷标记为待删除。标记后的卷会在清理间隔过后被清除（永久删除）。

在请求删除卷之后，与卷的所有活动 iSCSI 连接都会立即终止，在这种状态下，不再允许进行任何其他连接。标记的卷不会在目标发现请求中返回。

已标记为删除的卷的所有快照都不会受影响。这些快照会一直保留到相关卷从系统中清除为止。如果某个卷被标记为删除，但该卷正在执行批量卷读取或批量卷写入操作，则批量卷读取或写入操作将停止。

如果要删除的卷已与某个卷配对，则系统将暂停两个配对卷之间的复制过程，并且处于已删除状态的卷不会再发生任何数据传输。与已删除卷配对的远程卷将进入 `PausedMisconfigured` 状态，不会再向其发送数据，也不会再从已删除的卷发送数据。在清除已删除的卷之前，您可以还原这些卷，并继续传输数据。如果已删除卷已从系统中清除，其配对卷将进入 `StoppedMisconfigured` 状态，且卷配对状态也会被删除。已清除的卷将永久不可用。

参数

此方法具有以下输入参数。

注：至少需要提供以下参数之一，并且您只能使用其中一个参数（这些参数不能同时使用）。

名称	说明	类型	默认值	必需
volumeIDs	要从系统中删除的卷的 ID 列表。	整型数组	无	请参见注释。
volumeAccessGroupIDs	卷访问组 ID 列表。在此列表中指定的所有卷访问组中的所有卷将从系统删除。	整型数组	无	请参见注释。
accountIDs	帐户 ID 列表。这些帐户中的所有卷将从系统删除。	整型数组	无	请参见注释。

返回值

此方法具有以下返回值：

名称	说明	类型
volumes	新删除卷的相关信息。	<i>volume</i>
curve	此曲线是一组键-值对。这些键表示 I/O 大小（以字节为单位）。这些值表示在特定 I/O 大小情形下执行 IOP 的成本。此曲线相对于 100 IOPS 处的 4096 字节操作集计算而得。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "DeleteVolumes",
  "params": {
    "accountIDs" : [1, 2, 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": {
    "volumes" : [ {
      "access": "readWrite",
      "accountID": 1,
      "attributes": {},
      "blockSize": 4096,
      "createTime": "2015-03-06T18:50:56Z",
      "deleteTime": "",
      "enable512e": False,
      "iqn": "iqn.2010-01.com.solidfire:pzsr.vclient-030-v00001.1",
      "name": "vclient-030-v00001",
      "qos": {
        "burstIOPS": 15000,
```

```

        "burstTime": 60,
        "curve": {},
        "maxIOPS": 15000,
        "minIOPS": 100
    },
    "purgeTime": "",
    "sliceCount": 1,
    "scsiEUIDeviceID": "707a737200000001f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000707a737200000001",
    "status": "active",
    "totalSize": 10000003072,
    "virtualVolumeID": 5,
    "volumeAccessGroups": [],
    "volumePairs": [],
    "volumeID": 1
} ]
}
}

```

发布以下版本后新增

9.6

DeleteVolumeAccessGroup

您可以使用 DeleteVolumeAccessGroup 删除卷访问组。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeAccessGroupID	要删除的卷访问组的 ID。	整型	无	是
deleteOrphanInitiators	指定是否删除启动程序对象。可能值： <ul style="list-style-type: none"> • true: 在启动程序对象从卷访问组中删除之后，删除这些对象。 • false: 在启动程序对象从卷访问组中删除之后，不删除这些对象。这是默认值。 	布尔型	false	否
force	添加此标志将强制删除卷访问组，即使其具有虚拟网络 ID 或标记也是如此。可能值： <ul style="list-style-type: none"> • true: 卷访问组将被删除。 • false: 默认值。如果卷访问组具有虚拟网络 ID 或标记，则不删除。 	布尔型	false	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": "DeleteVolumeAccessGroup",
  "params": {
    "force": true,
    "volumeAccessGroupID" : 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": {}
}
```

发布以下版本后新增

9.6

GetBackupTarget

您可以使用 `GetBackupTarget` 方法返回已创建的特定备份目标的相关信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
backupTargetID	分配给备份目标的唯一标识符。	整型	无	是
name	备份目标的名称。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
backupTarget	名称-值对的列表，采用 JSON 对象格式。	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 1,
  "method": "GetBackupTarget",
  "params": {
    "backupTargetID": 1
  }
}
```

```
}  
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id": 1,  
  "result": {  
    "backupTarget": {  
      "attributes": {  
        "size": 100  
      },  
      "backupTargetID": 1,  
      "name": "mytargetbackup"  
    }  
  }  
}
```

发布以下版本后新增

9.6

GetVolumeStats

您可以使用 `GetVolumeStats` 方法获取单个卷的活动测量结果概要。具体值是在卷创建过程中累计的值。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
volumeID	指定为其收集统计信息的卷。	整型	无	是

返回值

此方法具有以下返回值:

名称	说明	类型
volumeStats	卷活动信息。	<i>volumeStats</i>

请求示例

此方法的请求类似于以下示例:

```
{  
  "method": "GetVolumeStats",  
  "params": {  
    "volumeID": 32  
  },  
  "id": 1  
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "volumeStats": {
      "accountID": 1,
      "actualIOPS": 0,
      "asyncDelay": null,
      "averageIOPSize": 0,
      "burstIOPSCredit": 0,
      "clientQueueDepth": 0,
      "desiredMetadataHosts": null,
      "latencyUsec": 0,
      "metadataHosts": {
        "deadSecondaries": [],
        "liveSecondaries": [
          32
        ],
        "primary": 60
      },
      "nonZeroBlocks": 0,
      "readBytes": 0,
      "readBytesLastSample": 0,
      "readLatencyUsec": 0,
      "readOps": 0,
      "readOpsLastSample": 0,
      "samplePeriodMsec": 0,
      "throttle": 0,
      "timestamp": "2016-04-01T21:01:39.130840Z",
      "unalignedReads": 0,
      "unalignedWrites": 0,
      "volumeAccessGroups": [],
      "volumeID": 1,
      "volumeSize": 5000658944,
      "volumeUtilization": 0,
      "writeBytes": 0,
      "writeBytesLastSample": 0,
      "writeLatencyUsec": 0,
      "writeOps": 0,
      "writeOpsLastSample": 0,
      "zeroBlocks": 1220864
    }
  }
}
```

发布以下版本后新增

9.6

GetDefaultQoS

您可以使用 `GetDefaultQoS` 方法获取新建卷的默认服务质量 (QoS) 值。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
QoS	默认 QoS 值。	<i>QoS</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetDefaultQoS",
  "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" : {
    "burstIOPS" : 15000,
    "burstTime" : 60,
    "curve" : {
      "1048576" : 15000,
      "131072" : 1900,
      "16384" : 270,
      "262144" : 3000,
      "32768" : 500,
      "4096" : 100,
      "524288" : 7500,
      "65536" : 1000,
      "8192" : 160
    },
    "maxIOPS" : 15000,
    "minIOPS" : 100
  }
}
```

发布以下版本后新增

9.6

GetQoSPolicy

您可以使用 `GetQoSPolicy` 方法从系统获取有关特定 QoS 策略的详细信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
qosPolicyID	要检索的策略的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
qosPolicy	请求的 QoS 策略的详细信息。	<i>QoSPolicy</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetQoSPolicy",
  "params": {
    "qosPolicyID": 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": {
    "qosPolicy": {
      "name": "bronze",
      "qos": {
        "burstIOPS": 15002,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        }
      },
      "maxIOPS": 15002,
      "minIOPS": 51
    },
    "qosPolicyID": 2,
    "volumeIDs": [
      2
    ]
  }
}
```

发布以下版本后新增

10.0

GetVolumeCount

您可以使用 `GetVolumeCount` 方法获取系统中当前所含的卷数量。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
count	系统中当前所含的卷数量。	整型

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "GetVolumeCount",  
  "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": {  
    "count": 7  
  }  
}
```

发布以下版本后新增

9.6

GetVolumeAccessGroupEfficiency

您可以使用 `GetVolumeAccessGroupEfficiency` 方法获取有关卷访问组的效率信息。仅会使用在此 API 方法中作为参数提供的卷访问组来计算容量。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeAccessGroupID	指定用于计算容量的卷访问组。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
compression	通过压缩数据为卷访问组中的所有卷节省的空间量。用一个比率表示，其中值 1 表示已在未压缩的情况下存储了数据。	浮点型
deduplication	通过执行重复数据删除为卷访问组中的所有卷节省的空间量。用一个比率表示。	浮点型
thinProvisioning	已用空间与已分配用于存储数据的空间量之比。用一个比率表示。	浮点型
timestamp	执行垃圾收集后最后一次收集效率数据的时间。	ISO 8601 数据字符串
missingVolumes	无法在其中查询效率数据的卷。造成卷缺失的可能原因包括近期执行了垃圾收集操作、网络暂时断开或服务在垃圾收集周期后重新启动。	整型数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetVolumeAccessGroupEfficiency",
  "params": {
    "volumeAccessGroupID": 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": {
    "compression": 2.006012925331075,
    "deduplication": 1,
    "missingVolumes": [],
    "thinProvisioning": 1.009861932938856,
    "timestamp": "2014-03-10T17:05:27Z"
  }
}
```

发布以下版本后新增

9.6

GetVolumeEfficiency

您可以使用 `GetVolumeEfficiency` 方法获取有关卷的信息。仅作为此 API 方法的参数而提供的卷用于计算容量。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	指定用于计算容量的卷。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
compression	通过压缩数据在单个卷上节省的空间量。用一个比率表示，其中 1 表示已在未压缩的情况下存储了数据。	浮点型
deduplication	通过执行重复数据删除在单个卷上节省的空间量。用一个比率表示。	浮点型
missingVolumes	无法在其中查询效率数据的卷。造成卷缺失的可能原因包括垃圾收集 (GC) 操作还不到一小时、网络暂时性丢失或服务在 GC 周期之后重新启动。	整型数组
thinProvisioning	已用空间与已分配用于存储数据的空间量之比。用一个比率表示。	浮点型
timestamp	GC 后最后一次收集效率数据的时间。	ISO 8601 数据字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetVolumeEfficiency",
  "params": {
    "volumeID": 606
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "compression": 2.001591240821456,
    "deduplication": 1,
    "missingVolumes": [],
    "thinProvisioning": 1.009861932938856,
    "timestamp": "2014-03-10T16:06:33Z"
  }
}
```

发布以下版本后新增

9.6

ListActiveVolumes

您可以使用 `ListActiveVolumes` 方法获取系统中当前所含活动卷的列表。卷列表按卷 ID 排序，并且可以在多个部分（页面）中返回。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>includeVirtualVolumes</code>	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否
<code>startVolumeID</code>	要返回的起始卷 ID。如果不存在具有此卷 ID 的卷，则会使用下一个卷（按卷 ID 顺序）作为列表的起点。要翻阅该列表，传递的值应是上一个响应中最后一个卷的卷 ID 加 1。	整型	0	否
<code>limit</code>	要返回的 <code>volume info</code> 对象的最大数量。如果指定 0（零），将返回所有卷（无限制）。	整型	（无限制）	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumes</code>	活动卷的列表。	<i>volume</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListActiveVolumes",
  "params": {
    "startVolumeID" : 0,
    "limit" : 1000
  },
  "id" : 1
}
```

响应示例

由于此响应示例较长，因此将其记录在补充主题中。

发布以下版本后新增

9.6

ListBackupTargets

您可以使用 `ListBackupTargets` 方法获取已创建的所有备份目标的相关信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
backupTargets	为每个备份目标返回的对象。包含的对象： <ul style="list-style-type: none"><code>attributes</code>：名称-值对的列表，采用 JSON 对象格式。（JSON 对象）<code>backupTargetID</code>：分配给备份目标的唯一标识符。（整型）<code>name</code>：备份目标的名称。（字符串）	JSON 对象

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListBackupTargets",
  "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": {
    "backupTargets": [
      {
        "attributes" : {},
        "backupTargetID" : 1,
        "name" : "mytargetbackup"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListBulkVolumeJobs

您可以使用 `ListBulkVolumeJobs` 方法获取有关系统中正在发生的每个批量卷读取或写入操作的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
bulkVolumeJobs	包含每个批量卷作业的信息的数组。	<i>bulkVolumeJob</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListBulkVolumeJobs",
  "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": {
    "bulkVolumeJobs": [
      {
        "attributes": {
          "blocksPerTransfer": 1024,
          "firstPendingLba": 216064,
          "nLbas": 2441472,
          "nextLba": 226304,
          "pendingLbas": "[220160, 223232, 221184, 224256, 217088, 225280, 222208, 218112, 219136, 216064]",
          "percentComplete": 8,
          "startLba": 0
        },
        "bulkVolumeID": 2,
        "createTime": "2015-05-07T14:52:17Z",
        "elapsedTime": 44,
        "format": "native",
        "key": "eaffb0526d4fb47107061f09bfc9a806",
        "percentComplete": 8,
        "remainingTime": 506,
        "script": "bv_internal.py",
        "snapshotID": 509,
        "srcVolumeID": 3,
        "status": "running",
        "type": "read"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListDeletedVolumes

您可以使用 `ListDeletedVolumes` 方法检索已标记为待删除且已从系统中清除的卷的列表。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>includeVirtualVolumes</code>	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumes</code>	列出已删除的卷。	<i>volume</i> 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法的响应类似于以下示例：

```
{
  "id": 1,
  "result": {
    "volumes": [
      {
        "access": "readWrite",
        "accountID": 2,
        "attributes": {},
        "blockSize": 4096,
        "createTime": "2018-06-24T03:13:13Z",
        "deleteTime": "2018-07-22T16:12:39Z",
        "enable512e": true,
        "iqn": "iqn.2010-01.com.solidfire:0oto.deletethis.23",
        "name": "deleteThis",
        "purgeTime": "2016-07-23T00:12:39Z",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,

```

```

        "65536": 1000,
        "131072": 1950,
        "262144": 3900,
        "524288": 7600,
        "1048576": 15000
    },
    "maxIOPS": 15000,
    "minIOPS": 50
},
"scsiEUIDeviceID": "306f746f00000017f47acc0100000000",
"scsiNAADeviceID": "6f47acc100000000306f746f00000017",
"sliceCount": 1,
"status": "deleted",
"totalSize": 1396703232,
"virtualVolumeID": null,
"volumeAccessGroups": [],
"volumeID": 23,
"volumePairs": []
}
    ]
}
}

```

发布以下版本后新增

9.6

ListInitiators

您可以使用 `ListInitiators` 方法获取启动程序 IQN 或全球通用端口名称 (World Wide Port Name, WWPN) 的列表。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>initiators</code>	要检索的启动程序 ID 列表。您可以应用此参数或 <code>startInitiatorID</code> 参数，但不能同时应用这两个参数。	整型数组	无	否
<code>startInitiatorID</code>	列表开头处的启动程序 ID。您可以应用此参数或 <code>initiators</code> 参数，但不能同时应用这两个参数。	整型	0	否
<code>limit</code>	要返回的启动程序对象的最大数量。	整型	(无限制)	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>initiators</code>	启动程序信息列表。	<i>initiator</i> 数组

异常

此方法可能存在以下异常：

名称	说明
xInvalidParameter	如果将 startInitiatorID 和 initiators 参数包含在同一方法调用中，则会引发此异常。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListInitiators",
  "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": {
    "initiators": [
      {
        "alias": "",
        "attributes": {},
        "initiatorID": 2,
        "initiatorName": "iqn.1993-08.org.debian:01:c84ffd71216",
        "volumeAccessGroups": [
          1
        ]
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListQoSPolicies

您可以使用 ListQoSPolicies 方法列出系统上所有 QoS 策略的设置。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
qosPolicies	列出有关每个 QoS 策略的详细信息。	<i>QoS</i> Policy 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 231,
  "method": "ListQoS Policies",
  "params": {}
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 231,
  "result": {
    "qosPolicies": [
      {
        "name": "silver",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 14000,
          "minIOPS": 50
        },
        "qosPolicyID": 1,
        "volumeIDs": [
          1
        ]
      },
      {
        "name": "bronze",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 15000,
          "minIOPS": 50
        }
      }
    ]
  }
}
```

```
    },  
    "qosPolicyID": 2,  
    "volumeIDs": [  
      2  
    ]  
  }  
]  
}
```

发布以下版本后新增

10.0

ListSyncJobs

您可以使用 `ListSyncJobs` 方法获取有关在 **Element** 存储集群上运行的同步作业的信息。此方法可返回有关分区、克隆、块和远程同步作业的信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
syncJobs	描述当前正在系统中运行的同步进程的对象列表。	<i>syncJob</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "ListSyncJobs",  
  "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":{  
    "syncJobs":[  
      {  
        "bytesPerSecond":275314.8834458956,  
        "currentBytes":178257920,  
        "dstServiceID":36,  
        "elapsedTime":289.4568382049871,  
        "percentComplete":8.900523560209423,  
        "remainingTime":2962.675921065957,  
        "sliceID":5,  
        "srcServiceID":16,  
        "stage":"whole",  
        "totalBytes":200278016remote0,  
        "type":"slice"  
      }  
    ]  
  }  
}
```

```

    },
    {
      "bytesPerSecond":305461.3198607744,
      "cloneID":1,
      "currentBytes":81788928,
      "dstServiceID":16,
      "dstVolumeID":6,
      "elapsedTime":291.7847648200743,
      "nodeID":1,
      "percentComplete":8.167539267015707,
      "remainingTime":3280.708270981153,
      "sliceID":6,
      "srcServiceID":16,
      "srcVolumeID":5,
      "stage":"whole",
      "totalBytes":1001390080,
      "type":"clone"
    },
    {
      "blocksPerSecond":0,
      "branchType": "snapshot",
      "dstServiceID":8,
      "dstVolumeID":2,
      "elapsedTime":0,
      "percentComplete":0,
      "remainingTime":0,
      "sliceID":2,
      "stage":"metadata",
      "type":"remote"
    }
  ]
}

```

发布以下版本后新增

9.6

ListVolumeQoSHistograms

您可以使用 ListVolumeQoSHistograms 方法生成一个或多个卷的卷 QoS 使用情况直方图。这样可以帮助您更好地了解卷使用 QoS 的方式。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
volumeIDs	一个可选的卷列表，用于指定哪些卷应生成 QoS 直方图。	整型数组	无	否

返回值

此方法具有以下返回值:

名称	说明	类型
qosHistograms	一个对象列表，用于说明一个或多个卷的卷使用情况。	JSON 对象数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVolumeQoSHistograms",
  "params": {
    "volumeIDs": [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": {
    "qosHistograms": [
      {
        "histograms": {
          "belowMinIopsPercentages": {
            "Bucket1To19": 2406,
            "Bucket20To39": 3,
            "Bucket40To59": 0,
            "Bucket60To79": 4,
            "Bucket80To100": 0
          },
          "minToMaxIopsPercentages": {
            "Bucket101Plus": 0,
            "Bucket1To19": 0,
            "Bucket20To39": 0,
            "Bucket40To59": 2,
            "Bucket60To79": 0,
            "Bucket80To100": 0
          },
          "readBlockSizes": {
            "Bucket131072Plus": 0,
            "Bucket16384To32767": 0,
            "Bucket32768To65535": 0,
            "Bucket4096To8191": 0,
            "Bucket65536To131071": 0,
            "Bucket8192To16383": 0
          },
          "targetUtilizationPercentages": {
            "Bucket0": 134943,
            "Bucket101Plus": 0,
            "Bucket1To19": 2409,
            "Bucket20To39": 4,
            "Bucket40To59": 0,
            "Bucket60To79": 2,
            "Bucket80To100": 0
          },
          "throttlePercentages": {
            "Bucket0": 137358,
            "Bucket1To19": 0,
            "Bucket20To39": 0,
            "Bucket40To59": 0,
            "Bucket60To79": 0,

```

```

        "Bucket80To100": 0
    },
    "writeBlockSizes": {
        "Bucket131072Plus": 0,
        "Bucket16384To32767": 0,
        "Bucket32768To65535": 0,
        "Bucket4096To8191": 0,
        "Bucket65536To131071": 0,
        "Bucket8192To16383": 0
    }
},
"timestamp": "2018-06-21T18:45:52.010844Z",
"volumeID": 1
}
]
}
}

```

ListVolumes

您可以使用 `ListVolumes` 方法获取集群中所含卷的列表。可以使用可用参数来指定要在列表中返回的卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
accounts	只会返回由此处指定的帐户拥有的卷。与 <code>volumeIDs</code> 参数不能共存。	整型数组	无	否
includeVirtualVolumes	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否
isPaired	返回已配对或未配对的卷。可能值： <ul style="list-style-type: none"> <code>true</code>：返回所有已配对的卷。 <code>false</code>：返回所有未配对的卷。 	布尔型	无	否
limit	用于设置返回结果中包含的最大卷数。与 <code>volumeIDs</code> 参数不能共存。	整型	10000	否
startVolumeID	只会返回其 ID 大于或等于此值的卷。与 <code>volumeIDs</code> 参数不能共存。	整型	无	否
volumeIDs	卷 ID 列表。如果指定此参数，则其他参数仅对这组卷起作用。与 <code>accounts</code> 、 <code>startVolumeID</code> 和 <code>limit</code> 参数不能共存。	整型数组	否	否

名称	说明	类型	默认值	必需
volumeName	只会返回与卷名称匹配的卷对象信息。	字符串	否	否
volumeStatus	只会返回其状态等于状态值的卷。可能值： <ul style="list-style-type: none"> creating snapshotting active deleted 	字符串	否	否

返回值

此方法具有以下返回值：

名称	说明	类型
volumes	卷的列表。	<i>volume</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVolumes",
  "params": {
    "volumeIDs": [1],
    "volumeStatus": "active",
    "isPaired": "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": {
    "volumes": [
      {
        "access": "readWrite",
        "accountID": 1,
        "attributes": {},
        "blockSize": 4096,
        "createTime": "2016-03-28T14:39:05Z",
        "deleteTime": "",
        "enable512e": true,
        "iqn": "iqn.2010-01.com.solidfire:testvolume1.1",
        "name": "testVolume1",
        "purgeTime": "",
        "qos": {
          "burstIOPS": 15000,
          "burstTime": 60,
          "curve": {
            "4096": 100,
            "8192": 160,
```

```

        "16384": 270,
        "32768": 500,
        "65536": 1000,
        "131072": 1950,
        "262144": 3900,
        "524288": 7600,
        "1048576": 15000
    },
    "maxIOPS": 15000,
    "minIOPS": 50
  },
  "scsiEUIDeviceID": "6a79617900000001f47acc0100000000",
  "scsiNAADeviceID": "6f47acc1000000006a79617900000001",
  "sliceCount": 1,
  "status": "active",
  "totalSize": 5000658944,
  "virtualVolumeID": null,
  "volumeAccessGroups": [],
  "volumeID": 1,
  "volumePairs": []
}
]
}
}

```

发布以下版本后新增

9.6

ListVolumeStats

您可以使用 `ListVolumeStats` 方法获取单个卷、卷列表或所有卷（如果省略 `volumeIDs` 参数）的活动测量结果概要。测量值是在卷创建过程中累计的值。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>includeVirtualVolumes</code>	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否
<code>volumeIDs</code>	要从中检索活动信息的卷列表。	整型数组	否	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumeStats</code>	卷活动信息的列表。	volumeStats 数组

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListVolumeStats",
  "params": {

```

```
        "volumeIDs": [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": {
    "volumeStats": [
      {
        "accountID": 1,
        "actualIOPS": 0,
        "asyncDelay": null,
        "averageIOPSize": 0,
        "burstIOPSCredit": 30000,
        "clientQueueDepth": 0,
        "desiredMetadataHosts": null,
        "latencyUsec": 0,
        "metadataHosts": {
          "deadSecondaries": [],
          "liveSecondaries": [
            47
          ],
          "primary": 33
        },
        "nonZeroBlocks": 22080699,
        "readBytes": 657262370816,
        "readBytesLastSample": 0,
        "readLatencyUsec": 0,
        "readOps": 1604644446,
        "readOpsLastSample": 0,
        "samplePeriodMsec": 500,
        "throttle": 0,
        "timestamp": "2016-03-09T19:39:15.771697Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [
          1
        ],
        "volumeID": 1,
        "volumeSize": 107374182400,
        "volumeUtilization": 0,
        "writeBytes": 219117547520,
        "writeBytesLastSample": 0,
        "writeLatencyUsec": 0,
        "writeOps": 53495495,
        "writeOpsLastSample": 0,
        "zeroBlocks": 4133701
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListVolumeAccessGroups

您可以使用 `ListVolumeAccessGroups` 方法获取系统中当前所含卷访问组的相关消息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
limit	要返回的 <code>volumeAccessGroup</code> 对象的最大数量。与 <code>volumeAccessGroups</code> 参数不能共存。	整型	无限制	否
startVolumeAccessGroupID	列表开头处的卷访问组 ID。与 <code>volumeAccessGroups</code> 参数不能共存。	整型	0	否
volumeAccessGroups	要检索的卷访问组 ID 值列表。不能与 <code>startVolumeAccessGroupID</code> 和 <code>limit</code> 参数共存。	整型数组	[]	否

返回值

此方法具有以下返回值：

名称	说明	类型
volumeAccessGroups	描述每个卷访问组的对象列表。	<code>volumeAccessGroup</code> 数组
volumeAccessGroupsNotFound	系统未找到的卷访问组列表。如果使用了 <code>volumeAccessGroups</code> 参数，但系统找不到指定的一个或多个卷访问组，则显示此值。	整型数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVolumeAccessGroups",
  "params": {
    "startVolumeAccessGroupID": 3,
    "limit" : 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": {
    "volumeAccessGroups": [
      {
```

```

        "attributes": {},
        "deletedVolumes": [],
        "initiatorIDs": [],
        "initiators": [],
        "name": "example1",
        "volumeAccessGroupID": 3,
        "volumes": []
    }
  ]
}
}

```

发布以下版本后新增

9.6

ListVolumesForAccount

您可以使用 `ListVolumesForAccount` 方法列出帐户的活动卷和（待定）已删除卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>includeVirtualVolumes</code>	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否
<code>accountID</code>	返回该帐户 ID 拥有的所有卷。	整型	否	是

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumes</code>	卷信息列表。	<i>volume</i> 数组

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListVolumesForAccount",
  "params": {
    "accountID" : 1
  },
  "id" : 1
}

```

响应示例

此方法的响应类似于以下示例：

```

{
  "id": 1,
  "result": {
    "volumes": [
      {

```

```

    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2018-07-22T16:15:25Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.test1.25",
    "name": "test1",
    "purgeTime": "",
    "qos": {
      "burstIOPS": 15000,
      "burstTime": 60,
      "curve": {
        "4096": 100,
        "8192": 160,
        "16384": 270,
        "32768": 500,
        "65536": 1000,
        "131072": 1950,
        "262144": 3900,
        "524288": 7600,
        "1048576": 15000
      },
      "maxIOPS": 15000,
      "minIOPS": 50
    },
    "scsiEUIDeviceID": "306f746f00000019f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000019",
    "sliceCount": 1,
    "status": "active",
    "totalSize": 1000341504,
    "virtualVolumeID": null,
    "volumeAccessGroups": [],
    "volumeID": 25,
    "volumePairs": []
  }
}
]
}

```

发布以下版本后新增

9.6

ListVolumeStatsByAccount

您可以使用 `ListVolumeStatsByAccount` 方法列出每个帐户的卷活动测量结果概要。其中的值为该帐户下所有卷的值之和。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>includeVirtualVolumes</code>	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否

名称	说明	类型	默认值	必需
accounts	要返回卷统计信息的帐户 ID 列表。如果省略，则会返回所有帐户的统计信息。	整型数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
volumeStats	列出每个帐户的卷活动信息。 注： 每个条目的 volumeID 成员均为 0，因为值表示该帐户拥有的所有卷的总和。	<i>volumeStats</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVolumeStatsByAccount",
  "params": {"accounts": [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": {
    "volumeStats": [
      {
        "accountID": 3,
        "nonZeroBlocks": 155040175,
        "readBytes": 3156273328128,
        "readBytesLastSample": 0,
        "readOps": 770574543,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "timestamp": "2016-10-17T20:42:26.231661Z",
        "unalignedReads": 0,
        "unalignedWrites": 0,
        "volumeAccessGroups": [],
        "volumeID": 0,
        "volumeSize": 1127428915200,
        "writeBytes": 1051988406272,
        "writeBytesLastSample": 0,
        "writeOps": 256833107,
        "writeOpsLastSample": 0,
        "zeroBlocks": 120211025
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListVolumeStatsByVirtualVolume

您可以使用 `ListVolumeStatsByVirtualVolume` 方法列出系统中与虚拟卷关联的任何卷的卷统计信息。统计信息从创建卷之后开始累计。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
virtualVolumeIDs	要检索其信息的一个或多个虚拟卷 ID 列表。如果指定此参数，则该方法仅返回这些虚拟卷的相关信息。	UUID 字符串数组	否	否

返回值

此方法具有以下返回值：

名称	说明	类型
volumeStats	包含系统中所有虚拟卷的活动信息的对象列表。	<i>volumeStats</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVolumeStatsByVirtualVolume",
  "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": {
    "volumeStats": [
      {
        "accountID": 17,
        "actualIOPS": 0,
        "asyncDelay": null,
        "averageIOPSize": 1074265444,
        "burstIOPSCredit": 0,
        "clientQueueDepth": 0,
        "desiredMetadataHosts": null,
        "latencyUsec": 0,
        "metadataHosts": {
          "deadSecondaries": [],
          "liveSecondaries": [
            26
          ]
        }
      }
    ]
  }
}
```

```

        "primary": 56
      },
      "nonZeroBlocks": 36,
      "readBytes": 18366464,
      "readBytesLastSample": 0,
      "readLatencyUsec": 0,
      "readOps": 156,
      "readOpsLastSample": 0,
      "samplePeriodMsec": 500,
      "throttle": 0,
      "timestamp": "2016-10-10T17:46:35.914642Z",
      "unalignedReads": 156,
      "unalignedWrites": 185,
      "virtualVolumeID": "070ac0ba-f344-4f4c-b79c-142efa3642e8",
      "volumeAccessGroups": [],
      "volumeID": 12518,
      "volumeSize": 91271200768,
      "volumeUtilization": 0,
      "writeBytes": 23652213248,
      "writeBytesLastSample": 0,
      "writeLatencyUsec": 0,
      "writeOps": 185,
      "writeOpsLastSample": 0,
      "zeroBlocks": 22282972
    }
  ]
}

```

发布以下版本后新增

9.6

ListVolumeStatsByVolume

您可以使用 `ListVolumeStatsByVolume` 方法按卷列出每个卷的活动测量结果概要。具体值是在卷创建过程中累计的值。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>includeVirtualVolumes</code>	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumeStats</code>	卷活动信息的列表。	volumeStats 数组

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListVolumeStatsByVolume",

```

```
    "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": {  
    "volumeStats": [  
      {  
        "accountID": 3,  
        "actualIOPS": 0,  
        "asyncDelay": null,  
        "averageIOPSize": 4096,  
        "burstIOPSCredit": 30000,  
        "clientQueueDepth": 0,  
        "desiredMetadataHosts": null,  
        "latencyUsec": 0,  
        "metadataHosts": {  
          "deadSecondaries": [],  
          "liveSecondaries": [  
            16  
          ],  
          "primary": 12  
        },  
        "nonZeroBlocks": 7499205,  
        "readBytes": 159012818944,  
        "readBytesLastSample": 0,  
        "readLatencyUsec": 0,  
        "readOps": 38821489,  
        "readOpsLastSample": 0,  
        "samplePeriodMsec": 500,  
        "throttle": 0,  
        "timestamp": "2016-10-17T20:55:31.087537Z",  
        "unalignedReads": 0,  
        "unalignedWrites": 0,  
        "volumeAccessGroups": [  
          1  
        ],  
        "volumeID": 1,  
        "volumeSize": 53687091200,  
        "volumeUtilization": 0,  
        "writeBytes": 52992585728,  
        "writeBytesLastSample": 0,  
        "writeLatencyUsec": 0,  
        "writeOps": 12937643,  
        "writeOpsLastSample": 0,  
        "zeroBlocks": 5607995  
      }  
    ]  
  }  
}
```

发布以下版本后新增

9.6

ListVolumeStatsByVolumeAccessGroup

您可以使用 `ListVolumeStatsByVolumeAccessGroup` 方法列出指定卷访问组中所有成员卷的活动测量结果总计。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>includeVirtualVolumes</code>	默认情况下，响应中包含虚拟卷。要排除虚拟卷，请设置为 <code>false</code> 。	布尔型	<code>true</code>	否
<code>volumeAccessGroups</code>	返回其卷活动的卷访问组 ID 的数组。如果省略，则会返回所有卷访问组的统计信息。	整型数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumeStats</code>	列出指定的卷访问组中所有卷的卷活动信息。 注： 每个条目的 <code>volumeID</code> 成员均为 0，因为值表示该帐户拥有的所有卷的总和。	<i>volumeStats</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVolumeStatsByVolumeAccessGroup",
  "params": {"volumeAccessGroups": [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": {
    "volumeStats": [
      {
        "accountID": 0,
        "nonZeroBlocks": 149366393,
        "readBytes": 3156273328128,
        "readBytesLastSample": 0,
        "readOps": 770574543,
        "readOpsLastSample": 0,
        "samplePeriodMSec": 500,
        "timestamp": "2016-10-17T21:04:10.712370Z",
        "unalignedReads": 0,

```

```
        "unalignedWrites": 0,
        "volumeAccessGroups": [
            1
        ],
        "volumeID": 0,
        "volumeSize": 1073741824000,
        "writeBytes": 1051988406272,
        "writeBytesLastSample": 0,
        "writeOps": 256833107,
        "writeOpsLastSample": 0,
        "zeroBlocks": 112777607
    }
}
]
```

发布以下版本后新增

9.6

ModifyBackupTarget

您可以使用 `ModifyBackupTarget` 方法更改备份目标的属性。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
backupTargetID	要修改的目标的唯一目标 ID。	整型	无	是
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
name	备份目标的新名称。	字符串	无	否

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyBackupTarget",
  "params": {
    "backupTargetID": 1,
    "name": "yourtargetS3",
    "attributes": {
      "size": 500,
    }
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

ModifyInitiators

您可以使用 `ModifyInitiators` 方法更改一个或多个现有启动程序的属性。

您不能更改现有启动程序的名称。如果需要更改启动程序的名称，请先使用 [DeleteInitiators](#) 方法将其删除，然后使用 [CreateInitiators](#) 方法创建一个新的启动程序。

如果 `ModifyInitiators` 无法更改此参数中提供的启动程序之一，则此方法将返回错误且不会修改任何启动程序（无法执行部分操作）。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
initiators	<p>包含要修改的每个启动程序的特性的对象列表。可能对象：</p> <ul style="list-style-type: none"> • alias: (可选) 要分配给该启动程序的新友好名称。(字符串) • attributes: (可选) 要分配给该启动程序的一组新 JSON 属性。(JSON 对象) • chapUsername: (可选) 此启动程序的新唯一 CHAP 用户名。(字符串) • forceDuringUpgrade: 升级期间完成启动程序修改。 • initiatorID: (必需) 要修改的启动程序的 ID。(整型) • initiatorSecret: (可选) 用于对此启动程序进行身份验证的新 CHAP 密钥。(字符串) • requireChap: (可选) 如果需要对此启动程序使用 CHAP, 则为 True。(布尔值) • targetSecret: (可选) 用于对目标进行身份验证的新 CHAP 密钥(使用相互 CHAP 身份验证时)。(字符串) • volumeAccessGroupID: (可选) 该启动程序应添加到的卷访问组的 ID。如果该启动程序先前位于不同的卷访问组中, 则会从旧卷访问组中将其删除。如果此密钥存在但为 null, 则会将该启动程序从其当前卷访问组中删除, 但不放置在任何新卷访问组中。(整型) 	JSON 对象数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
initiators	描述新修改的启动程序的对象列表。	<i>initiator</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 6683,
  "method": "ModifyInitiators",
  "params": {
```

```
"initiators": [
  {
    "initiatorID": 2,
    "alias": "alias1",
    "volumeAccessGroupID": null
  },
  {
    "initiatorID": 3,
    "alias": "alias2",
    "volumeAccessGroupID": 1
  }
]
```

响应示例

此方法返回类似于以下示例的响应:

```
{
  "id": 6683,
  "result": {
    "initiators": [
      {
        "alias": "alias1",
        "attributes": {},
        "initiatorID": 2,
        "initiatorName": "iqn.1993-08.org.debian:01:395543635",
        "volumeAccessGroups": []
      },
      {
        "alias": "alias2",
        "attributes": {},
        "initiatorID": 3,
        "initiatorName": "iqn.1993-08.org.debian:01:935573135",
        "volumeAccessGroups": [
          1
        ]
      }
    ]
  }
}
```

发布以下版本后新增

9.6

相关参考

[CreateInitiators](#) on page 373

您可以使用 `CreateInitiators` 创建多个新启动程序 IQN 或全球通用端口名称 (World Wide Port Name, WWPN)，也可以为其分配别名和属性。使用 `CreateInitiators` 创建新启动程序时，也可以将这些启动程序添加到卷访问组中。

[DeleteInitiators](#) on page 375

您可以使用 `DeleteInitiators` 从系统（以及任何关联的卷或卷访问组）中删除一个或多个启动程序。

ModifyQoSPolicy

您可以使用 `ModifyQoSPolicy` 方法修改系统上的现有 QoS 策略。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
qosPolicyID	要修改的策略的 ID。	整型	无	是
name	如果提供了此参数，则 QoS 策略的名称（例如 <code>gold</code> 、 <code>platinum</code> 、 <code>silver</code> ）将更改为此值。	字符串	无	否
qos	如果提供，则此策略的 QoS 设置将更改为这些设置。您可以提供部分 QoS 值并且只更改一些 QoS 设置。	QoS 对象	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
qosPolicy	新修改的 QoS 策略的详细信息。	QoSPolicy

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 1950,
  "method": "ModifyQoSPolicy",
  "params": {
    "qosPolicyID": 2,
    "qos": {
      "minIOPS": 51,
      "maxIOPS": 15002,
      "burstIOPS": 15002
    }
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1950,
  "result": {
```

```
    "qosPolicy": {  
      "name": "bronze",  
      "qos": {  
        "burstIOPS": 15002,  
        "burstTime": 60,  
        "curve": {  
          "4096": 100,  
          "8192": 160,  
          "16384": 270,  
          "32768": 500,  
          "65536": 1000,  
          "131072": 1950,  
          "262144": 3900,  
          "524288": 7600,  
          "1048576": 15000  
        },  
        "maxIOPS": 15002,  
        "minIOPS": 51  
      },  
      "qosPolicyID": 2,  
      "volumeIDs": [  
        2  
      ]  
    }  
  }  
}
```

发布以下版本后新增

10.0

ModifyVolume

您可以使用 `ModifyVolume` 方法修改现有卷上的设置。一次只能对一个卷进行修改，并且更改将立即生效。

如果在修改卷时未指定 QoS 值，则在修改卷之后，这些值保持不变。可以通过运行 `GetDefaultQoS` 方法来检索新创建的卷的默认 QoS 值。

如果需要增加所复制的卷的大小，请按以下顺序执行此操作以防止发生复制错误：

1. 增加具有 `replicationTarget` 访问权限的卷的大小。
2. 增加源卷或具有 `readWrite` 访问权限的卷的大小。

请确保目标卷和源卷的大小相同。

注：如果将 `access` 状态更改为 `locked` 或 `replicationTarget`，则所有现有 iSCSI 连接将终止。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	要修改的卷的卷 ID。	整型	无	是

名称	说明	类型	默认值	必需
access	<p>允许对卷执行的访问。可能值：</p> <ul style="list-style-type: none"> • <code>readOnly</code>: 仅允许执行读取操作。 • <code>readWrite</code>: 允许执行读取和写入。 • <code>locked</code>: 不允许执行任何读取或写入。如果未指定，则访问值不会更改。 • <code>replicationTarget</code>: 将一个卷标识为一组配对卷的目标卷。如果卷未进行配对，则访问状态为 <code>locked</code>。如果未指定值，则访问值不会更改。 	字符串	无	否
accountID	<p>将卷重新分配到的帐户 ID。如果未指定任何帐户 ID，将使用上一个帐户名称。</p>	整型	无	否
associateWithQoSPolicy	<p>将卷与指定的 QoS 策略相关联。可能值：</p> <ul style="list-style-type: none"> • <code>true</code>: 将卷与 <code>QoSPolicyID</code> 参数中指定的 QoS 策略相关联。 • <code>false</code>: 不将卷与 <code>QoSPolicyID</code> 参数中指定的 QoS 策略相关联。如果为 <code>false</code>，无论是否在 <code>QoSPolicy</code> 参数中指定 QoS 策略，都将删除任何现有的策略关联。 	布尔型	无	否
attributes	<p>名称-值对的列表，采用 JSON 对象格式。</p>	JSON 对象	无	否
createTime	<p>要设置为新卷创建日期的 ISO 8601 日期字符串。必填（如果 <code>setCreateTime</code> 设置为 <code>true</code>）。</p>	ISO 8601 字符串	无	否
enableSnapMirrorReplication	<p>确定卷是否可用于 <code>SnapMirror</code> 端点的复制。可能值：</p> <ul style="list-style-type: none"> • <code>true</code> • <code>false</code> 	布尔型	<code>false</code>	否
mode	<p>卷复制模式。可能值：</p> <ul style="list-style-type: none"> • <code>asynch</code>: 等待系统确认数据存储在源上，然后再写入到目标。 • <code>sync</code>: 立即开始将数据写入到目标，而不等待来自源的数据传输确认。 	字符串	无	否

名称	说明	类型	默认值	必需
qos	此卷的服务质量设置。如果未指定，则 QoS 设置不会更改。可能值： <ul style="list-style-type: none"> minIOPS maxIOPS burstIOPS 	<i>QoS</i>	无	否
qosPolicyID	策略 ID，该策略中的 QoS 设置应该应用于指定卷。此参数与 qos 参数不能同时使用。	整型	无	否
setCreateTime	设置为 true 可更改所记录的卷创建日期。	布尔型	无	否
totalSize	卷的新大小（以字节为单位）。1000000000 等于 1 GB。大小将向上取整为最接近的 MB。此参数只能用于增加卷的大小。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
volume	包含新修改卷的相关信息的对象。	<i>volume</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyVolume",
  "params": {
    "volumeID": 5,
    "attributes": {
      "name1": "value1",
      "name2": "value2",
      "name3": "value3"
    },
    "qos": {
      "minIOPS": 60,
      "maxIOPS": 100,
      "burstIOPS": 150,
      "burstTime": 60
    },
    "access": "readWrite"
  },
  "totalSize": 20000000000,
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "volume": {
      "access": "readWrite",
      "accountID": 1,
      "attributes": {
        "name1": "value1",
        "name2": "value2",
        "name3": "value3"
      },
      "blockSize": 4096,
      "createTime": "2016-03-28T16:16:13Z",
      "deleteTime": "",
      "enable512e": true,
      "iqn": "iqn.2010-01.com.solidfire:jyay.1459181777648.5",
      "name": "1459181777648",
      "purgeTime": "",
      "qos": {
        "burstIOPS": 150,
        "burstTime": 60,
        "curve": {
          "4096": 100,
          "8192": 160,
          "16384": 270,
          "32768": 500,
          "65536": 1000,
          "131072": 1950,
          "262144": 3900,
          "524288": 7600,
          "1048576": 15000
        },
        "maxIOPS": 100,
        "minIOPS": 60
      },
      "scsiEUIDeviceID": "6a79617900000005f47acc0100000000",
      "scsiNAADeviceID": "6f47acc1000000006a79617900000005",
      "sliceCount": 1,
      "status": "active",
      "totalSize": 1000341504,
      "virtualVolumeID": null,
      "volumeAccessGroups": [
        1
      ],
      "volumeID": 5,
      "volumePairs": []
    }
  }
}
```

发布以下版本后新增

9.6

相关参考

[GetDefaultQoS](#) on page 383

您可以使用 `GetDefaultQoS` 方法获取新建卷的默认服务质量 (QoS) 值。

ModifyVolumes

您可以使用 `ModifyVolumes` 方法一次性配置最多 500 个现有卷。所做更改将立即生效。如果 `ModifyVolumes` 无法修改任一指定卷，则所有指定卷都不会发生更改。

如果在修改卷时未指定 QoS 值，则每个卷的 QoS 值保持不变。可以通过运行 `GetDefaultQoS` 方法来检索新创建的卷的默认 QoS 值。

如果需要增加所复制的卷的大小，请按以下顺序执行此操作以防止发生复制错误：

1. 增加具有 `replicationTarget` 访问权限的卷的大小。
2. 增加源卷或具有 `readWrite` 访问权限的卷的大小。

请确保目标卷和源卷的大小相同。

注：如果将 `access` 状态更改为 `locked` 或 `replicationTarget`，则所有现有 iSCSI 连接将终止。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>access</code>	<p>允许对卷执行的访问。可能值：</p> <ul style="list-style-type: none"> • <code>readOnly</code>：仅允许执行读取操作。 • <code>readWrite</code>：允许执行读取和写入。 • <code>locked</code>：不允许执行任何读取或写入。如果未指定，则访问值不会更改。 • <code>replicationTarget</code>：将一个卷标识为一组配对卷的目标卷。如果卷未进行配对，则访问状态为 <code>locked</code>。如果未指定值，则访问值不会更改。 	字符串	无	否
<code>accountID</code>	<p>将卷重新分配到的帐户 ID。如果未指定任何帐户 ID，将使用上一个帐户名称。</p>	整型	无	否

名称	说明	类型	默认值	必需
associateWithQoSPolicy	<p>将卷与指定的 QoS 策略相关联。可能值：</p> <ul style="list-style-type: none"> • <code>true</code>：将卷与 <code>QoSPolicyID</code> 参数中指定的 QoS 策略相关联。 • <code>false</code>：不将卷与 <code>QoSPolicyID</code> 参数中指定的 QoS 策略相关联。如果为 <code>false</code>，无论是否在 <code>QoSPolicy</code> 参数中指定 QoS 策略，都将删除任何现有的策略关联。 	布尔型	无	否
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
createTime	要设置为新卷创建日期的 ISO 8601 日期字符串。必填（如果 <code>setCreateTime</code> 设置为 <code>true</code> ）。	ISO 8601 字符串	无	否
enableSnapMirrorReplication	<p>确定卷是否可用于 <code>SnapMirror</code> 端点的复制。可能值：</p> <ul style="list-style-type: none"> • <code>true</code> • <code>false</code> 	布尔型	<code>false</code>	否
mode	<p>卷复制模式。可能值：</p> <ul style="list-style-type: none"> • <code>asynch</code>：等待系统确认数据存储在源上，然后再写入到目标。 • <code>sync</code>：立即开始将数据写入到目标，而不等待来自源的数据传输确认。 	字符串	无	否
qos	<p>卷的新服务质量设置。如果未指定，则 QoS 设置不会更改。可能值：</p> <ul style="list-style-type: none"> • <code>minIOPS</code> • <code>maxIOPS</code> • <code>burstIOPS</code> 	QoS	无	否
qosPolicyID	策略 ID，该策略中的 QoS 设置应该应用于指定卷。此参数与 <code>qos</code> 参数不能同时使用。	整型	无	否
setCreateTime	设置为 <code>true</code> 可更改所记录的卷创建日期。	布尔型	无	否
totalSize	卷的新大小（以字节为单位）。1000000000 等于 1 GB。大小将向上取整为最接近的 MB。此参数只能用于增加卷的大小。	整型	无	否

名称	说明	类型	默认值	必需
volumeIDs	要修改的卷的卷 ID 列表。	整型数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
volume	包含每个新修改卷的相关信息的对象数组。	<i>volume</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyVolumes",
  "params": {
    "volumeIDs": [2,3],
    "attributes": {
      "name1": "value1",
      "name2": "value2",
      "name3": "value3"
    },
    "qos": {
      "minIOPS": 50,
      "maxIOPS": 100,
      "burstIOPS": 150,
      "burstTime": 60
    },
    "access": "replicationTarget"
  },
  "totalSize": 80000000000,
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "volumes": [
      {
        "access": "replicationTarget",
        "accountID": 1,
        "attributes": {
          "name1": "value1",
          "name2": "value2",
          "name3": "value3"
        },
        "blockSize": 4096,
        "createTime": "2016-04-06T17:25:13Z",
        "deleteTime": "",
        "enable512e": false,
        "iqn": "iqn.2010-01.com.solidfire:jo73.2",
        "name": "doctest1",
        "purgeTime": ""
      }
    ]
  }
}
```

```
"qos": {
  "burstIOPS": 150,
  "burstTime": 60,
  "curve": {
    "4096": 100,
    "8192": 160,
    "16384": 270,
    "32768": 500,
    "65536": 1000,
    "131072": 1950,
    "262144": 3900,
    "524288": 7600,
    "1048576": 15000
  },
  "maxIOPS": 100,
  "minIOPS": 50
},
"scsiEUIDeviceID": "6a6f373300000002f47acc0100000000",
"scsiNAADeviceID": "6f47acc1000000006a6f373300000002",
"sliceCount": 1,
"status": "active",
"totalSize": 1000341504,
"virtualVolumeID": null,
"volumeAccessGroups": [],
"volumeID": 2,
"volumePairs": []
},
{
  "access": "replicationTarget",
  "accountID": 1,
  "attributes": {
    "name1": "value1",
    "name2": "value2",
    "name3": "value3"
  },
  "blockSize": 4096,
  "createTime": "2016-04-06T17:26:31Z",
  "deleteTime": "",
  "enable512e": false,
  "iqn": "iqn.2010-01.com.solidfire:jo73.3",
  "name": "doctest2",
  "purgeTime": "",
  "qos": {
    "burstIOPS": 150,
    "burstTime": 60,
    "curve": {
      "4096": 100,
      "8192": 160,
      "16384": 270,
      "32768": 500,
      "65536": 1000,
      "131072": 1950,
      "262144": 3900,
      "524288": 7600,
      "1048576": 15000
    },
    "maxIOPS": 100,
    "minIOPS": 50
  },
  "scsiEUIDeviceID": "6a6f373300000003f47acc0100000000",
```

```

        "scsiNAADeviceID": "6f47acc1000000006a6f373300000003",
        "sliceCount": 1,
        "status": "active",
        "totalSize": 1000341504,
        "virtualVolumeID": null,
        "volumeAccessGroups": [],
        "volumeID": 3,
        "volumePairs": []
    }
  ]
}
}

```

发布以下版本后新增

9.6

相关参考

[GetDefaultQoS](#) on page 383

您可以使用 `GetDefaultQoS` 方法获取新建卷的默认服务质量 (QoS) 值。

ModifyVolumeAccessGroup

您可以使用 `ModifyVolumeAccessGroup` 方法更新启动程序以及在卷访问组中添加或删除卷。

如果指定的启动程序或卷与现有启动程序或卷重复，则卷访问组保留不变。如果您未指定卷或启动程序的值，则当前启动程序和卷列表不会发生变化。

您也可以使卷访问组与某个虚拟网络关联起来。这种关联将应用于卷访问组中的所有卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>volumeAccessGroupID</code>	要修改的卷访问组的 ID。	整型	无	是
<code>name</code>	此卷访问组的新名称。	字符串	无	否
<code>attributes</code>	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
<code>initiators</code>	列出要包括在卷访问组中的启动程序 ID 或名称 (IQN 和 WWPN)。如果传递启动程序名称列表，但不存在相关启动程序，则会创建这些启动程序。如果传递启动程序 ID 列表，但不存在任何启动程序，则该方法将返回错误。传递启动程序名称已弃用；您应尽可能使用启动程序 ID。	整型数组（建议）或字符串数组（已弃用）	无	否

名称	说明	类型	默认值	必需
deleteOrphanInitiators	指定是否在启动程序对象从卷访问组中删除之后删除这些对象。可能值： <ul style="list-style-type: none"> • true: 在启动程序对象从卷访问组中删除之后，删除这些对象。 • false: 在启动程序对象从卷访问组中删除之后，不删除这些对象。这是默认值。 	布尔型	false	否
volumes	要修改的卷的卷 ID 列表。	整型数组	无	<i>volumeAccessGroup</i>

返回值

此方法具有以下返回值：

名称	说明	类型
volumeAccessGroup	包含新修改的卷访问组相关信息的对象。	<i>volumeAccessGroup</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyVolumeAccessGroup",
  "params": {
    "volumeAccessGroupID": 96,
    "name": "accessgrouptest",
    "initiators": [115,114],
    "volumes": [
      346
    ],
    "attributes": {}
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": null,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        327
      ],
      "initiatorIDs": [
        114,
        115
      ],
      "initiators": [
```

```
        "iqn.1998-01.com.vmware:desk1-esx1-577b283a",  
        "iqn.1998-01.com.vmware:donesq-esx1-421b281b"  
    ],  
    "name": "accessgrouptest",  
    "volumeAccessGroupID": 96,  
    "volumes": [  
        346  
    ]  
  }  
}  
}
```

发布以下版本后新增

9.6

相关参考

[AddInitiatorsToVolumeAccessGroup](#) on page 355

您可以使用 `AddInitiatorsToVolumeAccessGroup` 方法将启动程序添加到指定的卷访问组。

[AddVolumesToVolumeAccessGroup](#) on page 356

您可以使用 `AddVolumesToVolumeAccessGroup` 方法将卷添加到指定的卷访问组。

[RemoveInitiatorsFromVolumeAccessGroup](#) on page 429

您可以使用 `RemoveInitiatorsFromVolumeAccessGroup` 方法从指定的卷访问组中删除启动程序。

[RemoveVolumesFromVolumeAccessGroup](#) on page 430

您可以使用 `RemoveVolumesFromVolumeAccessGroup` 方法从指定的卷访问组中删除卷。

PurgeDeletedVolume

您可以使用 `PurgeDeletedVolume` 方法立即永久清除已删除的卷。您必须先使用 `DeleteVolume` 删除卷，然后才能清除此卷。

系统会在一段时间后自动清除卷，因此通常不需要使用此方法。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	要清除的卷的卷 ID。	整型	否	是

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": "PurgeDeletedVolume",  
  "params": {  
    "volumeID" : 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": {}  
}
```

发布以下版本后新增

9.6

相关参考

[Delete Volume](#) on page 376

您可以使用 `DeleteVolume` 方法将活动卷标记为待删除。标记后的卷会在清理间隔过后被清除（永久删除）。

PurgeDeletedVolumes

您可以使用 `PurgeDeletedVolumes` 方法立即永久清除已删除的卷；使用此方法可一次性清除最多 500 个卷。

您必须先使用 `DeleteVolumes` 删除卷，然后才能清除此卷。系统会在一段时间后自动清除卷，因此通常不需要使用此方法。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeIDs	要从系统中清除的卷的卷 ID 列表。	整型数组	否	否
accountIDs	帐户 ID 列表。所有指定帐户中的所有卷均会从系统中清除。	整型数组	否	否
volumeAccessGroupIDs	卷访问组 ID 的列表。所有指定卷访问组中的所有卷均会从系统中清除。	整型数组	否	否

注：您只能为每个方法调用指定上述参数之一。指定多个参数或者不指定任何参数均会导致出现错误。

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": "PurgeDeletedVolumes",  
  "params": {  
    "accountIDs" : [1, 2, 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": {}  
}
```

发布以下版本后新增

9.6

相关参考

[Delete Volumes](#) on page 378

您可以使用 DeleteVolumes 方法将多个（最多 500 个）活动卷标记为待删除。标记后的卷会在清理间隔过后被清除（永久删除）。

RemoveBackupTarget

您可以使用 RemoveBackupTarget 方法删除备份目标。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
backupTargetID	要删除的目标的唯一目标 ID。	整型	无	是

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": "RemoveBackupTarget",  
  "params": {  
    "backupTargetID" : 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": {}  
}
```

发布以下版本后新增

9.6

RemoveInitiatorsFromVolumeAccessGroup

您可以使用 `RemoveInitiatorsFromVolumeAccessGroup` 方法从指定的卷访问组中删除启动程序。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>volumeAccessGroupID</code>	从中删除启动程序的卷访问组的 ID。	整型	无	是
<code>initiators</code>	列出要包括在卷访问组中的启动程序 ID 或名称（IQN 和 WWPN）。如果传递启动程序名称列表，但不存在相关启动程序，则会创建这些启动程序。如果传递启动程序 ID 列表，但不存在任何启动程序，则该方法将返回错误。传递启动程序名称已弃用；您应尽可能使用启动程序 ID。	整型数组（建议）或字符串数组（已弃用）	无	否
<code>deleteOrphanInitiators</code>	指定是否在启动程序对象从卷访问组中删除之后删除这些对象。可能值： <ul style="list-style-type: none"> <code>true</code>：在启动程序对象从卷访问组中删除之后，删除这些对象。 <code>false</code>：在启动程序对象从卷访问组中删除之后，不删除这些对象。这是默认值。 	布尔型	<code>false</code>	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>volumeAccessGroup</code>	包含新修改的卷访问组相关信息的对象。	<i><code>volumeAccessGroup</code></i>

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 13171,
```

```
"method": "RemoveInitiatorsFromVolumeAccessGroup",  
"params": {  
  "initiators": [114,115],  
  "volumeAccessGroupID": 96  
}  
}
```

响应示例

此方法返回类似于以下示例的响应:

```
{  
  "id": 13171,  
  "result": {  
    "volumeAccessGroup": {  
      "attributes": {},  
      "deletedVolumes": [  
        327  
      ],  
      "initiatorIDs": [],  
      "initiators": [],  
      "name": "test",  
      "volumeAccessGroupID": 96,  
      "volumes": [  
        346  
      ]  
    }  
  }  
}
```

发布以下版本后新增

9.6

RemoveVolumesFromVolumeAccessGroup

您可以使用 `RemoveVolumesFromVolumeAccessGroup` 方法从指定的卷访问组中删除卷。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
volumeAccessGroupID	要从中删除卷的卷访问组 ID。	整型	无	是
volumes	要从卷访问组中删除的卷的卷 ID。	整型数组	无	是

返回值

此方法具有以下返回值:

名称	说明	类型
volumeAccessGroup	包含新修改的卷访问组相关信息的对象。	<i>volumeAccessGroup</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "RemoveVolumesFromVolumeAccessGroup",
  "params": {
    "volumeAccessGroupID": 96,
    "volumes": [1,2]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "volumeAccessGroup": {
      "attributes": {},
      "deletedVolumes": [
        346
      ],
      "initiatorIDs": [
        116,
        117
      ],
      "initiators": [
        "iqn.1993-08.org.debian:01:181324777",
        "iqn.1993-08.org.debian:01:181324888"
      ],
      "name": "northbanktest",
      "volumeAccessGroupID": 96,
      "volumes": []
    }
  }
}
```

发布以下版本后新增

9.6

RestoreDeletedVolume

您可以使用 `RestoreDeletedVolume` 方法将已删除的卷重新标记为活动。此操作会使卷立即对 iSCSI 连接可用。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	要还原的已删除卷的卷 ID。	整型	无	是

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": "RestoreDeletedVolume",  
  "params": {  
    "volumeID" : 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": {}  
}
```

发布以下版本后新增

9.6

SetDefaultQoS

您可以使用 `SetDefaultQoS` 方法配置卷的默认服务质量 (Quality of Service, QoS) 值（以 IOPS 为单位）。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
minIOPS	集群为卷提供的最小可持续 IOPS。	整型	无	否
maxIOPS	集群为卷提供的最大可持续 IOPS。	整型	无	否
burstIOPS	在短时突发情形下允许的最大 IOPS 数。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
minIOPS	集群为卷提供的最小可持续 IOPS。	整型
maxIOPS	集群为卷提供的最大可持续 IOPS。	整型
burstIOPS	在短时突发情形下允许的最大 IOPS 数。	整型

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "SetDefaultQoS",
  "params": {
    "burstIOPS":8000,
    "maxIOPS":1000,
    "minIOPS":200
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id":1,
  "result": {
    "burstIOPS":8000,
    "maxIOPS":1000,
    "minIOPS":200
  }
}
```

发布以下版本后新增

9.6

StartBulkVolumeRead

您可以使用 `StartBulkVolumeRead` 方法在指定卷上启动批量卷读取会话。

一个卷上只能同时运行两个批量卷进程。初始化该会话后，将从 **SolidFire** 存储卷读取要存储在外部备份源上的数据。外部数据可通过 **Element** 存储节点上运行的 **Web** 服务器访问。外部数据访问的服务器交互信息由运行于存储系统上的脚本传递。

在批量卷读取操作开始时，将生成卷的快照，并且会在读取操作完成后删除该快照。您也可以通过输入快照 **ID** 作为参数来读取卷的快照。读取先前的快照时，系统不会创建卷的新快照，而且不会在读取操作完成后删除先前的快照。

注：如果未提供现有快照的 **ID**，则此过程会创建新快照。当集群填充度处于第 2 或第 3 阶段时，可以创建快照。当集群填充度处于第 4 或第 5 阶段时，不会创建快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
format	卷数据的格式。可以是下列任一项： <ul style="list-style-type: none"> uncompressed: 返回卷的每个字节，而不进行任何数据压缩。 native: 返回较小的不透明数据，并在后续的批量卷写入操作更高效地存储和写入不透明数据。 	字符串	无	是
volumeID	要读取的卷的 ID。	整型	无	是
snapshotID	先前为批量卷读取操作创建的快照的 ID。如果未输入 ID，则会指定当前的活动卷映像的快照。	整型	无	否
script	可执行脚本的名称。如果未指定脚本名称，则需要使用密钥和 URL 来访问 Element 存储节点。脚本将在主节点上运行，并且密钥和 URL 会返回到脚本中，以便可以与本地 Web 服务器联系。	字符串	无	否
scriptParameters	要传递到脚本的 JSON 参数。	JSON 对象	无	否
attributes	批量卷作业的 JSON 属性。	JSON 对象	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
asyncHandle	要检查其完成情况的异步进程的 ID。	整型
key	唯一标识会话的不透明密钥。	字符串
url	用于访问节点的 Web 服务器的 URL。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "StartBulkVolumeRead",
  "params": {
    "volumeID" : 5,
    "format" : "native",
    "snapshotID" : 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" : {
    "asyncHandle" : 1,
    "key" : "11eed8f086539205beeaadd981aad130",
    "url" : "https://127.0.0.1:44000/"
  }
}
```

发布以下版本后新增

9.6

StartBulkVolumeWrite

您可以使用 `StartBulkVolumeWrite` 方法在指定卷上启动批量卷写入会话。

一个卷上只能同时运行两个批量卷进程。初始化此会话后，外部备份源中的数据将写入 Element 存储卷。外部数据可通过 Element 存储节点上运行的 Web 服务器访问。外部数据访问的服务器交互信息由运行于存储系统上的脚本传递。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
format	卷数据的格式。可以是下列任一项： <ul style="list-style-type: none"> <code>uncompressed</code>：返回卷的每个字节，而不进行任何数据压缩。 <code>native</code>：返回较小的不透明数据，并在后续的批量卷写入操作更高效地存储和写入不透明数据。 	字符串	无	是
volumeID	要写入到的卷的 ID。	整型	无	是
script	可执行脚本的名称。如果未指定脚本名称，则需要使用密钥和 URL 来访问 Element 存储节点。脚本将在主节点上运行，并且密钥和 URL 会返回到脚本中，以便可以与本地 Web 服务器联系。	字符串	无	否
scriptParameters	要传递到脚本的 JSON 参数。	JSON 对象	无	否
attributes	批量卷作业的 JSON 属性。	JSON 对象	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
asyncHandle	要检查其完成情况的异步进程的 ID。	整型
key	唯一标识会话的不透明密钥。	字符串
url	用于访问节点的 Web 服务器的 URL。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "StartBulkVolumeWrite",
  "params": {
    "volumeID" : 5,
    "format" : "native",
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id" : 1,
  "result" : {
    "asyncHandle" : 1,
    "key" : "11eed8f086539205beeaadd981aad130",
    "url" : "https://127.0.0.1:44000/"
  }
}
```

发布以下版本后新增

9.6

UpdateBulkVolumeStatus

您可以使用 `UpdateBulkVolumeStatus` 方法更新通过 `StartBulkVolumeRead` 或 `StartBulkVolumeWrite` 方法启动的批量卷作业的状态。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
key	在初始化 <i>StartBulkVolumeRead</i> 或 <i>StartBulkVolumeWrite</i> 会话期间分配的密钥。	字符串	无	是
status	系统会设置给定批量卷作业的状态。可能值： <ul style="list-style-type: none"> running: 作业仍处于活动状态。 complete: 作业已完成。 failed: 作业已失败。 	字符串	无	是

名称	说明	类型	默认值	必需
percentComplete	批量卷作业的完成进度，以百分比表示。	字符串	无	否
message	在作业完成后返回批量卷作业的状态。	字符串	无	否
attributes	JSON 属性；更新批量卷作业的内容。	JSON 对象	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
status	所请求的会话的状态。返回的状态： <ul style="list-style-type: none"> • preparing • active • done • failed 	字符串
attributes	返回在方法调用中指定的属性。返回值指示属性是否已更改。	字符串
url	用于访问节点的 Web 服务器的 URL；仅在会话仍处于活动状态时提供。	字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "UpdateBulkVolumeStatus",
  "params": {
    "key": "0b2f532123225febda2625f55dcb0448",
    "status": "running"
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id" : 1,
  "result": {
    "status" : "running",
    "url" : "https://10.10.23.47:8443/"
  }
}
```

发布以下版本后新增

9.6

相关参考

[StartBulkVolumeRead](#) on page 433

您可以使用 `StartBulkVolumeRead` 方法在指定卷上启动批量卷读取会话。

[*StartBulkVolumeWrite*](#) on page 435

您可以使用 `StartBulkVolumeWrite` 方法在指定卷上启动批量卷写入会话。

卷快照 API 方法

您可以使用 Element 软件卷快照 API 方法管理卷快照。可以使用卷快照 API 方法创建、修改、克隆和删除卷快照。

快照概述

卷快照是卷的时间点副本。您可以使用快照将卷回滚到创建快照时的状态。

您可以将卷快照分组到一起，以便以一致的方式备份或回滚相关卷。组快照可捕获所有卷分区文件的时间点映像。随后，您可以使用该映像将一组卷回滚到某个时间点状态，并确保所有数据在组中的所有卷上是一致的。

您可以对卷快照创建进行计划，使其按定义的间隔自主进行。可以按时间、周中的某些天或月份中的某些天定义间隔。您还可以使用计划快照，以确保将快照备份到远程存储以用于归档。

CreateGroupSnapshot

您可以使用 CreateGroupSnapshot 创建一组卷的时间点副本。

您可以在以后使用该快照作为备份或回滚，以确保卷组上的数据在创建快照时的时间点是

注：当集群填充度处于第 1、2 或第 3 阶段时，您可以创建快照。当集群填充度达到第 4 或第 5 阶段时，您不能创建快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
enableRemoteReplication	指定快照是否会复制到远程存储。可能值： <ul style="list-style-type: none">• true: 快照会复制到远程存储。• false: 快照不会复制到远程存储。	布尔型	false	否
name	组快照的名称。如果未输入任何名称，则会使用生成组快照时的日期和时间。允许的最大名称长度为 255 个字符。	字符串	无	否
retention	快照的保留时间。如果留空，则快照会永久保留。请以 HH:mm:ss 格式输入值。	字符串	无	否
snapMirrorLabel	SnapMirror 软件用于在 SnapMirror 端点上指定快照保留策略的标签。	字符串	无	否

名称	说明	类型	默认值	必需
volumes	从中进行复制的卷映像的唯一 ID。	volumeID 数组	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
members	组中每个成员的 checksum、volumeIDs 和 snapshotIDs 的列表。有效值： <ul style="list-style-type: none"> checksum: 以短字符串表示已存储快照中的数据。此校验和以后可用于比较其他快照，以检测数据中存在的错误。（字符串） snapshotID: 从其生成新快照的快照的唯一 ID。快照 ID 必须来自给定卷上的快照。（整型） volumeID: 快照的源卷 ID。（整型） 	JSON 对象数组
groupSnapshotID	新的组快照的唯一 ID。	groupSnapshot ID
groupSnapshot	包含新创建组快照的相关信息的对象。	<i>groupSnapshot</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateGroupSnapshot",
  "params": {
    "volumes": [1,2]
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "groupSnapshot": {
      "attributes": {},
      "createTime": "2016-04-04T22:43:29Z",
      "groupSnapshotID": 45,
      "groupSnapshotUUID": "473b78a3-ef85-4541-9438-077306b2d3ca",
      "members": [
        {
          "attributes": {},

```

```
    "checksum": "0x0",
    "createTime": "2016-04-04T22:43:29Z",
    "enableRemoteReplication": false,
    "expirationReason": "None",
    "expirationTime": null,
    "groupID": 45,
    "groupSnapshotUUID": "473b78a3-ef85-4541-9438-077306b2d3ca",
    "name": "2016-04-04T22:43:29Z",
    "snapshotID": 3323,
    "snapshotUUID": "7599f200-0092-4b41-b362-c431551937d1",
    "status": "done",
    "totalSize": 5000658944,
    "virtualVolumeID": null,
    "volumeID": 1
  },
  {
    "attributes": {},
    "checksum": "0x0",
    "createTime": "2016-04-04T22:43:29Z",
    "enableRemoteReplication": false,
    "expirationReason": "None",
    "expirationTime": null,
    "groupID": 45,
    "groupSnapshotUUID": "473b78a3-ef85-4541-9438-077306b2d3ca",
    "name": "2016-04-04T22:43:29Z",
    "snapshotID": 3324,
    "snapshotUUID": "a0776a48-4142-451f-84a6-5315dc37911b",
    "status": "done",
    "totalSize": 6001000448,
    "virtualVolumeID": null,
    "volumeID": 2
  }
],
"name": "2016-04-04T22:43:29Z",
"status": "done"
},
"groupSnapshotID": 45,
"members": [
  {
    "checksum": "0x0",
    "snapshotID": 3323,
    "snapshotUUID": "7599f200-0092-4b41-b362-c431551937d1",
    "volumeID": 1
  },
  {
    "checksum": "0x0",
    "snapshotID": 3324,
    "snapshotUUID": "a0776a48-4142-451f-84a6-5315dc37911b",
    "volumeID": 2
  }
]
}
}
```

发布以下版本后新增

9.6

CreateSchedule

您可以使用 `CreateSchedule` 制定按定义的间隔自动生成卷快照的计划。

您可以在以后使用已创建的快照作为备份或回滚，以确保卷或卷组上的数据在创建快照时的时间点是一致的。如果计划按无法被 5 分钟整除的时间段运行快照，快照将在下一个可被 5 分钟整除的时间段运行。例如，如果计划在 12:42:00 UTC 运行快照，它将在 12:45:00 UTC 运行。您无法计划以低于 5 分钟的间隔运行快照。

注：当集群填充度处于第 1、2 或第 3 阶段时，您可以创建快照。当集群填充度达到第 4 或第 5 阶段时，您不能创建快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
attributes	使用“frequency”字符串以指示生成快照的频率。可能值： <ul style="list-style-type: none"> Days of Week Days of Month Time Interval 	JSON 对象	无	否
hours	两次重复生成快照相隔的小时数，或者在 Days of Week 或 Days of Month 模式下生成快照的 GMT 时间（小时）。有效值为 0 到 23。	整型	无	否
name	快照的名称。如果未输入任何名称，则会使用生成组快照时的日期和时间。允许的最大名称长度为 244 个字符。	字符串	无	否
minutes	两次重复生成快照相隔的分钟数，或者在 Days of Week 或 Days of Month 模式下生成快照的 GMT 时间（分钟）。有效值为 5 到 59。	整型	无	否
paused	指示是否应暂停该计划。有效值： <ul style="list-style-type: none"> true false 	布尔型	无	否
recurring	指示该计划是否重复运行。有效值为： <ul style="list-style-type: none"> true false 	布尔型	无	否
scheduleName	计划的唯一名称。允许的最大计划名称长度为 244 个字符。	字符串	无	是

名称	说明	类型	默认值	必需
scheduleType	指示要创建的计划的类型。有效值为 <code>snapshot</code> 。	字符串	无	是
scheduleInfo	有关如何在每个计划的间隔创建快照的信息的对象。 <ul style="list-style-type: none"> • <code>volumeID</code>: 要包括在快照中的卷的 ID。(整型) • <code>volumes</code>: 要包括在组快照中的卷的 ID 列表。(整型数组) • <code>name</code>: 要使用的快照名称。(字符串) • <code>enableRemoteReplication</code>: 指示是否应将快照包括在远程复制中。(布尔值) • <code>retention</code>: 快照的保留时间, 格式为 <code>HH:mm:ss</code>。(字符串) 	JSON 对象	无	是
snapMirrorLabel	SnapMirror 软件用于在 SnapMirror 端点上指定快照保留策略的标签。	字符串	无	否
startingDate	运行该计划之前经过的时间。如果未设置此参数, 计划将立即启动。采用 UTC 时间格式。	ISO 8601 日期字符串	无	否
monthdays	将在月份中的哪些天生成快照。有效值为 1 到 31。	整型数组	无	是 (如果计划在月份中的某些天生成快照)
weekdays	将在星期几创建快照。所需的值 (如果使用): <ul style="list-style-type: none"> • <code>Day</code>: 0 到 6 (星期日到星期六) • <code>偏移量</code>: 在一个月中的每一个可能的星期、1 到 6 (如果大于 1、则仅在一周的第 1 天匹配)。例如, <code>Offset:3</code> 表示星期日, 而 <code>Offset: 3</code> 表示月的第三个星期日, 而 <code>Offset: 4</code> 表示星期三的第四个星期三。偏移量: 0 表示未执行任何操作。偏移量: 1 (默认值) 表示为一周中的这一天创建快照、而不管该快照在哪个月份中) 	JSON 对象数组	无	是 (如果计划在周中的某些天生成快照)

返回值

此方法具有以下返回值：

名称	说明	类型
scheduleID	已创建的计划的 ID。	整型
schedule	包含新创建计划的相关信息的对象。	<i>schedule</i>

请求示例 1

以下示例计划具有下列参数：

- 由于未指定开始 hours 或 minutes，因此该计划在尽可能接近午夜 (00:00:00Z) 时启动。
- 此计划不重复运行（仅运行一次）。
- 此计划在 2015 年 6 月 1 日 UTC 19:17:15Z 之后的第一个星期日或星期三（以较早日期为准）仅运行一次。
- 此计划仅包括一个卷 (volumeID = 1)。

```
{
  "method": "CreateSchedule",
  "params": {
    "hours": 0,
    "minutes": 0,
    "paused": false,
    "recurring": false,
    "scheduleName": "MCAsnapshot1",
    "scheduleType": "snapshot",
    "attributes": {
      "frequency": "Days Of Week"
    },
    "scheduleInfo": {
      "volumeID": "1",
      "name": "MCA1"
    },
    "monthdays": [],
    "weekdays": [
      {
        "day": 0,
        "offset": 1
      },
      {
        "day": 3,
        "offset": 1
      }
    ],
    "startingDate": "2015-06-01T19:17:54Z"
  },
  "id": 1
}
```

响应示例 1

上述请求返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Days Of Week"
      },
      "hasError": false,
      "hours": 0,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": null,
      "minutes": 0,
      "monthdays": [],
      "paused": false,
      "recurring": false,
      "runNextInterval": false,
      "scheduleID": 4,
      "scheduleInfo": {
        "name": "MCA1",
        "volumeID": "1"
      },
      "scheduleName": "MCAsnapshot1",
      "scheduleType": "Snapshot",
      "startingDate": "2015-06-01T19:17:54Z",
      "toBeDeleted": false,
      "weekdays": [
        {
          "day": 0,
          "offset": 1
        },
        {
          "day": 3,
          "offset": 1
        }
      ]
    },
    "scheduleID": 4
  }
}
```

请求示例 2

以下示例计划具有下列参数：

- 此计划重复运行（将在月中每个计划间隔的指定时间运行）。
- 此计划在开始日期之后每个月的 1 日、10 日、15 日和 30 日运行。
- 此计划在每个计划日期的中午 12:15 运行。
- 此计划仅包括一个卷 (volumeID = 1)。

```
{
  "method": "CreateSchedule",
  "params": {
    "hours": 12,
    "minutes": 15,
    "paused": false,
    "recurring": true,
  }
}
```

```
    "scheduleName": "MCASnapshot1",
    "scheduleType": "snapshot",
    "attributes": {
      "frequency": "Days Of Month"
    },
    "scheduleInfo": {
      "volumeID": "1"
    },
    "weekdays": [
    ],
    "monthdays": [
      1,
      10,
      15,
      30
    ],
    "startingDate": "2015-04-02T18:03:15Z"
  },
  "id": 1
}
```

响应示例 2

上述请求返回类似于以下示例的响应:

```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Days Of Month"
      },
      "hasError": false,
      "hours": 12,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": null,
      "minutes": 15,
      "monthdays": [
        1,
        10,
        15,
        30
      ],
      "paused": false,
      "recurring": true,
      "runNextInterval": false,
      "scheduleID": 5,
      "scheduleInfo": {
        "volumeID": "1"
      },
      "scheduleName": "MCASnapshot1",
      "scheduleType": "Snapshot",
      "startingDate": "2015-04-02T18:03:15Z",
      "toBeDeleted": false,
      "weekdays": []
    },
    "scheduleID": 5
  }
}
```

请求示例 3

以下示例计划具有下列参数:

- 此计划在 2015 年 4 月 2 日的计划的间隔的 5 分钟内启动。
- 此计划重复运行 (将在月中每个计划间隔的指定时间运行)。
- 此计划在开始日期之后每个月的 2 日、3 日和 4 日运行。
- 此计划在每个计划日期的下午 14:45 运行。
- 此计划包括一组卷 (volumes = 1 和 2)。

```
{
  "method": "CreateSchedule",
  "params": {
    "hours": 14,
    "minutes": 45,
    "paused": false,
    "recurring": true,
    "scheduleName": "MCASnapUser1",
    "scheduleType": "snapshot",
    "attributes": {
      "frequency": "Days Of Month"
    },
    "scheduleInfo": {
      "volumes": [1, 2]
    },
    "weekdays": [],
    "monthdays": [2, 3, 4],
    "startingDate": "2015-04-02T20:38:23Z"
  },
  "id": 1
}
```

响应示例 3

上述请求返回类似于以下示例的响应:

```
{
  "id": 1,
  "result": {
    "schedule": {
      "attributes": {
        "frequency": "Days Of Month"
      },
      "hasError": false,
      "hours": 14,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": null,
      "minutes": 45,
      "monthdays": [
        2,
        3,
        4
      ],
      "paused": false,
      "recurring": true,
      "runNextInterval": false,
      "scheduleID": 6,
      "scheduleInfo": {
        "volumes": [
```

```

        1,
        2
    ]
  },
  "scheduleName": "MCASnapUser1",
  "scheduleType": "Snapshot",
  "startingDate": "2015-04-02T20:38:23Z",
  "toBeDeleted": false,
  "weekdays": []
},
"scheduleID": 6
}
}

```

发布以下版本后新增

9.6

CreateSnapshot

您可以使用 CreateSnapshot 创建卷的时间点副本。可以从任何卷或现有快照创建快照。

如果没有为此 API 方法提供快照 ID，则会从卷的活动分支创建快照。如果从其创建快照的卷将复制到远程集群，则该快照也可以复制到同一个目标。使用 enableRemoteReplication 参数可以启用快照复制。

注：当集群填充度处于第 1、2 或第 3 阶段时，您可以创建快照。当集群填充度达到第 4 或第 5 阶段时，您不能创建快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
enableRemoteReplication	指定快照是否会复制到远程存储。可能值： <ul style="list-style-type: none"> • true: 快照会复制到远程存储。 • false: 快照不会复制到远程存储。 	布尔型	false	否
name	快照的名称。如果未输入任何名称，则会使用生成快照时的日期和时间。允许的最大名称长度为 255 个字符。	字符串	无	否
retention	快照的保留时间。如果留空，则快照会永久保留。请以 HH:mm:ss 格式输入值。	字符串	无	否

名称	说明	类型	默认值	必需
snapMirrorLabel	SnapMirror 软件用于在 SnapMirror 端点上指定快照保留策略的标签。	字符串	无	否
snapshotID	从其生成新快照的快照的唯一 ID。传递的快照 ID 必须来自给定卷上的快照。	整型	无	否
volumeID	从中进行复制的卷映像的唯一 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
checksum	一个字符串，表示已存储的快照中的正确位数。此校验和以后可用于比较其他快照，以检测数据中存在的错误。	字符串
snapshotID	新快照的唯一 ID。	snapshot ID
snapshot	包含新创建快照的相关信息的对象。	snapshot

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateSnapshot",
  "params": {
    "volumeID": 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": {
    "checksum": "0x0",
    "snapshot": {
      "attributes": {},
      "checksum": "0x0",
      "createTime": "2016-04-04T17:14:03Z",
      "enableRemoteReplication": false,
      "expirationReason": "None",
      "expirationTime": null,
      "groupID": 0,
      "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",
      "name": "2016-04-04T17:14:03Z",
      "snapshotID": 3110,
    }
  }
}
```

```
        "snapshotUUID": "6f773939-c239-44ca-9415-1567eae79646",  
        "status": "done",  
        "totalSize": 5000658944,  
        "virtualVolumeID": null,  
        "volumeID": 1  
    },  
    "snapshotID": 3110  
}
```

发布以下版本后新增

9.6

DeleteGroupSnapshot

您可以使用 DeleteGroupSnapshot 删除组快照。

可以使用 saveMembers 参数保留为组中的卷生成的所有快照，但会删除组关联。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
groupSnapshotID	组快照的唯一 ID。	整型	无	是
saveMembers	指定删除组快照时要删除的内容。有效值： <ul style="list-style-type: none">• true: 将保留快照，但会删除组关联。• false: 将删除组和快照。	布尔型	false	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": "DeleteGroupSnapshot",  
  "params": {  
    "groupSnapshotID": 10,  
    "saveMembers": 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": {}  
}
```

发布以下版本后新增

9.6

DeleteSnapshot

您可以使用 DeleteSnapshot 方法删除快照。

无法删除当前处于活动状态的快照。您必须先回滚另一个快照并使其处于活动状态，然后才能删除当前快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
snapshotID	要删除的快照的 ID。	整型	无	是
overrideSnapMirrorHold	覆盖复制期间为快照设置的锁定。删除关联的 SnapMirror 关系后，可以使用此参数删除过时的 SnapMirror 快照。	布尔型	false	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": "DeleteSnapshot",
  "params": {
    "snapshotID": 8,
    "overrideSnapMirrorHold": 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": {}
}
```

发布以下版本后新增

9.6

相关参考

[RollbackToSnapshot](#) on page 466

您可以使用 `RollbackToSnapshot` 方法生成活动卷映像的现有快照。此方法将从现有快照创建新快照。

GetSchedule

您可以使用 `GetSchedule` 获取计划快照的相关信息。

如果系统中存在许多快照计划，您可以查看有关特定计划的信息。您还可以使用此方法通过在 `scheduleID` 参数中指定其他计划 ID 来检索有关多个计划的信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
scheduleID	要显示的一个或多个计划的唯一 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
schedule	计划属性数组。	schedule 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetSchedule",
  "params": {
    "scheduleID" : 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": {
    "schedule": {
      "attributes": {
        "frequency": "Time Interval"
      },
      "hasError": false,
      "hours": 0,
      "lastRunStatus": "Success",
      "lastRunTimeStarted": "2015-03-23T21:25:00Z",
      "minutes": 2,
      "monthdays": [],
      "paused": false,
      "recurring": true,
      "runNextInterval": false,
      "scheduleID": 2,
      "scheduleInfo": {
```

```

        "name": "MCA2",
        "volumeID": "3"
    },
    "scheduleName": "MCAsnapshot2",
    "scheduleType": "Snapshot",
    "startingDate": "2015-03-23T19:28:57Z",
    "toBeDeleted": false,
    "weekdays": []
}
}
}

```

发布以下版本后新增

9.6

ListGroupSnapshots

您可以使用 `ListGroupSnapshots` 方法返回已创建的所有组快照的相关信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
groupSnapshotID	检索单个组快照 ID 的信息。	整型	无	否
volumes	要查询的唯一卷 ID 数组。如果未指定此参数，将包括集群上的所有组快照。	volumeID 数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
groupSnapshots	一个对象列表，包含有关每个组快照的信息。	groupSnapshot 数组

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListGroupSnapshots",
  "params": {
    "volumes": [
      31,
      49
    ]
  },
  "id": 1
}

```

响应示例

此方法返回类似于以下示例的响应:

```
{
  "groupSnapshots": [
    {
      "status": "Done",
      "remoteStatuses": [
        {
          "volumePairUUID": "abcdef-1234-5678-90ab-cdef0123",
          "remoteStatus": "Present"
        }
      ],
      "attributes": {},
      "groupSnapshotID": 1,
      "createTime": "2014-06-17T17:35:05Z",
      "members": [
        {
          "snapshotUUID": "abcdef-1234-5678-90ab-cdef0123",
          "expirationReason": "None",
          "virtualVolumeID": "abcdef-1234-5678-90ab-cdef0123",
          "groupID": 1,
          "createTime": "2014-06-17T17:35:05Z",
          "totalSize": 1,
          "snapMirrorLabel": "test1",
          "volumeName": "test1",
          "instanceCreateTime": "2014-06-17T17:35:05Z",
          "volumeID": 1,
          "checksum": "0x0",
          "attributes": {},
          "instanceSnapshotUUID": "abcdef-1234-5678-90ab-cdef0123",
          "snapshotID": 1,
          "status": "Done",
          "groupSnapshotUUID": "abcdef-1234-5678-90ab-cdef0123",
          "expirationTime": "2014-06-17T17:35:05Z",
          "enableRemoteReplication": true,
          "name": "test1",
          "remoteStatuses": [
            {
              "volumePairUUID": "abcdef-1234-5678-90ab-cdef0123",
              "remoteStatus": "Present"
            }
          ]
        }
      ],
      "enableRemoteReplication": true,
      "name": "test1",
      "groupSnapshotUUID": "abcdef-1234-5678-90ab-cdef0123"
    }
  ]
}
```

发布以下版本后新增

9.6

ListSchedules

您可以使用 `ListSchedules` 获取已创建的所有计划快照的相关信息。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
<code>schedules</code>	集群中的当前计划列表。	schedule 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListSchedules",
  "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": {
    "schedules": [
      {
        "attributes": {
          "frequency": "Days Of Week"
        },
        "hasError": false,
        "hours": 0,
        "lastRunStatus": "Success",
        "lastRunTimeStarted": null,
        "minutes": 1,
        "monthdays": [],
        "paused": false,
        "recurring": false,
        "runNextInterval": false,
        "scheduleID": 3,
        "scheduleInfo": {
          "name": "Wednesday Schedule",
          "retention": "00:02:00",
          "volumeID": "2"
        },
        "scheduleName": "Vol2Schedule",
        "scheduleType": "Snapshot",
        "startingDate": "2015-03-23T20:08:33Z",
        "toBeDeleted": false,
        "weekdays": [
          {
            "day": 3,
            "offset": 1
          }
        ]
      }
    ]
  }
}
```

```

    ]
  },
  {
    "attributes": {
      "frequency": "Time Interval"
    },
    "hasError": false,
    "hours": 0,
    "lastRunStatus": "Success",
    "lastRunTimeStarted": "2015-03-23T21:40:00Z",
    "minutes": 2,
    "monthdays": [],
    "paused": false,
    "recurring": true,
    "runNextInterval": false,
    "scheduleID": 2,
    "scheduleInfo": {
      "name": "MCA2",
      "volumeID": "3"
    },
    "scheduleName": "MCAsnapshot2",
    "scheduleType": "Snapshot",
    "startingDate": "2015-03-23T19:28:57Z",
    "toBeDeleted": false,
    "weekdays": []
  }
]
}
}

```

发布以下版本后新增

9.6

ListSnapshots

您可以使用 `ListSnapshots` 返回在卷上生成的每个快照的属性。

从源集群调用此方法时，有关驻留在目标集群上的快照的信息将显示在源集群上。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	检索卷的快照。如果未提供卷 ID，将返回所有卷的所有快照。	整型	无	否
snapshotID	检索单个快照 ID 的信息。	整型	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
snapshots	有关每个卷的每个快照的信息。如果未提供卷 ID，将返回所有卷的所有快照。将为组中的快照返回一个组 ID。	<i>snapshot</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListSnapshots",
  "params": {
    "volumeID": "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": {
    "snapshots": [
      {
        "attributes": {},
        "checksum": "0x0",
        "createTime": "2015-05-08T13:15:00Z",
        "enableRemoteReplication": true,
        "expirationReason": "None",
        "expirationTime": "2015-05-08T21:15:00Z",
        "groupID": 0,
        "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",
        "name": "Hourly",
        "remoteStatuses": [
          {
            "remoteStatus": "Present",
            "volumePairUUID": "237e1cf9-fb4a-49de-a089-a6a9a1f0361e"
          }
        ],
        "snapshotID": 572,
        "snapshotUUID": "efa98e40-cb36-4c20-a090-a36c48296c14",
        "status": "done",
        "totalSize": 10000269312,
        "volumeID": 1
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ModifyGroupSnapshot

您可以使用 `ModifyGroupSnapshot` 更改快照组的属性。您还可以使用此方法启用读取/写入（源）卷上已创建的快照到目标存储系统的远程复制。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
enableRemoteReplication	用于启用已创建的快照到远程集群的复制。可能值： <ul style="list-style-type: none"> • true: 快照会复制到远程存储。 • false: 快照不会复制到远程存储。 	布尔型	false	否
expirationTime	用于设置应在何时删除快照。如果未输入时间，则会使用当前时间。如果设置为“null”，则快照永不过期。	ISO 8601 日期字符串	无	否
name	组快照的名称。如果未输入任何名称，则会使用生成组快照时的日期和时间。允许的最大名称长度为 255 个字符。	字符串	无	否
groupSnapshotID	快照组的 ID。	字符串	无	是
snapMirrorLabel	SnapMirror 软件用于在 SnapMirror 端点上指定快照保留策略的标签。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
groupSnapshot	包含新修改组快照的相关信息对象。	<i>groupSnapshot</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 695,
  "method": "ModifyGroupSnapshot",
  "params": {
    "groupSnapshotID": 3,
    "enableRemoteReplication": true,
    "expirationTime": "2016-04-08T22:46:25Z"
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 695,
  "result": {
    "groupSnapshot": {
```

```

    "attributes": {},
    "createTime": "2016-04-06T17:31:41Z",
    "groupSnapshotID": 3,
    "groupSnapshotUUID": "8b2e101d-c5ab-4a72-9671-6f239de49171",
    "members": [
      {
        "attributes": {},
        "checksum": "0x0",
        "createTime": "2016-04-06T17:31:41Z",
        "enableRemoteReplication": true,
        "expirationReason": "None",
        "expirationTime": "2016-04-08T22:46:25Z",
        "groupID": 3,
        "groupSnapshotUUID": "8b2e101d-c5ab-4a72-9671-6f239de49171",
        "name": "grpsnap1-2",
        "snapshotID": 2,
        "snapshotUUID": "719b162c-e170-4d80-b4c7-1282ed88f4e1",
        "status": "done",
        "totalSize": 1000341504,
        "virtualVolumeID": null,
        "volumeID": 2
      }
    ],
    "name": "grpsnap1",
    "status": "done"
  }
}
}

```

发布以下版本后新增

9.6

ModifySchedule

您可以使用 `ModifySchedule` 更改执行计划快照的间隔。您还可以使用此方法删除或暂停计划。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
attributes	用于更改快照的发生频率。可能值： <ul style="list-style-type: none"> Days of Week Days of Month Time Interval 	JSON 对象	无	否
hours	两次创建快照之间的小时数，或者在 Days of Week 或 Days of Month 模式下创建快照的小时值。有效值为 0 到 24。	字符串	无	否

名称	说明	类型	默认值	必需
name	快照的名称。如果未输入任何名称，则会使用生成组快照时的日期和时间。允许的最大名称长度为 244 个字符。	字符串	无	否
minutes	两次创建快照之间的分钟数，或者在 Days of Week 或 Days of Month 模式下创建快照的分钟值。有效值为 0 到 59。	整型	无	否
paused	指示是否应暂停该计划。有效值： <ul style="list-style-type: none"> • true • false 	布尔型	无	否
recurring	指示该计划是否重复运行。有效值为： <ul style="list-style-type: none"> • true • false 	布尔型	无	否
runNextInterval	用于选择是否在计划程序下次处于活动状态时运行快照。有效值： <ul style="list-style-type: none"> • true • false 如果设置为 true，计划快照将在计划程序下次处于活动状态时运行，然后重置为 false。	布尔型	false	否
scheduleID	计划的唯一 ID。	整型	无	是
scheduleName	计划的唯一名称。允许的最大计划名称长度为 244 个字符。	字符串	无	否
scheduleType	指示要创建的计划的类型。有效值为 snapshot。	字符串	无	是

名称	说明	类型	默认值	必需
scheduleInfo	为计划指定的唯一名称、已创建的快照的保留期限以及从其创建快照的卷的卷 ID。有效值： <ul style="list-style-type: none"> • volumeID: 要包括在快照中的卷的 ID。（整型） • volumes: 要包括在组快照中的卷的 ID 列表。（整型数组） • name: 要使用的快照名称。（字符串） • enableRemoteReplication: 指示是否应将快照包括在远程复制中。（布尔值） • retention: 快照的保留时间，格式为 HH:mm:ss。（字符串） 	JSON 对象	无	否
snapMirrorLabel	SnapMirror 软件用于在 SnapMirror 端点上指定快照保留策略的标签。	字符串	无	否
toBeDeleted	指示计划是否标记为待删除。有效值： <ul style="list-style-type: none"> • true • false 	布尔型	无	否
startingDate	指示首次开始该计划的日期。	ISO 8601 日期字符串	无	否
monthdays	将在月份中的哪些天生成快照。有效值为 1 到 31。	整型数组	无	是
weekdays	将在星期几创建快照。星期从星期日（其值为 0，偏移量为 1）开始。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
schedule	包含修改计划属性的对象。	schedule

请求示例

```
{
  "method": "ModifySchedule",
  "params": {
    "scheduleName" : "Chicago",
    "scheduleID" : 3
  },
}
```

```
"id": 1  
}
```

响应示例

```
{  
  "id": 1,  
  "result": {  
    "schedule": {  
      "attributes": {  
        "frequency": "Days Of Week"  
      },  
      "hasError": false,  
      "hours": 5,  
      "lastRunStatus": "Success",  
      "lastRunTimeStarted": null,  
      "minutes": 0,  
      "monthdays": [],  
      "paused": false,  
      "recurring": true,  
      "runNextInterval": false,  
      "scheduleID": 3,  
      "scheduleInfo": {  
        "volumeID": "2"  
      },  
      "scheduleName": "Chicago",  
      "scheduleType": "Snapshot",  
      "startingDate": null,  
      "toBeDeleted": false,  
      "weekdays": [  
        {  
          "day": 2,  
          "offset": 1  
        }  
      ]  
    }  
  }  
}
```

发布以下版本后新增

9.6

ModifySnapshot

您可以使用 `ModifySnapshot` 更改当前分配给快照的属性。此外，还可以使用此方法将读/写（源）卷上创建的快照远程复制到运行 **Element** 软件的目标存储集群。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
enableRemoteReplication	用于将已创建的快照复制到远程存储集群。可能值： <ul style="list-style-type: none"> • true: 快照会复制到远程存储。 • false: 快照不会复制到远程存储。 	布尔型	false	否
expirationTime	用于设置应在何时删除快照。如果设置为“null”，则快照永不过期。	ISO 8601 日期字符串	无	否
name	快照的名称。如果未输入任何名称，则会使用生成快照时的日期和时间。允许的最大名称长度为 255 个字符。	字符串	无	否
snapMirrorLabel	SnapMirror 软件用于在 SnapMirror 端点上指定快照保留策略的标签。	字符串	无	否
snapshotID	快照的标识符。	字符串	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
snapshot	包含新修改快照的相关信息的对象。	snapshot

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifySnapshot",
  "params": {
    "snapshotID": 3114,
    "enableRemoteReplication": "true",
    "name": "Chicago"
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "snapshot": {
      "attributes": {},
      "checksum": "0x0",

```

```

        "createTime": "2016-04-04T17:26:20Z",
        "enableRemoteReplication": true,
        "expirationReason": "None",
        "expirationTime": null,
        "groupID": 0,
        "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",
        "name": "test1",
        "snapshotID": 3114,
        "snapshotUUID": "5809a671-4ad0-4a76-9bf6-01cccf1e65eb",
        "status": "done",
        "totalSize": 5000658944,
        "virtualVolumeID": null,
        "volumeID": 1
    }
}
}

```

发布以下版本后新增

9.6

RollbackToGroupSnapshot

您可以使用 RollbackToGroupSnapshot 将快照组中的所有卷回滚到每个卷各自的快照。

回滚到组快照会创建组快照中每个卷的临时快照。

注: 当集群填充度处于第 1、2 或第 3 阶段时，允许创建快照。当集群填充度处于第 4 或第 5 阶段时，不会创建快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
groupSnapshotID	组快照的唯一 ID。	整型	无	是
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 对象	无	否
name	如果 saveCurrentState 设置为 true，则为用于创建组快照的卷当前状态的名称。如果未指定名称，则会将快照（组和各个卷）的名称设置为进行回滚时的时间戳。	字符串	无	否
saveCurrentState	指定是否保存上一个活动卷映像。有效值： <ul style="list-style-type: none"> • true: 保留上一个活动卷映像。 • false: 删除上一个活动卷映像。 	布尔型	false	否

返回值

此方法具有以下返回值：

名称	说明	类型
members	<p>一个数组，其中包含组快照成员的 volumeID 和 snapshotID。值：</p> <ul style="list-style-type: none"> checksum: 以短字符串表示已存储快照中的数据。此校验和以后可用于比较其他快照，以检测数据中存在的错误。（字符串） snapshotID: 从其生成新快照的快照的唯一 ID。快照 ID 必须来自给定卷上的快照。（整型） volumeID: 快照的源卷 ID。（整型） 	JSON 对象数组
groupSnapshotID	<p>如果 saveCurrentState 设置为 false，则此值为 null。</p> <p>如果 saveCurrentState 设置为 true，则为新建组快照的唯一 ID。</p>	整型
groupSnapshot	<p>如果 saveCurrentState 设置为 false，则此值为 null。</p> <p>如果 saveCurrentState 设置为 true，则为一个对象，其中包含有关 RollbackToGroupSnapshot 刚刚回滚到的组快照的信息。</p>	<i>groupSnapshot</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "id": 438,
  "method": "RollbackToGroupSnapshot",
  "params": {
    "groupSnapshotID": 1,
    "name": "grpsnap1",
    "saveCurrentState": true
  }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 438,
  "result": {
    "groupSnapshot": {
      "attributes": {},
      "createTime": "2016-04-06T17:27:17Z",
      "groupSnapshotID": 1,
      "groupSnapshotUUID": "468fe181-0002-4b1d-ae7f-8b2a5c171eee",
      "members": [
```

```

    {
      "attributes": {},
      "checksum": "0x0",
      "createTime": "2016-04-06T17:27:17Z",
      "enableRemoteReplication": false,
      "expirationReason": "None",
      "expirationTime": null,
      "groupID": 1,
      "groupSnapshotUUID": "468fe181-0002-4b1d-ae7f-8b2a5c171eee",
      "name": "2016-04-06T17:27:17Z",
      "snapshotID": 4,
      "snapshotUUID": "03563c5e-51c4-4e3b-a256-a4d0e6b7959d",
      "status": "done",
      "totalSize": 1000341504,
      "virtualVolumeID": null,
      "volumeID": 2
    }
  ],
  "name": "2016-04-06T17:27:17Z",
  "status": "done"
},
"groupSnapshotID": 3,
"members": [
  {
    "checksum": "0x0",
    "snapshotID": 2,
    "snapshotUUID": "719b162c-e170-4d80-b4c7-1282ed88f4e1",
    "volumeID": 2
  }
]
}
}
}

```

发布以下版本后新增

9.6

RollbackToSnapshot

您可以使用 `RollbackToSnapshot` 方法生成活动卷映像的现有快照。此方法将从现有快照创建新快照。

新快照将变为活动，现有快照将保留直至用户手动将其删除为止。系统会删除先前的活动快照，除非您将 `saveCurrentState` 参数设置为 `true`。

注：当集群填充度处于第 1、2 或第 3 阶段时，您可以创建快照。当集群填充度达到第 4 或第 5 阶段时，您不能创建快照。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
volumeID	卷的卷 ID。	整型	无	是
attributes	名称-值对的列表，采用 JSON 对象格式。	JSON 属性	无	否

名称	说明	类型	默认值	必需
name	快照的名称。如果未指定任何名称，则会使用将回滚到的快照的名称并在该名称的结尾附加“-copy”。	字符串	无	否
snapshotID	先前在给定卷上创建的快照的 ID。	整型	无	是
saveCurrentState	指定是否保存上一个活动卷映像。有效值： <ul style="list-style-type: none"> • true: 保留上一个活动卷映像。 • false: 删除上一个活动卷映像。 	布尔型	false	否

返回值

此方法具有以下返回值：

名称	说明	类型
checksum	以短字符串表示已存储快照中的数据。	字符串
snapshotID	如果 saveCurrentState 设置为 false，则此值为 null。 如果 saveCurrentState 设置为 true，则为新建快照的唯一 ID。	整型
snapshot	如果 saveCurrentState 设置为 false，则此值为 null。 如果 saveCurrentState 设置为 true，则为一个对象，其中包含有关新建快照的信息。	<i>snapshot</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "RollbackToSnapshot",
  "params": {
    "volumeID": 1,
    "snapshotID": 3114,
    "saveCurrentState": 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": {  
  "checksum": "0x0",  
  "snapshot": {  
    "attributes": {},  
    "checksum": "0x0",  
    "createTime": "2016-04-04T17:27:32Z",  
    "enableRemoteReplication": false,  
    "expirationReason": "None",  
    "expirationTime": null,  
    "groupID": 0,  
    "groupSnapshotUUID": "00000000-0000-0000-0000-000000000000",  
    "name": "test1-copy",  
    "snapshotID": 1,  
    "snapshotUUID": "30d7e3fe-0570-4d94-a8d5-3cc8097a6bfb",  
    "status": "done",  
    "totalSize": 5000658944,  
    "virtualVolumeID": null,  
    "volumeID": 1  
  },  
  "snapshotID": 1  
}
```

发布以下版本后新增

9.6

虚拟卷 API 方法

您可以使用 Element 软件虚拟卷 API 方法管理虚拟卷 (VVol)。您可使用这些 API 方法查看现有 VVol，以及创建、修改和删除虚拟卷存储容器。虽然您无法使用这些方法对普通卷进行操作，但可以使用普通卷 API 方法列出有关 VVol 的信息。

CreateStorageContainer

您可以使用 CreateStorageContainer 方法创建虚拟卷 (VVol) 存储容器。您可以使用存储容器进行报告和资源分配。您需要创建至少一个存储容器才能使用虚拟卷功能。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
name	存储容器的名称。请遵循 Element 软件帐户命名限制。	字符串	无	是
accountID	将成为存储容器的非存储容器帐户。	整型	无	否
initiatorSecret	启动程序的 CHAP 身份验证密钥。	字符串	无	否
targetSecret	目标的 CHAP 身份验证密钥。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
storageContainer	包含新创建存储容器的相关信息的对象。	<i>storageContainer</i>

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "CreateStorageContainer",
  "params": {
    "name" : "example"
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "storageContainer": {
      "accountID": 8,

```

```
    "initiatorSecret": "rVTOi25^H.d;cP}l",  
    "name": "example",  
    "protocolEndpointType": "SCSI",  
    "status": "active",  
    "storageContainerID": "a9ec1138-e386-4a44-90d7-b9acbbc05176",  
    "targetSecret": "6?AEIxWpvo6,!boM"  
  }  
}  
}
```

发布以下版本后新增

9.6

DeleteStorageContainers

您可以使用 DeleteStorageContainers 参数一次性从系统中删除最多 2000 个虚拟卷 (VVol) 存储容器。删除的存储容器不能包含任何 VVol。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
storageContainerIDs	要删除的存储容器的 ID 列表。最多可以在该列表中指定 2000 个 ID。	UUID 数组	无	是

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": "DeleteStorageContainers",  
  "params": {  
    "storageContainerIDs" : ["a9ec1138-e386-4a44-90d7-b9acbbc05176"]  
  },  
  "id": 1  
}
```

响应示例

此方法返回类似于以下示例的响应：

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

发布以下版本后新增

9.6

GetStorageContainerEfficiency

您可以使用 `GetStorageContainerEfficiency` 方法检索有关虚拟卷存储容器的效率信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
storageContainerID	要检索其效率信息的存储容器的 ID。	整型	无	是

返回值

此方法具有以下返回值：

名称	说明	类型
compression	通过压缩数据为存储容器中的所有虚拟卷节省的空间量。用一个比率表示，其中值 1 表示已在未压缩的情况下存储了数据。	浮点型
deduplication	通过执行重复数据删除为存储容器中的所有虚拟卷节省的空间量。用一个比率表示。	浮点型
missingVolumes	无法在其中查询效率数据的虚拟卷。造成卷缺失的原因可能包括垃圾收集 (GC) 周期不到一小时、网络连接暂时性丢失或服务在 GC 周期之后重新启动。	整型数组
thinProvisioning	已用空间与已分配用于存储数据的空间量之比。用一个比率表示。	浮点型
timestamp	GC 后最后一次收集效率数据的时间。	ISO 8601 数据字符串

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "GetStorageContainerEfficiency",
  "params": {
    "storageContainerID" : "6c95e24f-9f0b-4793-affb-5a4bc6c3d7e1"
  },
  "id" : 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "compression": 1,
    "deduplication": 1,
    "missingVolumes": [],
    "thinProvisioning": 1,
    "timestamp": "2016-04-12T15:39:49Z"
  }
}
```

```
}  
}
```

发布以下版本后新增

9.6

GetVirtualVolumeCount

您可以使用 `GetVirtualVolumeCount` 方法检索系统中当前所含的虚拟卷数量。

参数

此方法没有任何输入参数。

返回值

此方法具有以下返回值：

名称	说明	类型
count	系统中当前所含的虚拟卷数量。	整型

请求示例

此方法的请求类似于以下示例：

```
{  
  "method": "GetVirtualVolumeCount",  
  "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": {  
    "count": 5  
  }  
}
```

发布以下版本后新增

9.6

ListProtocolEndpoints

您可以使用 `ListProtocolEndpoints` 方法检索有关集群中所有协议端点的信息。协议端点控制对其关联虚拟卷存储容器的访问。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
protocolEndpointIDs	要检索其信息的协议端点 ID 列表。如果省略此参数，则该方法将返回所有协议端点的相关信息。	protocolEndpointID UUID 数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
protocolEndpoints	包含系统中所有协议端点的相关信息对象列表。	<i>protocolEndpoint</i> 数组

请求示例

此方法的请求类似于以下示例：

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

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "protocolEndpoints": [
      {
        "primaryProviderID": 1,
        "protocolEndpointID": "1387e257-d2e3-4446-be6d-39db71583e7b",
        "protocolEndpointState": "Active",
        "providerType": "Primary",
        "scsiNAADeviceID": "6f47acc2000000016970687200000000",
        "secondaryProviderID": 2
      },
      {
        "primaryProviderID": 2,
        "protocolEndpointID": "1f16ed86-3f31-4c76-b004-a1251187700b",
        "protocolEndpointState": "Active",
        "providerType": "Primary",
        "scsiNAADeviceID": "6f47acc2000000026970687200000000",
        "secondaryProviderID": 3
      },
      {
        "primaryProviderID": 4,
        "protocolEndpointID": "c6458dfe-9803-4350-bb4e-68a3feb7e830",
        "protocolEndpointState": "Active",
        "providerType": "Primary",
        "scsiNAADeviceID": "6f47acc2000000046970687200000000",
        "secondaryProviderID": 1
      }
    ]
  }
}
```

```

        },
        {
            "primaryProviderID": 3,
            "protocolEndpointID": "f3e7911d-0e86-4776-97db-7468c272213f",
            "protocolEndpointState": "Active",
            "providerType": "Primary",
            "scsiNAADeviceID": "6f47acc2000000036970687200000000",
            "secondaryProviderID": 4
        }
    ]
}
}
}

```

发布以下版本后新增

9.6

ListStorageContainers

您可以使用 `ListStorageContainers` 方法检索有关系统已知的所有虚拟卷存储容器的信息。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
storageContainerIDs	要检索信息的存储容器 ID 列表。如果省略此参数，则该方法将返回系统中所有存储容器的相关信息。	UUID 数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
storageContainers	包含系统中所有存储容器相关信息的对象列表。	<i>storageContainer</i> 数组

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListStorageContainers",
  "params": {
    "storageContainerIDs": ["efda8307-b916-4424-979e-658a3f16894d"]
  },
  "id" : 1
}

```

响应示例

此方法返回类似于以下示例的响应：

```

{
  "id": 6395,
  "result": {

```

```

    "storageContainers": [
      {
        "accountID": 64,
        "initiatorSecret": "EJ:08An1MyNQmL!7",
        "name": "VvolContainer",
        "protocolEndpointType": "SCSI",
        "status": "active",
        "storageContainerID": "efda8307-b916-4424-979e-658a3f16894d",
        "targetSecret": "g38}zWBK%206jQr~",
        "virtualVolumes": []
      }
    ]
  }
}

```

发布以下版本后新增

9.6

ListVirtualVolumeBindings

您可以使用 `ListVirtualVolumeBindings` 方法获取集群中绑定到协议端点的所有虚拟卷的列表。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
virtualVolumeBindingIDs	要检索其信息的虚拟卷绑定 ID 的列表。如果省略此参数，则该方法将返回所有虚拟卷绑定的相关信息。	整型数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
bindings	描述集群中绑定至协议端点的所有虚拟卷的对象列表。	<i>binding</i>

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListVirtualVolumeBindings",
  "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": {
  "bindings": [
    {
      "protocolEndpointID": "5dd53da0-b9b7-43f9-9b7e-b41c2558e92b",
      "protocolEndpointInBandID": "naa.6f47acc2000000016a67746700000000",
      "protocolEndpointType": "SCSI",
      "virtualVolumeBindingID": 177,
      "virtualVolumeHostID": "564de1a4-9a99-da0f-8b7c-3a41dfd64bf1",
      "virtualVolumeID": "269d3378-1ca6-4175-a18f-6d4839e5c746",
      "virtualVolumeSecondaryID": "0xe200000000a6"
    }
  ]
}

```

发布以下版本后新增

9.6

ListVirtualVolumeHosts

您可以使用 ListVirtualVolumeHosts 方法获取集群已知的所有虚拟卷主机的列表。虚拟卷主机是通过 VASA API 提供程序启动会话的 VMware ESX 主机。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
virtualVolumeHostIDs	要检索信息的虚拟卷主机 ID 列表。如果省略此参数，则该方法将返回所有虚拟卷主机的相关信息。	virtualVolumeHostID UUID 数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
hosts	描述集群中的虚拟卷主机的对象列表。	<i>host</i> 数组

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListVirtualVolumeHosts",
  "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": {
    "hosts": [
      {
        "bindings": [],
        "clusterID": "5ebdb4ad-9617-4647-adfd-c1013578483b",
        "hostAddress": "172.30.89.117",
        "initiatorNames": [
          "iqn.1998-01.com.vmware:zdc-dhcp-0-c-29-d6-4b-f1-1a0cd614",
          "iqn.1998-01.com.vmware:zdc-dhcp-0-c-29-d6-4b-f1-5bcf9254"
        ],
        "virtualVolumeHostID": "564de1a4-9a99-da0f-8b7c-3a41dfd64bf1",
        "visibleProtocolEndpointIDs": [
          "5dd53da0-b9b7-43f9-9b7e-b41c2558e92b"
        ]
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ListVirtualVolumes

您可以使用 `ListVirtualVolumes` 方法列出系统中当前所含的虚拟卷。您可以使用此方法列出所有虚拟卷或者仅列出一部分虚拟卷。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
details	响应中的详细信息级别。可能值： <ul style="list-style-type: none"><code>true</code>：将有关每个 VVol 的更多详细信息包括在响应中。<code>false</code>：将有关每个 VVol 的标准详细信息级别包括在响应中。	布尔型	False	否
limit	要列出的虚拟卷的最大数量。	整型	10000	否

名称	说明	类型	默认值	必需
recursive	指定是否将有关每个 VVol 的子卷信息包括在响应中。 可能值： <ul style="list-style-type: none"> • true: 将有关每个 VVol 的子卷信息包括在响应中。 • false: 不将有关每个 VVol 的子卷信息包括在响应中。 	布尔型	False	否
startVirtualVolumeID	响应列表开头处的虚拟卷的 ID。	UUID 类型	无	否
virtualVolumeIDs	要其检索信息的虚拟卷 ID 列表。如果省略此参数，此方法将仅返回这些虚拟卷的相关信息。	virtualVolume ID UUID 数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
nextVirtualVolumeID	列表中的下一个虚拟卷的 ID。	UUID
virtualVolumes	描述系统中当前所含虚拟卷的对象列表。	<i>virtualVolume</i> 数组

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ListVirtualVolumes",
  "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": {
    "nextVirtualVolumeID": "00000000-0000-0000-0000-000000000000",
    "virtualVolumes": [
      {
        "bindings": [
          177
        ],
        "children": [],
        "metadata": {
          "SFProfileId": "f4e5bade-15a2-4805-bf8e-52318c4ce443",
          "SFgenerationId": "0",
          "VMW_ContainerId": "abaab415-bedc-44cd-98b8-f37495884db0",

```

```

        "VMW_VVolName": "asdf",
        "VMW_VVolType": "Config",
        "VMW_VmID": "502e0676-e510-ccdd-394c-667f6867fcdf",
        "VMW_VVolProfile": "f4e5bade-15a2-4805-bf8e-52318c4ce443:0"
    },
    "parentVirtualVolumeID": "00000000-0000-0000-0000-000000000000",
    "snapshotID": 0,
    "snapshotInfo": null,
    "status": "done",
    "storageContainer": {
        "accountID": 1,
        "initiatorSecret": "B5)D1y10K)8IDN58",
        "name": "test",
        "protocolEndpointType": "SCSI",
        "status": "active",
        "storageContainerID": "abaab415-bedc-44cd-98b8-f37495884db0",
        "targetSecret": "qgae@{o{~8\"2U)U^"
    },
    "virtualVolumeID": "269d3378-1ca6-4175-a18f-6d4839e5c746",
    "virtualVolumeType": "config",
    "volumeID": 166,
    "volumeInfo": null
}
]
}
}

```

发布以下版本后新增

9.6

ListVirtualVolumeTasks

您可以使用 `ListVirtualVolumeTasks` 方法获取系统中的虚拟卷任务的列表。

参数

此方法具有以下输入参数：

名称	说明	类型	默认值	必需
<code>virtualVolumeTaskIDs</code>	要检索其信息的虚拟卷任务 ID 列表。如果省略此参数，则该方法将返回所有虚拟卷任务的相关信息。	UUID 数组	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
<code>tasks</code>	描述集群中的虚拟卷任务的对象列表。	task 数组

请求示例

此方法的请求类似于以下示例：

```

{
  "method": "ListVirtualVolumeTasks",
  "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": {
    "tasks": [
      {
        "cancelled": false,
        "cloneVirtualVolumeID": "fafeb3a0-7dd9-4c9f-8a07-80e0bbf6f4d0",
        "operation": "clone",
        "parentMetadata": {
          "SFProfileID": "f4e5bade-15a2-4805-bf8e-52318c4ce443",
          "SFgenerationID": "0",
          "VMW_ContainerId": "abaab415-bedc-44cd-98b8-f37495884db0",
          "VMW_GosType": "windows7Server64Guest",
          "VMW_VVolName": "asdf.vmdk",
          "VMW_VVolNamespace": "/vmfs/volumes/vvol:abaab415bedc44cd-98b8f37495884db0/rfc4122.269d3378-1ca6-4175-a18f-6d4839e5c746",
          "VMW_VVolType": "Data",
          "VMW_VmID": "502e0676-e510-ccdd-394c-667f6867fcd",
          "VMW_VvolAllocationType": "4",
          "VMW_VvolProfile": "f4e5bade-15a2-4805-bf8e-52318c4ce443:0"
        },
        "parentTotalSize": 42949672960,
        "parentUsedSize": 0,
        "status": "success",
        "virtualVolumeHostID": "564dela4-9a99-da0f-8b7c-3a41dfd64bf1",
        "virtualVolumeTaskID": "alb72df7-66a6-489a-86e4-538d0dbe05bf",
        "virtualVolumeID": "fafeb3a0-7dd9-4c9f-8a07-80e0bbf6f4d0"
      }
    ]
  }
}
```

发布以下版本后新增

9.6

ModifyStorageContainer

您可以使用 `ModifyStorageContainer` 方法更改现有虚拟卷存储容器。

参数

此方法具有以下输入参数:

名称	说明	类型	默认值	必需
storageContainerID	要修改的虚拟卷存储容器的唯一 ID。	UUID	无	是
initiatorSecret	启动程序的新 CHAP 身份验证密钥。	字符串	无	否
targetSecret	目标的新 CHAP 身份验证密钥。	字符串	无	否

返回值

此方法具有以下返回值：

名称	说明	类型
storageContainer	新创建的存储容器的相关信息。	storageContainer

请求示例

此方法的请求类似于以下示例：

```
{
  "method": "ModifyStorageContainer",
  "params": {
    "storageContainerID": "6c95e24f-9f0b-4793-affb-5a4bc6c3d7e1",
    "targetSecret": "O,IM;tOQdn9$JJ*8"
  },
  "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": 1,
  "result": {
    "storageContainer": {
      "accountID": 8,
      "initiatorSecret": "T$|5TO>2IY5sk4@k",
      "name": "doctest1",
      "protocolEndpointType": "SCSI",
      "status": "active",
      "storageContainerID": "6c95e24f-9f0b-4793-affb-5a4bc6c3d7e1",
      "targetSecret": "O,IM;tOQdn9$JJ*8"
    }
  }
}
```

发布以下版本后新增

9.6

集群故障代码

如果存储集群出现可能需要引起管理员注意的错误或状况，它会生成集群故障。您可以使用 `ListClusterFaults` 方法检索存储集群中当前已解决和未解决的故障列表。

以下列表提供了有关 NetApp Element 存储集群故障以及可能的解决方案的详细信息：

AuthenticationServiceDefault

一个或多个集群节点上的验证服务无法正常工作。

请联系 NetApp 支持部门获得帮助。

availableVirtualNetworkIPAddressesLow

IP 地址块中的虚拟网络地址数量较少。

要解决此故障，请向虚拟网络地址块添加更多 IP 地址。

blockClusterFull

没有足够的可用块存储空间可用于支持单节点丢失。有关 `GetClusterFullThreshold` 集群完全级别的详细信息，请参见 API 方法。此集群故障表示出现以下情况之一：

- 第 3 阶段低（警告）：用户定义的阈值已超过。调整“集群已满”设置，或者添加更多节点。
- 阶段 4 严重（错误）：没有足够的空间从 1 节点故障中恢复。不允许创建卷、快照和克隆。
- 第 5 阶段完全消耗（严重） 1：不允许写入或新的 iSCSI 连接。将保持当前的 iSCSI 连接。向集群添加更多容量之前，写入将失败。

要解决此故障，请清除或删除卷，或者向存储集群再添加一个存储节点。

块降级

由于故障，块数据不再完全复制。

严重性	说明
错误	只有一个完整块数据副本仍然可用。
严重	没有完整的块数据副本可用。

要解决此故障、请恢复任何脱机节点或块服务、或联系 NetApp 支持。

blockServiceTooFull

块服务当前所用空间过多。

要解决此故障，请清除或删除卷，或者向存储集群再添加一个存储节点。

时钟偏移超出故障阈值

集群主节点和呈现令牌的节点之间的时间偏差超出了建议的阈值。

存储集群无法自动纠正节点之间的时间偏差。要解决此故障，请使用您网络内部的 NTP 服务器，而不是默认安装的 NTP 服务器。如果您已经在使用内部 NTP 服务器、请联系 NetApp 支持以获得帮助。

clusterCannotSync

集群块数据处于降级状态、恢复完全块数据冗余的自动修复过程无法继续；太多节点或块服务脱机或集群块服务太满。

要解决此故障、请添加更多块容量或联系 NetApp 支持。

clusterFull

此存储集群没有更多可用存储空间。

要解决此故障，请添加更多存储。

clusterIOPSAreOverProvisioned

存储集群 IOPS 配置过度。所有最小 QoS IOPS 的总和超过集群的预期 IOPS。系统无法同时对所有卷保持最小 QoS。

要解决此故障，请减小卷的最低 QoS IOPS 设置。

disableDriveSecurityFailed

在空闲加密功能处于关闭的情况下，无法对驱动器禁用安全保护。此驱动器仍会启用驱动器安全保护。

故障详细信息会显示无法禁用驱动器安全保护的原因；您可能需要根据此原因调查问题所在。如果您需要恢复未成功禁用安全保护的磁盘，请执行以下步骤：

1. 将此驱动器移至“available”状态以逻辑删除此驱动器。
2. 对此驱动器执行安全擦除。
3. 将此驱动器移至“active”状态。

如果上述步骤仍无法解决此问题，请更换此驱动器。

disconnectedClusterPair

集群对已断开或配置不正确。

检查集群的网络连接。

disconnectedRemoteNode

远程节点已断开或配置不正确。检查节点之间的网络连通性。

disconnectedSnapMirrorEndpoint

远程 SnapMirror 端点已断开或配置不正确。检查集群和远程 SnapMirrorEndpoint 之间的网络连接。

driveAvailable

可将一个或多个驱动器添加存储集群中。一般来说，所有存储集群均应已添加所有驱动器，并且任何驱动器均不处于可用状态。如果意外出现此故障，请联系 NetApp 支持部门。

要解决此故障，请向存储集群添加任何可用驱动器。

driveFailed

当一个或多个驱动器出现故障时，群集将返回此故障、表明以下情况之一：

- 驱动器管理器无法访问驱动器。
- 片或块服务失败次数太多、可能是由于驱动器读取或写入故障、无法重新启动。
- 驱动器丢失。
- 无法访问节点的主服务（节点中的所有驱动器均被视为丢失 / 失败）。
- 驱动器已锁定、无法获取驱动器的身份验证密钥。
- 驱动器已锁定、解锁操作失败。

要解决此问题：

- 检查节点的网络连接性。
- 更换驱动器。
- 确保身份验证密钥可用。

driveWearFault

驱动器的剩余寿命已降到阈值以下、但仍在运行。此故障有两种可能的严重级别：严重和警告：

- 带有串行端口的驱动器： <节点插槽><驱动器插槽> 具有严重的磨损级别。
- 带有串行端口的驱动器： <节点插槽> 具有低磨损预留空间。

要解决此故障，请尽快更换驱动器。

duplicateClusterMasterCandidates

存储集群存在多个候选主节点。

请联系 NetApp 支持部门获得帮助。

enableDriveSecurityFailed

在空闲加密功能处于打开的情况下，无法对驱动器启用安全保护。

确保用于启用安全保护的密钥正确无误。如果您需要恢复无法成功启用安全保护的磁盘，请执行以下步骤：

1. 将此驱动器移至“available”状态以逻辑删除此驱动器。
2. 对此驱动器执行安全擦除。
3. 将此驱动器移至“active”状态。

如果上述步骤仍无法解决此问题，请更换此驱动器。

ensembleDegraded

某个集合节点已断开网络连接或电源。

要解决此故障，请为受影响的节点还原网络连接或电源。

exception

发生异常故障。这些故障不会自动从故障队列中清除。

请联系 NetApp 支持部门获得帮助。

failedSpaceTooFull

块服务未对数据写入请求做出响应。这会导致分区服务空间不足，从而无法存储失败的写入。

请联系 NetApp 支持部门获得帮助。

fanSensor

风扇传感器出现故障或丢失。

请联系 NetApp 支持部门获得帮助。

fibreChannelAccessDegraded

光纤通道节点无法通过其存储 IP 地址对存储集群中的其他节点做出响应。

检查集群的网络连接。

fibreChannelAccessUnavailable

所有光纤通道节点均无响应。此时将显示节点 ID。

检查集群的网络连接。

FibreChannelActiveXL

每个光纤通道节点的 IXL Nexus 计数接近支持的 8000 个活动会话数限制。

- 最佳实践限制为 5500 。
- 警告限制为 7500 。
- 最大限制（非强制）为 8192 。

要解决此故障、请将 IXL Nexus 计数降低到最佳实践限制 5500 以下。

fibreChannelConfig

此集群故障表示出现以下情况之一：

- 在 PCI 插槽上存在意外的光纤通道端口。
- 存在意外的光纤通道 HBA 型号。
- 光纤通道 HBA 固件出现问题。
- 某个光纤通道端口未联机。
- 配置光纤通道直通时出现持久性问题。

请联系 NetApp 支持部门获得帮助。

FibreChannelStaticXL

每个光纤通道节点的 IXL Nexus 计数接近支持的 16000 个静态会话数限制。

- 最佳实践限制为 11000 。
- 警告限制为 15000 。
- 最大限制（强制）为 16384 。

要解决此故障、请将 IXL Nexus 计数降低到最佳实践限制 11000 以下。

fileSystemCapacityLow

某个文件系统空间不足。

要解决此故障，请向此文件系统添加更多容量。

fipsDrivesMismatch

将非 FIPS 驱动器插入了 FIPS 存储节点中，或者将 FIPS 驱动器插入了非 FIPS 存储节点中。

请卸下或更换有问题的驱动器。

fipsDrivesOutOfCompliance

系统检测到已禁用空闲加密或存储集群中存在非 FIPS 硬件。

请启用空闲加密或从此存储集群中卸下非 FIPS 硬件。

fipsSelfTestFailure

系统在 FIPS 自检期间检测到故障。

请联系 NetApp 支持部门获得帮助。

hardwareConfigMismatch

此集群故障表示出现以下情况之一：

- 此配置与此节点定义不匹配。
- 此类节点的驱动器大小不正确。
- 节点正在使用不受支持的驱动器。
- 驱动器固件不匹配。
- 驱动器加密功能状态与其父节点不匹配。

请联系 NetApp 支持部门获得帮助。

IDPCertificateExpiration

与第三方身份认证提供者一起使用的集群的服务提供商 SSL 证书即将过期或已过期。此故障会根据紧急程度采用以下严重性：

严重性	说明
警告	证书将在 30 天内到期。

严重性	说明
错误	证书将在 7 天内到期。
严重	证书将在 3 天内过期或已过期。

要解决此问题、请在 SSL 证书过期之前更新该证书。将 `UpdateIdpConfiguration` 方法与 `RefreshCertificateExpirationtime=true` 一起使用以提供更新的 SSL 证书。

inconsistentBondModes

VLAN 设备上缺少绑定模式。此故障将显示预期的绑定模式和当前正在使用的绑定模式。

要解决此故障，请在每节点 Web UI 中修改绑定模式。

inconsistentInterfaceConfiguration

接口配置不一致。

要解决此故障，请确保存储集群中的节点接口配置一致。

inconsistentMtus

此集群故障表示出现以下情况之一：

- **Bond1G mismatch:** 在绑定 1G 接口上检测到 MTU 不一致。
- **Bond10G mismatch:** 在绑定 10G 接口上检测到 MTU 不一致。

此故障将显示存在问题的节点以及关联的 MTU 值。

要解决此故障，请在每节点 Web UI 中修改 MTU 设置。

inconsistentRoutingRules

此接口的路由规则不一致。

inconsistentSubnetMasks

VLAN 设备上的网络掩码与内部记录的 VLAN 网络掩码不匹配。此故障将显示预期网络掩码与当前使用的网络掩码。

要解决此故障，请在 Element（存储集群）Web UI 中修改子网掩码。

incorrectBondPortCount

绑定端口数不正确。

invalidConfiguredFibreChannelNodeCount

所需的两个光纤通道节点连接之一性能下降。如果仅连接了一个光纤通道节点，则会出现此故障。

要解决此故障，请检查集群网络连接和网络布线，并查看是否存在服务故障。如果网络或服务均无问题，请联系 NetApp 支持部门更换光纤通道节点。

irqBalanceFailed

尝试平衡中断时出现异常。

请联系 NetApp 支持部门获得帮助。

kmipCertificateFault（根证书颁发机构 [Certification Authority, CA] 证书即将到期）

根证书颁发机构 (Certification Authority, CA) 证书即将到期。此故障会根据紧急程度采用以下严重性：

严重性	说明
警告	证书将在 30 天内到期。

严重性	说明
错误	证书将在 7 天内到期。
严重	证书将在 3 天内到期。

要解决此故障，请在证书到期前更新此证书。从根 CA 获取一个新证书，此证书至少在未来 30 天后到期。使用 `ModifyKeyServerKmip` API 方法提供更新后的根 CA 证书。

kmipCertificateFault (客户端证书即将到期)

客户端证书即将到期。此故障会根据紧急程度采用以下严重性：

严重性	说明
警告	证书将在 30 天内到期。
错误	证书将在 7 天内到期。
严重	证书将在 3 天内到期。

要解决此故障，请使用 `GetClientCertificateSigningRequest` 方法创建一个新的 CSR。对此 CSR 进行签名并使其在 30 天后到期，然后使用 `ModifyKeyServerKmip` API 方法将即将到期的 KMIP 客户端证书替换为此新证书。

kmipCertificateFault (根证书颁发机构 [Certification Authority, CA] 证书已到期)

根 CA 证书已到期。

从根 CA 获取一个新证书，此证书至少在未来 30 天后到期。使用 `ModifyKeyServerKmip` API 方法提供更新后的根 CA 证书。

kmipCertificateFault (客户端证书已到期)

客户端证书已到期。

使用 `GetClientCertificateSigningRequest` API 方法创建一个新的 CSR 并对其进行签名，确保新的到期日期至少在未来 30 天后。使用 `ModifyKeyServerKmip` API 方法将已到期的客户端证书替换为此新证书。

kmipCertificateFault (根证书颁发机构 [Certification Authority, CA] 证书无效)

根 CA 证书无效。

确保提供的证书正确无误。如果需要，请从根 CA 重新获取此证书。使用 `ModifyKeyServerKmip` API 方法安装此正确的证书。

kmipCertificateFault (客户端证书无效)

客户端证书无效。

确保安装的 KMIP 客户端证书正确无误。应将此客户端证书的根 CA 安装在外部密钥管理服务器上。如果需要更新此客户端证书，请使用 `ModifyKeyServerKmip` API 方法。

kmipServerFault (连接失败)

一个或多个节点无法访问外部密钥管理服务器。

故障详细信息会提供此密钥服务器 ID。确保此服务器运行正常并可通过管理网络访问。如果只有部分节点无法访问此外部密钥管理服务器，则故障详细信息会列出这些无法访问此密钥服务器的节点。在网络或特定节点级别执行故障排除，以确定为什么只有部分节点可以访问此外部密钥管理服务器。

kmipServerFault (身份验证失败)

一个或多个节点无法向外部密钥管理服务器进行身份验证。

确保使用的根 CA 证书和 KMIP 客户端证书正确无误。如果需要更新任何证书，请使用 `ModifyKeyServerKmip` 方法安装正确的证书。

kmipServerFault (服务器错误)

外部密钥管理服务器出现错误。

故障详细信息会提供此错误的详细信息。您可能需要根据此错误对外部密钥管理服务器进行故障排除。

内存阈值

检测到大量可纠正或不可纠正的 ECC 错误。如果返回类型为 `Error` 的严重级别、则可能是由于 DIMM 故障造成的。

请联系 NetApp 支持部门获得帮助。

memoryUsageThreshold

内存利用率超过正常值。

请联系 NetApp 支持部门获得帮助。

metadataClusterFull

没有足够的可用元数据存储空间来支持单个节点丢失。有关 `GetClusterFullThreshold` 集群完全级别的详细信息，请参见 API 方法。此集群故障表示出现以下情况之一：

- 第 3 阶段低（警告）：用户定义的阈值已超过。调整“集群已满”设置，或者添加更多节点。
- 阶段 4 严重（错误）：没有足够的空间从 1 节点故障中恢复。不允许创建卷、快照和克隆。
- 第 5 阶段完全消耗（严重） 1：不允许写入或新的 iSCSI 连接。将保持当前的 iSCSI 连接。向集群添加更多容量之前，写入将失败。清除或删除数据或添加更多节点。

有关更多信息，请参见 [Understanding Cluster Fullness Levels](#)。

要解决此故障，请清除或删除卷，或者向存储集群再添加一个存储节点。

mtuCheckFailure

没有为网络设备设置正确的 MTU 大小。

要解决此故障，请确保已为所有网络接口和交换机端口配置巨型帧（MTU 大小高达 9000 字节）。

networkConfig

此集群故障表示出现以下情况之一：

- 不存在预期网络接口。
- 存在重复网络接口。
- 已配置网络接口，但它已关闭。
- 需要重新启动网络接口。

请联系 NetApp 支持部门获得帮助。

noAvailableVirtualNetworkIPAddresses

IP 地址块中没有可用的虚拟网络地址。无法向存储集群添加更多存储节点。

要解决此故障，请向虚拟网络地址块添加更多 IP 地址。

NodeHardwareFault（网络接口已关闭或电缆已断开）

网络接口已关闭或电缆已拔下。

要解决此故障、请检查节点或节点的网络连接性。

nodeHardwareFault（驱动器加密功能状态不匹配节点在插槽 < 节点插槽 > < 驱动器插槽 > 中的驱动器的加密功能状态）

驱动器与安装它的存储节点的加密功能不匹配。

节点硬件故障（对于此节点类型而言，插槽 < 节点插槽 > < 驱动器插槽 > 中的驱动器的驱动器大小 < actual size > 不正确 - 预期大小 >）

存储节点包含的驱动器大小不正确。

NodeHardwareFault（在插槽 < 节点插槽 > 中检测到不支持的驱动器；驱动器统计信息和运行状况信息将不可用）

存储节点包含不支持的驱动器。

nodeHardwareFault（插槽中的驱动器应使用固件版本 < 预期版本 >、但使用的版本不受支持）

存储节点包含运行不支持的固件版本的驱动器。

nodeOffline

Element 软件无法与指定节点进行通信。

要解决此故障，请检查集群网络连接和网络布线。如果网络没有问题，请联系 NetApp 支持部门更换节点。

notUsingLACPBondMode

未配置 LACP 绑定模式。

要解决此故障，请在部署存储节点时使用 LACP 绑定；如果未启用并正确配置 LACP，客户端可能会出现性能问题。

ntpServerUnreachable

存储集群无法与指定 NTP 服务器进行通信。

要解决此故障，请检查 NTP 服务器配置、网络和防火墙。

ntpTimeNotInSync

存储集群时间和指定 NTP 服务器时间之间的时间差太大。存储集群无法自动更正此时间差。

要解决此故障，请使用您网络内部的 NTP 服务器，而不是默认安装的 NTP 服务器。如果在使用内部 NTP 服务器的情况下仍存在此问题，请联系 NetApp 支持部门获得帮助。

nvrAmDeviceStatus

NVRAM 设备存在错误、将要出现故障或已出现故障。此故障具有以下严重性：

严重性	说明
警告	硬件检测到警告。这种情况可能是暂时的，例如温度警告。
错误	硬件检测到错误或严重状态。Cluster Master 尝试从操作中删除片驱动器（这会生成驱动器删除事件）。如果辅助片服务不可用，则不会删除该驱动器。

严重性	说明
严重	硬件检测到错误或严重状态。Cluster Master 尝试从操作中删除片驱动器（这会生成驱动器删除事件）。如果辅助片服务不可用，则不会删除该驱动器。

更换节点中所有出现故障的硬件。如果这不能解决问题、请联系 NetApp 支持以获得帮助。

powerSupplyError

此集群故障表示出现以下情况之一：

- 电源不存在。
- 电源出现故障。
- 缺少电源输入或电源输入超出范围。

要解决此故障，请确认为所有节点提供冗余电源。如果仍存在此问题，请联系 NetApp 支持部门。

provisionedSpaceTooFull

存储集群的已配置总容量过满。

要解决此故障，请添加更多已配置空间，或者删除并清除卷或快照。

remoteRepAsyncDelayExceeded

已超过为复制配置的异步延迟。

remoteRepClusterFull

卷已暂停远程复制，因为目标存储集群太满。

要解决此故障，请释放目标存储集群上的部分空间。

remoteRepSnapshotClusterFull

卷已暂停快照远程复制，因为目标存储集群太满。

要解决此故障，请释放目标存储集群上的部分空间。

remoteRepSnapshotsExceededLimit

卷已暂停快照远程复制，因为目标存储集群卷已超过其快照限制。

要解决此故障，请从远程集群上删除一些快照。

scheduleActionError

运行的一个或多个计划活动失败。

在以下情况下，此故障会清除：计划活动再次运行并且成功，删除了计划活动，先暂停再继续该活动。

sensorReadingFailed

基板管理控制器 (BMC) 自检失败或传感器无法与 BMC 通信。

请联系 NetApp 支持部门获得帮助。

serviceNotRunning

所需服务未运行。

请联系 NetApp 支持部门获得帮助。

sliceServiceTooFull

为分区服务分配的已配置容量太少。

要解决此故障，请添加更多存储节点或联系 NetApp 支持部门。

sliceServiceUnhealthy

系统已检测到分区服务运行状况不正常，并且正在自动停用该服务。

- 严重性 = 警告：未采取任何操作。此警告期限将在 6 分钟后过期。
- 严重性 = 错误：系统将自动停用数据并将其数据重新复制到其他正常驱动器。

检查网络连接问题和硬件错误。如果特定硬件组件出现故障，则会出现其它故障。当可访问分片服务或服务已停用时、该故障将会清除。

sshEnabled

已在存储集群中的一个或多个节点上启用 SSH 服务。

要解决此故障，请禁用节点上的 SSH 服务。

sslCertificateExpiration

与此节点关联的 SSL 证书即将过期或已过期。此故障会根据紧急程度采用以下严重性：

严重性	说明
警告	证书将在 30 天内到期。
错误	证书将在 7 天内到期。
严重	证书将在 3 天内过期或已过期。

要解决此故障，请续订 SSL 证书。如果需要，请联系 NetApp 支持部门以获得帮助。

传输容量

单个节点占存储集群容量的一半以上。

为了保持数据冗余性、系统会减少最大节点的容量、从而使其部分块容量处于闲置状态（未使用）。要解决此故障、请向现有存储节点添加更多驱动器或向集群添加存储节点。

tempSensor

温度传感器报告的温度高于正常值。此故障可能会与 powerSupplyError 或 fanSensor 故障同时触发。

要解决此故障，请检查存储集群附近是否存在影响气流的障碍物。如果需要，请联系 NetApp 支持部门以获得帮助。

upgrade

升级已进行了超过 24 小时。

要解决此故障，请继续此升级或联系 NetApp 支持部门以获得帮助。

unbalancedMixedNodes

单个节点占用的存储集群容量超过三分之一。

请联系 NetApp 支持部门获得帮助。

unresponsiveService

系统服务已停止响应。

请联系 NetApp 支持部门获得帮助。

virtualNetworkConfig

此集群故障表示出现以下情况之一：

- 某个接口不存在。
- 某个接口上的命名空间不正确。

- 网络掩码不正确。
- IP 地址不正确。
- 某个接口未启动并正常运行。
- 某个节点上存在多余接口。

请联系 NetApp 支持部门获得帮助。

volumesDegraded

二级卷未完全复制并同步。

同步完成后，此故障将会清除。

如果此故障仍然存在，请检查是否存在网络连接问题和硬件错误。

volumesOffline

此存储集群中的一个或多个卷已脱机。

请联系 NetApp 支持部门获得帮助。

相关概念

[联系 NetApp 支持部门](#) on page 563

如果您就 NetApp 产品需要帮助或有任何疑问或建议，请联系 NetApp 支持部门。

相关参考

[GetClusterFullThreshold](#) on page 115

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量指标。

[ListClusterFaults](#) on page 131

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。

访问控制

可用的元素 API 方法因您设置的访问类型而异。

accounts

以下方法可用于帐户访问类型：

AddAccount	GetAccountByID
ModifyAccount	GetAccountByName
ListAccounts	GetAccountEfficiency
RemoveAccount	

administrator

所有方法均可用于管理员访问类型。

clusterAdmin

以下方法可用于集群管理员访问类型：

AddClusterAdmin	ListBackupTargets
AddInitiatorsToVolumeAccessGroup	ListBulkVolumeJobs
AddLdapClusterAdmin	ListClusterAdmins
AddVirtualNetwork	ListClusterPairs
AddVirtualNetwork	ListNodeFibreChannelPortInfo
AddVolumetoVolumeAccessGroup	ListBackupTargets
CloneMultipleVolumes	ListDriveHardware
CompleteClusterPairing	ListFibreChannelSessions
CompleteVolumePairing	ListFibreChannelPortInfo
CreateBackupTarget	ListGroupSnapshots
CreateSchedule	ListActivePairedVolumes
CreateSnapshot	ModifyBackupTarget
CreateSupportBundle	ModifyClusterAdmin
CreateClusterSupportBundle	ModifyGroupSnapshot
CreateGroupSnapshot	ModifyClusterFullThreshold
CreateVolumeAccessGroup	ModifyVolumeAccessGroup
DeleteAllSupportBundles	ModifyVolumeAccessGroupLunAssignments
DeleteSnapshot	ModifyVolumePair
DeleteGroupSnapshot	ModifyVirtualNetwork
DeleteVolumeAccessGroup	RemoveClusterAdmin
DisableEncryptionAtRest	RemoveVolumePair

DisableLdapAuthentication	RemoveVirtualNetwork
DisableSnmpp	RemoveVolumesFromVolumeAccessGroup
EnableEncryptionAtRest	RemoveInitiatorsFromVolumeAccessGroup
EnableLdapAuthentication	RollbackToSnapshot
EnableSnmpp	RollbackToGroupSnapshot
GetBackupTarget	SetLoginSessionInfo
GetClusterFullThreshold	SetNtpInfo
GetClusterMasterNodeID	SetSnmppACL
GetHardwareConfig	SetSnmppInfo
GetLdapConfiguration	SetSnmppTrapInfo
GetLoginSessionInfo	SetRemoteLoggingHosts
GetNtpInfo	Shutdown
GetNvramInfo	StartBulkVolumeRead
GetRawStats	StartBulkVolumeWrite
GetSnmppACL	StartClusterPairing
GetVolumeAccessGroupEfficiency	StartVolumePairing
GetVolumeAccessLunAssignments	TestLdapAuthentication
GetVirtualNetwork	

drives

以下方法可用于驱动器访问类型:

ListDrives	RemoveDrives
AddDrives	SecureEraseDrives

nodes

以下方法可用于节点访问类型:

AddNodes	ListPendingNodes
ListActiveNodes	RemoveNodes

read

以下方法可用于读取访问类型:

GetAccountByID	ListCloneJobs
GetAccountByName	ListDeletedVolumes
GetAsyncResult	ListDriveHardware
GetClusterCapacity	ListDrives
GetDefaultQoS	ListEvents

GetDriveStats	ListISCSISessions
GetSoftwareUpgrade	ListPendingNodes
GetVolumeStats	ListSyncJobs
ListAccounts	ListVolumeAccessGroups
ListActiveNodes	ListVolumeStatsByAccount
ListActiveNodes	ListVolumeStatsByVolume
ListActiveVolumes	ListVolumeStatsByVolumeAccessGroup
ListAllNodes	ListVolumesForAccount
ListBackupTargets	

reporting

以下方法可用于报告访问类型:

ClearClusterFaults	GetVolumeEfficiency
GetAccountEfficiency	GetVolumeStats
GetClusterCapacity	ListCloneJobs
GetClusterHardwareInfo	ListClusterFaults
GetClusterInfo	ListClusterPairs
GetClusterMasterNodeID	ListDriveHardware
GetClusterStats	ListEvents
GetDriveHardwareInfo	ListISCSISessions
GetDriveStats	ListSchedules
GetNetworkConfig	ListServices
GetNodeHardwareInfo	ListSyncJobs
GetNodeStats	ListVirtualNetworks
GetSnmpInfo	ListVolumeStatsByAccount
GetSnmpTrapInfo	ListVolumeStatsByVolume
GetVolumeAccessGroupEfficiency	ListVolumeStatsByVolumeAccessGroup

repositories

ListAllNodes 方法可用于存储库访问类型。

volumes

以下方法可用于卷访问类型:

CreateVolume	DeleteVolume	ModifyBackupTarget
CloneVolume	DeleteVolumePairing	ModifyVolumes
CloneMultipleVolumes	GetBackupTarget	ModifyVolumePair
CreateBackupTarget	GetDefaultQoS	PurgeDeletedVolume

CreateSnapshot	ListActiveVolumes	RemoveBackupTarget
CreateGroupSnapshot	ListBackupTarget	RemoveVolumePair
CompleteVolumePairing	ListGroupSnapshots	RestoreDeletedVolume
CloneMultipleVolumes	ListVolumesForAccount	RollbackToGroupSnapshot
DeleteGroupSnapshot	ListDeletedVolumes	RollbackToSnapshot
DeleteSnapshot	ListGroupSnapshots	StartBulkVolumeRead
StartBulkVolumeWrite	StartVolumePairing	UpdateBulkVolumeStatus

write

以下方法可用于写入访问类型：

AddDrives	RemoveNodes
AddNodes	RemoveAccount
AddAccount	RemoveVolumesFromVolumeAccessGroup
AddVolumeToVolumeAccessGroup	RemoveInitiatorsFromVolumeAccessGroup
AddInitiatorsToVolumeAccessGroup	DeleteVolumeAccessGroup
CreateVolumeAccessGroup	DeleteVolume
ModifyVolumeAccessGroup	RestoreDeletedVolume
ModifyAccount	PurgeDeletedVolume
CreateVolume	ModifyVolume
CloneVolume	GetAsyncResult
RemoveDrives	

响应示例

此处提供了完整的响应示例。

GetConfig

GetConfig 方法返回类似于以下示例的响应。鉴于篇幅限制，该响应仅包含集群中一个节点的信息。

```
{
  "id": 1,
  "result": {
    "config": {
      "cluster": {
        "cipi": "Bond10G",
        "cluster": "AutoTest2-Fjqt",
        "encryptionCapable": true,
        "ensemble": [
          "1:10.1.1.0",
          "3:10.1.1.0",
          "4:10.1.1.0"
        ],
        "mipi": "Bond1G",
        "name": "NLABP2605",
        "nodeID": 1,
        "pendingNodeID": 0,
        "role": "Storage",
        "sipi": "Bond10G",
        "state": "Active",
        "version": "11.0"
      },
      "network": {
        "Bond10G": {
          "#default": false,
          "address": "10.1.1.0",
          "auto": true,
          "bond-downdelay": "0",
          "bond-fail_over_mac": "None",
          "bond-miimon": "100",
          "bond-mode": "ActivePassive",
          "bond-primary_reselect": "Failure",
          "bond-slaves": "eth0 eth1",
          "bond-updelay": "200",
          "dns-nameservers": "10.1.1.0, 10.1.1.0",
          "dns-search": "ten.test.company.net., company.net.",
          "family": "inet",
          "gateway": "10.1.1.0",
          "linkSpeed": 10000,
          "macAddress": "c8:1f:66:ee:59:b9",
          "macAddressPermanent": "00:00:00:00:00:00",
          "method": "static",
          "mtu": "9000",
          "netmask": "255.255.240.0",
          "network": "10.1.1.0",
          "physical": {
            "address": "10.1.1.0",
            "macAddress": "c8:1f:66:ee:59:b9",
            "macAddressPermanent": "00:00:00:00:00:00",

```

```
        "mtu": "9000",
        "netmask": "255.255.240.0",
        "network": "10.1.1.0",
        "upAndRunning": true
    },
    "routes": [],
    "status": "UpAndRunning",
    "symmetricRouteRules": [
        "ip route add 10.1.1.1/20 dev Bond1G src 10.1.2.2 table Bond1G",
        "ip rule add from 10.1.1.1 table Bond1G",
        "ip route add default via 10.1.1.254"
    ],
    "upAndRunning": true,
    "virtualNetworkTag": "0"
},
"eth0": {
    "auto": true,
    "bond-master": "Bond10G",
    "family": "inet",
    "linkSpeed": 10000,
    "macAddress": "c8:1f:66:ee:59:b9",
    "macAddressPermanent": "c8:1f:66:ee:59:b9",
    "method": "bond",
    "physical": {
        "address": "0.0.0.0",
        "macAddress": "c8:1f:66:ee:59:b9",
        "macAddressPermanent": "c8:1f:66:ee:59:b9",
        "netmask": "N/A",
        "network": "N/A",
        "upAndRunning": true
    },
    "status": "UpAndRunning",
    "upAndRunning": true
},
"lo": {
    "auto": true,
    "family": "inet",
    "linkSpeed": 0,
    "macAddress": "00:00:00:00:00:00",
    "macAddressPermanent": "00:00:00:00:00:00",
    "method": "loopback",
    "physical": {
        "address": "0.0.0.0",
        "macAddress": "00:00:00:00:00:00",
        "macAddressPermanent": "00:00:00:00:00:00",
        "netmask": "N/A",
        "network": "N/A",
        "upAndRunning": true
    },
    "status": "UpAndRunning",
    "upAndRunning": true
}
}
}
}
```

GetClusterHardwareInfo

GetClusterHardwareInfo 方法返回类似于以下示例的响应。

```
{
  "id": null,
  "result": {
    "clusterHardwareInfo": {
      "drives": {
        "1": {
          "description": "ATA      Drive",
          "dev": "8:0",
          "devpath": "/dev/disk/by-id/scsi-SATA_VRFSD3400GNCVMT205121562-part4",
          "driveSecurityAtMaximum": false,
          "driveSecurityFrozen": true,
          "driveSecurityLocked": false,
          "logicalname": "/dev/sda",
          "product": "VRFSD3400GNCVMTJS1",
          "securityFeatureEnabled": false,
          "securityFeatureSupported": true,
          "serial": "205121562",
          "size": 299988156416,
          "uuid": "febe39ae-4984-edc0-e3a7-3c47608cface",
          "version": "515ABBF0"
        },
        "2": {...
      },
        "3": {...
      },
        "4": {...
      },
        "5": {...
      },
        "6": {...
      },
        .
        .
        .
        "44": {...
      },
    },
  },
  "nodes": {
    "1": {
      Storage Node
      "core_DMI:0200": {
        "description": "Motherboard",
        "physid": "0",
        "vendor": "SolidFire"
      },
      "fiber:0_PCI:0000:04:00.0": {
        "businfo": "pci@0000:04:00.0",
        "clock": "33000000",
        "description": "Fibre Channel",
        "physid": "0",
        "product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
        "vendor": "QLogic Corp.",
        "version": "02",
        "width": "64"
      },
    },
    "Repeat fiber information": {...}
    "Repeat fiber": {...},
  },
}
```

```
    "Repeat fiber": {...},
  },
  "fans": {
    "Fan1A RPM": {
      "baseUnit": "RPM",
      "threshold": 840,
      "value": 4800
    },
    "Fan1B RPM": {...},
    .
    .
    "Fan7B RPM": {...}
  },
  "fibreChannelPorts": [
    {
      "firmware": "7.04.00 (d0d5)",
      "hbaPort": 1,
      "model": "QLE2672",
      "nPortID": "0x110c36",
      "pciSlot": 3,
      "serial": "BFE1341E09329",
      "speed": "8 Gbit",
      "state": "Online",
      "switchWwn": "20:01:00:2a:6a:a0:25:01",
      "wwnn": "5f:47:ac:c8:82:23:e0:00",
      "wwpn": "5f:47:ac:c0:82:23:e0:02"
    },
    {
      "firmware": "7.04.00 (d0d5)", {...}
      "firmware": "7.04.00 (d0d5)", {...}
      "firmware": "7.04.00 (d0d5)", {...}
    }
  ],
  "hardwareConfig": {
    "BIOS_REVISION": {
      "Passed": true,
      "actual": "1.1",
      "comparator": ">=",
      "expected": "1.0"
    },
    "BIOS_VENDOR": {
      "Passed": true,
      "actual": "SolidFire",
      "comparator": "==",
      "expected": "SolidFire"
    },
    "BIOS_VERSION": {
      "Passed": true,
      "actual": "1.1.2",
      "comparator": ">=",
      "expected": "1.1.2"
    },
    "BMC_FIRMWARE_REVISION": {
      "Passed": true,
      "actual": "1.6",
      "comparator": ">=",
      "expected": "1.6"
    }
  },
}
```

```
"BMC_IPMI_VERSION": {
  "Passed": true,
  "actual": "2.0",
  "comparator": ">=",
  "expected": "2.0"
},
"CHASSIS_TYPE": {
  "Passed": true,
  "actual": "R620",
  "comparator": "==",
  "expected": "R620"
},
"CPU_CORES_00": {
  "Passed": true,
  "actual": "6",
  "comparator": "==",
  "expected": "6"
},
"CPU_CORES_01": {
  "Passed": true,
  "actual": "6",
  "comparator": "==",
  "expected": "6"
},
"CPU_CORES_ENABLED_00": {
  "Passed": true,
  "actual": "6",
  "comparator": "==",
  "expected": "6"
},
"CPU_CORES_ENABLED_01": {
  "Passed": true,
  "actual": "6",
  "comparator": "==",
  "expected": "6"
},
"CPU_MODEL_00": {
  "Passed": true,
  "actual": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
  "comparator": "==",
  "expected": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz"
},
"CPU_MODEL_01": {
  "Passed": true,
  "actual": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
  "comparator": "==",
  "expected": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz"
},
"CPU_THREADS_00": {
  "Passed": true,
  "actual": "12",
  "comparator": "==",
  "expected": "12"
},
"CPU_THREADS_01": {
  "Passed": true,
  "actual": "12",
  "comparator": "==",
  "expected": "12"
},
```

```
"DRIVE_SIZE_BYTES_SDIMM0": {
  "Passed": true,
  "actual": "100030242816",
  "comparator": ">=",
  "expected": "100030242816"
},
"FIBRE_CHANNEL_FIRMWARE_REVISION": {
  "Passed": true,
  "actual": "FW:v7.04.00",
  "comparator": "==",
  "expected": "FW:v7.04.00"
},
"FIBRE_CHANNEL_MODEL": {
  "Passed": true,
  "actual": "QLE2672",
  "comparator": "==",
  "expected": "QLE2672"
},
"IDRAC_VERSION": {
  "Passed": true,
  "actual": "1.06.06",
  "comparator": ">=",
  "expected": "1.06.06"
},
"LIFECYCLE_VERSION": {
  "Passed": true,
  "actual": "1.0.0.5747",
  "comparator": ">=",
  "expected": "1.0.0.5747"
},
"MEMORY_GB": {
  "Passed": true,
  "actual": "32",
  "comparator": ">=",
  "expected": "32"
},
"MEMORY_MHZ_00": {
  "Passed": true,
  "actual": "1333",
  "comparator": ">=",
  "expected": "1333"
},
"MEMORY_MHZ_01": {
  "Passed": true,
  "actual": "1333",
  "comparator": ">=",
  "expected": "1333"
},
"MEMORY_MHZ_02": {
  "Passed": true,
  "actual": "1333",
  "comparator": ">=",
  "expected": "1333"
},
"MEMORY_MHZ_03": {
  "Passed": true,
  "actual": "1333",
  "comparator": ">=",
  "expected": "1333"
},
}
```

```
"NETWORK_DRIVER_ETH0": {
  "Passed": true,
  "actual": "bnx2x",
  "comparator": "=~",
  "expected": "^bnx2x$"
},
{
  "NETWORK_DRIVER_ETH1":, {...
},
  "NETWORK_DRIVER_ETH2":, {...
},
  "NETWORK_DRIVER_ETH3":, {...
},
  "NETWORK_DRIVER_ETH4":, {...
},
  "NETWORK_DRIVER_ETH5":, {...
},
  "NODE_TYPE": {
    "Passed": true,
    "actual": "FC0025",
    "comparator": "==",
    "expected": "FC0025"
  },
  "NUM_CPU": {
    "Passed": true,
    "actual": "2",
    "comparator": "==",
    "expected": "2"
  },
  "NUM_DRIVES": {
    "Passed": true,
    "actual": "0",
    "comparator": "==",
    "expected": "0"
  },
  "NUM_DRIVES_INTERNAL": {
    "Passed": true,
    "actual": "1",
    "comparator": "==",
    "expected": "1"
  },
  "NUM_FIBRE_CHANNEL_PORTS": {
    "Passed": true,
    "actual": "4",
    "comparator": "==",
    "expected": "4"
  },
  "NVRAM_VENDOR": {
    "Passed": true,
    "actual": "",
    "comparator": "==",
    "expected": ""
  },
  "ROOT_DRIVE_REMOVABLE": {
    "Passed": true,
    "actual": "false",
    "comparator": "==",
    "expected": "false"
  }
},
```

```
"memory": {
  "firmware_": {
    "capacity": "8323072",
    "date": "03/08/2012",
    "description": "BIOS",
    "physid": "0",
    "size": "65536",
    "vendor": "SolidFire",
    "version": "1.1.2"
  },
  "memory_DMI:1000": {
    "description": "System Memory",
    "physid": "1000",
    "size": "34359738368",
    "slot": "System board or motherboard"
  }
},
"network": {
  "network:0_PCI:0000:01:00.0": {
    "businfo": "pci@0000:01:00.0",
    "capacity": "1000000000",
    "clock": "33000000",
    "description": "Ethernet interface",
    "logicalname": "eth0",
    "physid": "0",
    "product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
    "serial": "c8:1f:66:e0:97:2a",
    "vendor": "Broadcom Corporation",
    "version": "10",
    "width": "64"
  },
  "network:0_PCI:0000:41:00.0": {...
},
  "network:1_PCI:0000:01:00.1": {...
},
  "network:1_PCI:0000:41:00.1": {...
},
  "network:2_PCI:0000:01:00.2": {...
},
  "network:3_PCI:0000:01:00.3": {...
}
},
"networkInterfaces": {
  "Bond10G": {
    "isConfigured": true,
    "isUp": true
  },
  "Bond1G": {
    "isConfigured": true,
    "isUp": true
  },
  "eth0": {
    "isConfigured": true,
    "isUp": true
  },
  "eth1": {...
},
  "eth2": {...
},
  "eth3": {...
}
```



```
    "level_8": " 0",
    "level_9": " 0",
    "name": "enterpriseFlashControllerTemperature",
    "recent": "66 C"
  },
  {
    "level_0": " 0",
    "level_1": " 58",
    "level_2": " 1479058",
    "level_3": " 12885356",
    "level_4": " 308293",
    "level_5": " 851",
    "level_6": " 29",
    "level_7": " 0",
    "level_8": " 0",
    "level_9": " 0",
    "name": "capacitor1And2Temperature",
    "recent": "30.69 C"
  },
  {
    ...next temp measurement
  },
  {
    ...next temp measurement
  },
  {
    ...next temp measurement
  },
  {
    "name": "voltageOfCapacitor1",
    "recent": "2.198 V"
  },
  {
    "name": "voltageOfCapacitor2",
    "recent": "2.181 V"
  },
  {
    "name": "voltageOfCapacitor3",
    "recent": "2.189 V"
  },
  {
    "name": "voltageOfCapacitor4",
    "recent": "2.195 V"
  },
  {
    "level_0": " 4442034",
    "level_1": " 6800018",
    "level_2": " 2846869",
    "level_3": " 119140",
    "level_4": " 29506",
    "level_5": " 428935",
    "level_6": " 7143",
    "level_7": " 0",
    "level_8": " 0",
    "level_9": " 0",
    "name": "capacitorPackVoltage",
    "recent": "8.763 V"
  },
  {
    "level_0": " 0",
    "level_1": " 0",
    "level_2": " 0",
    "level_3": " 0",
```

```
    "level_4": " 189",
    "level_5": " 17",
    "level_6": " 36",
    "level_7": " 0",
    "level_8": " 2",
    "level_9": " 490",
    "name": "capacitorPackVoltageAtEndOfFlushToFlash",
    "recent": "4.636 V"
  },
  {
    "name": "currentDerivedFromV3V4",
    "recent": "-0.004 A"
  },
  {
    "level_0": " 230",
    "level_1": " 482",
    "level_2": " 22",
    "level_3": " 0",
    "level_4": " 0",
    "level_5": " 0",
    "level_6": " 0",
    "level_7": " 0",
    "level_8": " 0",
    "level_9": " 0",
    "name": "derivedEnergy",
    "recent": "172 Joules"
  },
  {...next voltage measurement
  },
  {...next voltage measurement
  },
  {...next voltage measurement
  },
],
"smartCounters": [
  {
    "name": "numberOf512ByteBlocksReadFromDdr",
    "value": "10530088847"
  },
  {
    "name": "numberOf512ByteBlocksWrittenToDdr",
    "value": "1752499453837"
  },
  {
    "name": "numberOfHostReadCommands",
    "value": "235317769"
  },
  {...next smartCounters measurement
  },
  {...next smartCounters measurement
  },
  {...next smartCounters measurement
  },
],
"snapshotTime": "2015-08-20 16:30:01"
},
"firmware": {
  "activeSlotNumber": "2",
  "slot1Version": "1e5817bc",
  "slot2Version": "5fb7565c",
```

```
    "slot3Version": "1e5817bc",
    "slot4Version": "1e5817bc"
  },
  "identify": {
    "firmwareVersion": "5fb7565c on slot 2",
    "hardwareRevision": "B04",
    "modelName": "RMS-200",
    "serialNumber": "0000862"
  },
  "smart": {
    "availableSpace": "0%",
    "availableSpaceThreshold": "0%",
    "controllerBusyTimeMinutes": "6793",
    "criticalErrorVector": "0x0",
    "mediaErrors": "0",
    "numberOf512ByteBlocksRead": "10530088847",
    "numberOf512ByteBlocksWritten": "1752499439063",
    "numberOfErrorInfoLogs": "1",
    "numberOfHostReadCommands": "235317769",
    "numberOfHostWriteCommands": "126030374065",
    "numberOfPowerCycles": "709",
    "powerOnHours": "11223",
    "temperature": "324 Kelvin",
    "unsafeShutdowns": "357"
  },
  "origin": null,
  "platform": {
    "chassisType": "R620",
    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
    "nodeMemoryGB": 32,
    "nodeType": "FC0025"
  },
  "powerSupplies": {
    "PS1 status": {
      "powerSupplyFailureDetected": false,
      "powerSupplyHasAC": true,
      "powerSupplyPredictiveFailureDetected": false,
      "powerSupplyPresent": true,
      "powerSupplyPresentLastCheck": true
    },
    "PS2 status": {
      "powerSupplyFailureDetected": false,
      "powerSupplyHasAC": true,
      "powerSupplyPredictiveFailureDetected": false,
      "powerSupplyPresent": true,
      "powerSupplyPresentLastCheck": true
    }
  },
  "storage": {
    "storage_PCI:0000:00:1f.2": {
      "businfo": "pci@0000:00:1f.2",
      "clock": "66000000",
      "description": "SATA controller",
      "physid": "1f.2",
      "product": "C600/X79 series chipset 6-Port SATA AHCI Controller",
      "vendor": "Intel Corporation",
      "version": "05",
      "width": "32"
    }
  }
}
```

```
    },
    "system": {
      "ubuntu_DMI:0100": {
        "description": "Rack Mount Chassis",
        "product": "SFx010 ()",
        "serial": "HTW1DZ1",
        "vendor": "SolidFire",
        "width": "64"
      }
    },
    "temperatures": {
      "Exhaust Temp": {
        "baseUnit": "C",
        "threshold": 70,
        "value": 41
      },
      "Inlet Temp": {
        "baseUnit": "C",
        "threshold": 42,
        "value": 18
      }
    },
    "uuid": "4C4C4544-0054-5710-8031-C8C04F445A31"
  },
  "2": { ... },           Storage Node "2"
  "3": { ... },           Storage Node "3"
  "4": { ... },           Storage Node "4"
  "5": {                   Fibre Channel Node
  }
}
}
```

GetLldpInfo

GetLldpInfo 方法返回类似于以下示例的响应。

```
{
  "id": null,
  "result": {
    "lldpInfo": {
      "lldpChassis": {
        "local-chassis": [
          {
            "chassis": [
              {
                "capability": [
                  {
                    "enabled": false,
                    "type": "Bridge"
                  },
                  {
                    "enabled": false,
                    "type": "Router"
                  },
                  {
                    "enabled": false,
                    "type": "Wlan"
                  },
                  {

```

```
        "enabled": true,
        "type": "Station"
      }
    ],
    "descr": [
      {
        "value": "Element OS 11.0"
      }
    ],
    "id": [
      {
        "type": "mac",
        "value": "08:00:27:3c:0a:f4"
      }
    ],
    "mgmt-ip": [
      {
        "value": "10.0.2.15"
      },
      {
        "value": "fe80::a00:27ff:fe3c:af4"
      }
    ],
    "name": [
      {
        "value": "SF-93FF"
      }
    ]
  ]
},
"lldp-med": [
  {
    "capability": [
      {
        "available": true,
        "type": "Capabilities"
      },
      {
        "available": true,
        "type": "Policy"
      },
      {
        "available": true,
        "type": "Location"
      },
      {
        "available": true,
        "type": "MDI/PSE"
      },
      {
        "available": true,
        "type": "MDI/PD"
      },
      {
        "available": true,
        "type": "Inventory"
      }
    ]
  }
],
"device-type": [
  {
```

```
        "value": "Generic Endpoint (Class I)"
      }
    ],
    "inventory": [
      {
        "firmware": [
          {
            "value": "VirtualBox"
          }
        ],
        "hardware": [
          {
            "value": "1.2"
          }
        ],
        "manufacturer": [
          {
            "value": "innotek GmbH"
          }
        ],
        "model": [
          {
            "value": "VirtualBox"
          }
        ],
        "serial": [
          {
            "value": "0"
          }
        ],
        "software": [
          {
            "value": "4.14.27-solidfire2"
          }
        ]
      }
    ]
  }
}
],
},
"lldpInterfaces": {
  "lldp": [
    {
      "interface": [
        {
          "age": "0 day, 00:01:04",
          "chassis": [
            {
              "capability": [
                {
                  "enabled": false,
                  "type": "Bridge"
                },
                {
                  "enabled": false,
                  "type": "Router"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```
        "enabled": false,
        "type": "Wlan"
      },
      {
        "enabled": true,
        "type": "Station"
      }
    ],
    "descr": [
      {
        "value": "Element OS 11.0"
      }
    ],
    "id": [
      {
        "type": "mac",
        "value": "08:00:27:3c:0a:f4"
      }
    ],
    "mgmt-ip": [
      {
        "value": "10.0.2.15"
      },
      {
        "value": "fe80::a00:27ff:fe3c:af4"
      }
    ],
    "name": [
      {
        "value": "SF-93FF"
      }
    ]
  }
],
"lldp-med": [
  {
    "capability": [
      {
        "available": true,
        "type": "Capabilities"
      },
      {
        "available": true,
        "type": "Policy"
      },
      {
        "available": true,
        "type": "Location"
      },
      {
        "available": true,
        "type": "MDI/PSE"
      },
      {
        "available": true,
        "type": "MDI/PD"
      },
      {
        "available": true,
        "type": "Inventory"
      }
    ]
  }
]
```

```
    }
  ],
  "device-type": [
    {
      "value": "Generic Endpoint (Class I)"
    }
  ],
  "inventory": [
    {
      "firmware": [
        {
          "value": "VirtualBox"
        }
      ],
      "hardware": [
        {
          "value": "1.2"
        }
      ],
      "manufacturer": [
        {
          "value": "innotek GmbH"
        }
      ],
      "model": [
        {
          "value": "VirtualBox"
        }
      ],
      "serial": [
        {
          "value": "0"
        }
      ],
      "software": [
        {
          "value": "4.14.27-solidfire2"
        }
      ]
    }
  ]
}
],
"name": "eth0",
"port": [
  {
    "aggregation": [
      {
        "value": "7"
      }
    ]
  },
  "auto-negotiation": [
    {
      "advertised": [
        {
          "fd": true,
          "hd": true,
          "type": "10Base-T"
        }
      ],
    }
  ]
}
```

```
        "fd": true,
        "hd": true,
        "type": "100Base-TX"
      },
      {
        "fd": true,
        "hd": false,
        "type": "1000Base-T"
      }
    ],
    "current": [
      {
        "value": "full duplex mode"
      }
    ],
    "enabled": true,
    "supported": true
  }
],
"descr": [
  {
    "value": "eth0"
  }
],
"id": [
  {
    "type": "mac",
    "value": "08:00:27:3c:0a:f4"
  }
]
}
],
"ttl": [
  {
    "ttl": "120"
  }
],
"via": "unknown"
},
{
  "age": "17722 days, 17:14:28",
  "chassis": [
    {
      "capability": [
        {
          "enabled": false,
          "type": "Bridge"
        },
        {
          "enabled": false,
          "type": "Router"
        },
        {
          "enabled": false,
          "type": "Wlan"
        },
        {
          "enabled": true,
          "type": "Station"
        }
      ]
    }
  ]
}
```

```
    ],  
    "descr": [  
      {  
        "value": "Element OS 11.0"  
      }  
    ],  
    "id": [  
      {  
        "type": "mac",  
        "value": "08:00:27:3c:0a:f4"  
      }  
    ],  
    "mgmt-ip": [  
      {  
        "value": "10.0.2.15"  
      },  
      {  
        "value": "fe80::a00:27ff:fe3c:af4"  
      }  
    ],  
    "name": [  
      {  
        "value": "SF-93FF"  
      }  
    ]  
  }  
],  
"lldp-med": [  
  {  
    "capability": [  
      {  
        "available": true,  
        "type": "Capabilities"  
      },  
      {  
        "available": true,  
        "type": "Policy"  
      },  
      {  
        "available": true,  
        "type": "Location"  
      },  
      {  
        "available": true,  
        "type": "MDI/PSE"  
      },  
      {  
        "available": true,  
        "type": "MDI/PD"  
      },  
      {  
        "available": true,  
        "type": "Inventory"  
      }  
    ],  
    "device-type": [  
      {  
        "value": "Generic Endpoint (Class I)"  
      }  
    ]  
  },  
  ]  
],
```

```
    "inventory": [
      {
        "firmware": [
          {
            "value": "VirtualBox"
          }
        ],
        "hardware": [
          {
            "value": "1.2"
          }
        ],
        "manufacturer": [
          {
            "value": "innotek GmbH"
          }
        ],
        "model": [
          {
            "value": "VirtualBox"
          }
        ],
        "serial": [
          {
            "value": "0"
          }
        ],
        "software": [
          {
            "value": "4.14.27-solidfire2"
          }
        ]
      }
    ]
  },
  "name": "eth1",
  "port": [
    {
      "aggregation": [
        {
          "value": "7"
        }
      ],
      "auto-negotiation": [
        {
          "advertised": [
            {
              "fd": true,
              "hd": true,
              "type": "10Base-T"
            },
            {
              "fd": true,
              "hd": true,
              "type": "100Base-TX"
            },
            {
              "fd": true,
              "hd": false,
```

```
        "type": "1000Base-T"
      }
    ],
    "current": [
      {
        "value": "unknown"
      }
    ],
    "enabled": true,
    "supported": true
  }
],
"descr": [
  {
    "value": "eth1"
  }
],
"id": [
  {
    "type": "mac",
    "value": "08:00:27:36:79:78"
  }
]
}
],
"ttl": [
  {
    "ttl": "120"
  }
],
"via": "unknown"
},
{
  "age": "0 day, 00:01:01",
  "chassis": [
    {
      "capability": [
        {
          "enabled": false,
          "type": "Bridge"
        },
        {
          "enabled": false,
          "type": "Router"
        },
        {
          "enabled": false,
          "type": "Wlan"
        },
        {
          "enabled": true,
          "type": "Station"
        }
      ],
      "descr": [
        {
          "value": "Element OS 11.0"
        }
      ],
      "id": [
```

```
    {
      "type": "mac",
      "value": "08:00:27:3c:0a:f4"
    }
  ],
  "mgmt-ip": [
    {
      "value": "10.0.2.15"
    },
    {
      "value": "fe80::a00:27ff:fe3c:af4"
    }
  ],
  "name": [
    {
      "value": "SF-93FF"
    }
  ]
}
],
"lldp-med": [
  {
    "capability": [
      {
        "available": true,
        "type": "Capabilities"
      },
      {
        "available": true,
        "type": "Policy"
      },
      {
        "available": true,
        "type": "Location"
      },
      {
        "available": true,
        "type": "MDI/PSE"
      },
      {
        "available": true,
        "type": "MDI/PD"
      },
      {
        "available": true,
        "type": "Inventory"
      }
    ]
  },
  {
    "device-type": [
      {
        "value": "Generic Endpoint (Class I)"
      }
    ]
  },
  {
    "inventory": [
      {
        "firmware": [
          {
            "value": "VirtualBox"
          }
        ]
      }
    ]
  }
],
```

```
    "hardware": [
      {
        "value": "1.2"
      }
    ],
    "manufacturer": [
      {
        "value": "innotek GmbH"
      }
    ],
    "model": [
      {
        "value": "VirtualBox"
      }
    ],
    "serial": [
      {
        "value": "0"
      }
    ],
    "software": [
      {
        "value": "4.14.27-solidfire2"
      }
    ]
  }
]
}
],
"name": "eth2",
"port": [
  {
    "aggregation": [
      {
        "value": "6"
      }
    ],
    "auto-negotiation": [
      {
        "advertised": [
          {
            "fd": true,
            "hd": true,
            "type": "10Base-T"
          },
          {
            "fd": true,
            "hd": true,
            "type": "100Base-TX"
          },
          {
            "fd": true,
            "hd": false,
            "type": "1000Base-T"
          }
        ]
      }
    ],
    "current": [
      {
        "value": "full duplex mode"
      }
    ]
  }
]
```

```
    ],
    "enabled": true,
    "supported": true
  }
],
"descr": [
  {
    "value": "eth2"
  }
],
"id": [
  {
    "type": "mac",
    "value": "08:00:27:fc:f0:a9"
  }
]
}
],
"ttl": [
  {
    "ttl": "120"
  }
],
"via": "LLDP"
},
{
  "age": "0 day, 00:01:01",
  "chassis": [
    {
      "capability": [
        {
          "enabled": false,
          "type": "Bridge"
        },
        {
          "enabled": false,
          "type": "Router"
        },
        {
          "enabled": false,
          "type": "Wlan"
        },
        {
          "enabled": true,
          "type": "Station"
        }
      ]
    },
    {
      "value": "Element OS 11.0"
    }
  ],
  "id": [
    {
      "type": "mac",
      "value": "08:00:27:3c:0a:f4"
    }
  ],
  "mgmt-ip": [
    {
```

```
        "value": "10.0.2.15"
      },
      {
        "value": "fe80::a00:27ff:fe3c:af4"
      }
    ],
    "name": [
      {
        "value": "SF-93FF"
      }
    ]
  }
],
"lldp-med": [
  {
    "capability": [
      {
        "available": true,
        "type": "Capabilities"
      },
      {
        "available": true,
        "type": "Policy"
      },
      {
        "available": true,
        "type": "Location"
      },
      {
        "available": true,
        "type": "MDI/PSE"
      },
      {
        "available": true,
        "type": "MDI/PD"
      },
      {
        "available": true,
        "type": "Inventory"
      }
    ],
    "device-type": [
      {
        "value": "Generic Endpoint (Class I)"
      }
    ],
    "inventory": [
      {
        "firmware": [
          {
            "value": "VirtualBox"
          }
        ],
        "hardware": [
          {
            "value": "1.2"
          }
        ],
        "manufacturer": [
          {
```

```
        "value": "innotek GmbH"
      }
    ],
    "model": [
      {
        "value": "VirtualBox"
      }
    ],
    "serial": [
      {
        "value": "0"
      }
    ],
    "software": [
      {
        "value": "4.14.27-solidfire2"
      }
    ]
  ]
}
],
"name": "eth3",
"port": [
  {
    "aggregation": [
      {
        "value": "6"
      }
    ],
    "auto-negotiation": [
      {
        "advertised": [
          {
            "fd": true,
            "hd": true,
            "type": "10Base-T"
          },
          {
            "fd": true,
            "hd": true,
            "type": "100Base-TX"
          },
          {
            "fd": true,
            "hd": false,
            "type": "1000Base-T"
          }
        ],
        "current": [
          {
            "value": "full duplex mode"
          }
        ],
        "enabled": true,
        "supported": true
      }
    ],
    "descr": [
      {
```

```
        "value": "eth3"
      }
    ],
    "id": [
      {
        "type": "mac",
        "value": "08:00:27:2c:e4:f8"
      }
    ]
  }
],
"ttl": [
  {
    "ttl": "120"
  }
],
"via": "LLDP"
}
]
}
},
"lldpNeighbors": {
  "lldp": [
    {
      "interface": [
        {
          "age": "0 day, 00:04:34",
          "chassis": [
            {
              "capability": [
                {
                  "enabled": true,
                  "type": "Bridge"
                },
                {
                  "enabled": true,
                  "type": "Router"
                },
                {
                  "enabled": true,
                  "type": "Wlan"
                },
                {
                  "enabled": false,
                  "type": "Station"
                }
              ],
            },
            "descr": [
              {
                "value": "x86_64"
              }
            ],
            "id": [
              {
                "type": "mac",
                "value": "50:7b:9d:2b:36:84"
              }
            ],
            "mgmt-ip": [
```

```
        {
          "value": "192.168.100.1"
        },
        {
          "value": "fe80::a58e:843:952e:d8eb"
        }
      ],
      "name": [
        {
          "value": "ConventionalWisdom.wlan.netapp.com"
        }
      ]
    }
  ],
  "name": "eth2",
  "port": [
    {
      "auto-negotiation": [
        {
          "current": [
            {
              "value": "full duplex mode"
            }
          ],
          "enabled": false,
          "supported": false
        }
      ],
      "descr": [
        {
          "value": "vboxnet1"
        }
      ],
      "id": [
        {
          "type": "mac",
          "value": "0a:00:27:00:00:01"
        }
      ],
      "ttl": [
        {
          "value": "120"
        }
      ]
    }
  ],
  "rid": "2",
  "via": "LLDP"
},
{
  "age": "0 day, 00:01:01",
  "chassis": [
    {
      "capability": [
        {
          "enabled": false,
          "type": "Bridge"
        },
        {
          "enabled": false,
```

```
        "type": "Router"
      },
      {
        "enabled": false,
        "type": "Wlan"
      },
      {
        "enabled": true,
        "type": "Station"
      }
    ],
    "descr": [
      {
        "value": "Element OS 11.0"
      }
    ],
    "id": [
      {
        "type": "mac",
        "value": "08:00:27:3c:0a:f4"
      }
    ],
    "mgmt-ip": [
      {
        "value": "10.0.2.15"
      },
      {
        "value": "fe80::a00:27ff:fe3c:af4"
      }
    ],
    "name": [
      {
        "value": "SF-93FF"
      }
    ]
  ]
},
"lldp-med": [
  {
    "capability": [
      {
        "available": true,
        "type": "Capabilities"
      },
      {
        "available": true,
        "type": "Policy"
      },
      {
        "available": true,
        "type": "Location"
      },
      {
        "available": true,
        "type": "MDI/PSE"
      },
      {
        "available": true,
        "type": "MDI/PD"
      }
    ],
  },
],
```

```
    {
      "available": true,
      "type": "Inventory"
    }
  ],
  "device-type": [
    {
      "value": "Generic Endpoint (Class I)"
    }
  ],
  "inventory": [
    {
      "firmware": [
        {
          "value": "VirtualBox"
        }
      ],
      "hardware": [
        {
          "value": "1.2"
        }
      ],
      "manufacturer": [
        {
          "value": "innotek GmbH"
        }
      ],
      "model": [
        {
          "value": "VirtualBox"
        }
      ],
      "serial": [
        {
          "value": "0"
        }
      ],
      "software": [
        {
          "value": "4.14.27-solidfire2"
        }
      ]
    }
  ]
}
],
"name": "eth2",
"port": [
  {
    "aggregation": [
      {
        "value": "6"
      }
    ],
    "auto-negotiation": [
      {
        "advertised": [
          {
            "fd": true,
            "hd": true,
```

```
        "type": "10Base-T"
      },
      {
        "fd": true,
        "hd": true,
        "type": "100Base-TX"
      },
      {
        "fd": true,
        "hd": false,
        "type": "1000Base-T"
      }
    ],
    "current": [
      {
        "value": "full duplex mode"
      }
    ],
    "enabled": true,
    "supported": true
  }
],
"descr": [
  {
    "value": "eth3"
  }
],
"id": [
  {
    "type": "mac",
    "value": "08:00:27:2c:e4:f8"
  }
],
"ttl": [
  {
    "value": "120"
  }
]
}
],
"rid": "1",
"via": "LLDP"
},
{
  "age": "0 day, 00:04:34",
  "chassis": [
    {
      "capability": [
        {
          "enabled": true,
          "type": "Bridge"
        },
        {
          "enabled": true,
          "type": "Router"
        },
        {
          "enabled": true,
          "type": "Wlan"
        }
      ]
    }
  ],
}
```

```
        {
          "enabled": false,
          "type": "Station"
        }
      ],
      "descr": [
        {
          "value": "x86_64"
        }
      ],
      "id": [
        {
          "type": "mac",
          "value": "50:7b:9d:2b:36:84"
        }
      ],
      "mgmt-ip": [
        {
          "value": "192.168.100.1"
        },
        {
          "value": "fe80::a58e:843:952e:d8eb"
        }
      ],
      "name": [
        {
          "value": ""
        }
      ]
    }
  ],
  "name": "eth3",
  "port": [
    {
      "auto-negotiation": [
        {
          "current": [
            {
              "value": "full duplex mode"
            }
          ],
          "enabled": false,
          "supported": false
        }
      ],
      "descr": [
        {
          "value": "vboxnet1"
        }
      ],
      "id": [
        {
          "type": "mac",
          "value": "0a:00:27:00:00:01"
        }
      ],
      "ttl": [
        {
          "value": "120"
        }
      ]
    }
  ]
}
```

```
    ]
  }
],
"rid": "2",
"via": "LLDP"
},
{
  "age": "0 day, 00:01:01",
  "chassis": [
    {
      "capability": [
        {
          "enabled": false,
          "type": "Bridge"
        },
        {
          "enabled": false,
          "type": "Router"
        },
        {
          "enabled": false,
          "type": "Wlan"
        },
        {
          "enabled": true,
          "type": "Station"
        }
      ],
      "descr": [
        {
          "value": "Element OS 11.0"
        }
      ],
      "id": [
        {
          "type": "mac",
          "value": "08:00:27:3c:0a:f4"
        }
      ],
      "mgmt-ip": [
        {
          "value": "10.0.2.15"
        },
        {
          "value": "fe80::a00:27ff:fe3c:af4"
        }
      ],
      "name": [
        {
          "value": "SF-93FF"
        }
      ]
    }
  ],
  "lldp-med": [
    {
      "capability": [
        {
          "available": true,
          "type": "Capabilities"
        }
      ]
    }
  ]
}
```

```
    },
    {
      "available": true,
      "type": "Policy"
    },
    {
      "available": true,
      "type": "Location"
    },
    {
      "available": true,
      "type": "MDI/PSE"
    },
    {
      "available": true,
      "type": "MDI/PD"
    },
    {
      "available": true,
      "type": "Inventory"
    }
  ],
  "device-type": [
    {
      "value": "Generic Endpoint (Class I)"
    }
  ],
  "inventory": [
    {
      "firmware": [
        {
          "value": "VirtualBox"
        }
      ],
      "hardware": [
        {
          "value": "1.2"
        }
      ],
      "manufacturer": [
        {
          "value": "innotek GmbH"
        }
      ],
      "model": [
        {
          "value": "VirtualBox"
        }
      ],
      "serial": [
        {
          "value": "0"
        }
      ],
      "software": [
        {
          "value": "4.14.27-solidfire2"
        }
      ]
    }
  ]
}
```

```
    ]
  }
],
"name": "eth3",
"port": [
  {
    "aggregation": [
      {
        "value": "6"
      }
    ],
    "auto-negotiation": [
      {
        "advertised": [
          {
            "fd": true,
            "hd": true,
            "type": "10Base-T"
          },
          {
            "fd": true,
            "hd": true,
            "type": "100Base-TX"
          },
          {
            "fd": true,
            "hd": false,
            "type": "1000Base-T"
          }
        ],
        "current": [
          {
            "value": "1000BaseTFD"
          }
        ],
        "enabled": true,
        "supported": true
      }
    ],
    "descr": [
      {
        "value": "eth2"
      }
    ],
    "id": [
      {
        "type": "mac",
        "value": "08:00:27:fc:f0:a9"
      }
    ],
    "ttl": [
      {
        "value": "120"
      }
    ]
  }
],
"rid": "1",
"via": "LLDP"
}
```

```
    ]
  }
]
}
}
```

GetNetworkConfig

GetNetworkConfig 方法返回类似于以下示例的响应。

```
{
  "id": 1,
  "result": {
    "network": {
      "Bond10G": {
        "#default": false,
        "address": "10.1.1.0",
        "auto": true,
        "bond-downndelay": "0",
        "bond-fail_over_mac": "None",
        "bond-miimon": "100",
        "bond-mode": "ActivePassive",
        "bond-primary_reselect": "Failure",
        "bond-slaves": "eth0 eth1",
        "bond-updelay": "200",
        "dns-nameservers": "10.1.1.0, 10.1.1.0",
        "dns-search": "ten.test.company.net., company.net.",
        "family": "inet",
        "gateway": "10.1.1.0",
        "linkSpeed": 10000,
        "macAddress": "c8:1f:66:ee:59:b9",
        "macAddressPermanent": "00:00:00:00:00:00",
        "method": "static",
        "mtu": "9000",
        "netmask": "255.255.240.0",
        "network": "10.1.1.0",
        "physical": {
          "address": "10.1.1.0",
          "macAddress": "c8:1f:66:ee:59:b9",
          "macAddressPermanent": "00:00:00:00:00:00",
          "mtu": "9000",
          "netmask": "255.255.240.0",
          "network": "10.1.1.0",
          "upAndRunning": true
        },
        "routes": [],
        "status": "UpAndRunning",
        "symmetricRouteRules": [
          "ip route add 10.1.1.1/20 dev Bond1G src 10.1.2.2 table Bond1G",
          "ip rule add from 10.1.1.1 table Bond1G",
          "ip route add default via 10.1.1.254"
        ],
        "upAndRunning": true,
        "virtualNetworkTag": "0"
      },
      "Bond1G": {
        "#default": true,
        "address": "10.1.1.0",
```

```
    "addressV6": "",
    "auto": true,
    "bond-downdelay": "0",
    "bond-fail_over_mac": "None",
    "bond-miimon": "100",
    "bond-mode": "ActivePassive",
    "bond-primary_reselect": "Failure",
    "bond-slaves": "eth2 eth3",
    "bond-updelay": "200",
    "dns-nameservers": "10.1.1.0, 10.1.1.0",
    "dns-search": "ten.test.company.net., company.net.",
    "family": "inet",
    "gateway": "10.1.1.254",
    "gatewayV6": "",
    "linkSpeed": 1000,
    "macAddress": "c8:1f:66:ee:59:bd",
    "macAddressPermanent": "00:00:00:00:00:00",
    "method": "static",
    "mtu": "1500",
    "netmask": "255.255.240.0",
    "network": "10.1.1.0",
    "physical": {
      "address": "10.1.1.0",
      "macAddress": "c8:1f:66:ee:59:bd",
      "macAddressPermanent": "00:00:00:00:00:00",
      "mtu": "1500",
      "netmask": "255.255.240.0",
      "network": "10.1.1.0",
      "upAndRunning": true
    },
    "routes": [],
    "status": "UpAndRunning",
    "symmetricRouteRules": [
      "ip route add 10.1.1.1/20 dev Bond1G src 10.1.2.2 table Bond1G",
      "ip rule add from 10.1.1.1 table Bond1G",
      "ip route add default via 10.1.1.254"
    ],
    "upAndRunning": true,
    "virtualNetworkTag": "0"
  },
  "eth0": {
    "auto": true,
    "bond-master": "Bond10G",
    "family": "inet",
    "linkSpeed": 10000,
    "macAddress": "c8:1f:66:ee:59:b9",
    "macAddressPermanent": "c8:1f:66:ee:59:b9",
    "method": "bond",
    "physical": {
      "address": "0.0.0.0",
      "macAddress": "c8:1f:66:ee:59:b9",
      "macAddressPermanent": "c8:1f:66:ee:59:b9",
      "netmask": "N/A",
      "network": "N/A",
      "upAndRunning": true
    },
    "status": "UpAndRunning",
    "upAndRunning": true
  },
  "eth1": {
```

```
    "auto": true,
    "bond-master": "Bond10G",
    "family": "inet",
    "linkSpeed": 10000,
    "macAddress": "c8:1f:66:ee:59:b9",
    "macAddressPermanent": "c8:1f:66:ee:59:bb",
    "method": "bond",
    "physical": {
      "address": "0.0.0.0",
      "macAddress": "c8:1f:66:ee:59:b9",
      "macAddressPermanent": "c8:1f:66:ee:59:bb",
      "netmask": "N/A",
      "network": "N/A",
      "upAndRunning": true
    },
    "status": "UpAndRunning",
    "upAndRunning": true
  },
  "eth2": {
    "auto": true,
    "bond-master": "Bond1G",
    "family": "inet",
    "linkSpeed": 1000,
    "macAddress": "c8:1f:66:ee:59:bd",
    "macAddressPermanent": "c8:1f:66:ee:59:bd",
    "method": "bond",
    "physical": {
      "address": "0.0.0.0",
      "macAddress": "c8:1f:66:ee:59:bd",
      "macAddressPermanent": "c8:1f:66:ee:59:bd",
      "netmask": "N/A",
      "network": "N/A",
      "upAndRunning": true
    },
    "status": "UpAndRunning",
    "upAndRunning": true
  },
  "eth3": {
    "auto": true,
    "bond-master": "Bond1G",
    "family": "inet",
    "linkSpeed": 1000,
    "macAddress": "c8:1f:66:ee:59:bd",
    "macAddressPermanent": "c8:1f:66:ee:59:bf",
    "method": "bond",
    "physical": {
      "address": "0.0.0.0",
      "macAddress": "c8:1f:66:ee:59:bd",
      "macAddressPermanent": "c8:1f:66:ee:59:bf",
      "netmask": "N/A",
      "network": "N/A",
      "upAndRunning": true
    },
    "status": "UpAndRunning",
    "upAndRunning": true
  },
  "lo": {
    "auto": true,
    "family": "inet",
    "linkSpeed": 0,
```

```
        "macAddress": "00:00:00:00:00:00",
        "macAddressPermanent": "00:00:00:00:00:00",
        "method": "loopback",
        "physical": {
            "address": "0.0.0.0",
            "macAddress": "00:00:00:00:00:00",
            "macAddressPermanent": "00:00:00:00:00:00",
            "netmask": "N/A",
            "network": "N/A",
            "upAndRunning": true
        },
        "status": "UpAndRunning",
        "upAndRunning": true
    }
}
}
```

GetNodeHardwareInfo (iSCSI 的输出)

iSCSI 的 GetNodeHardwareInfo 方法返回类似于以下示例的响应。

```
{
  "id": 1,
  "result": {
    "nodeHardwareInfo": {
      "bus": {
        "core_DMI:0200": {
          "description": "Motherboard",
          "physid": "0",
          "product": "0H47HH",
          "serial": "..CN7475141I0271.",
          "vendor": "SolidFire",
          "version": "A07"
        }
      }
    },
    "driveHardware": [
      {
        "canonicalName": "sda",
        "connected": true,
        "dev": 2048,
        "devPath": "/dev/slot0",
        "driveEncryptionCapability": "fips",
        "driveType": "Slice",
        "lifeRemainingPercent": 98,
        "lifetimeReadBytes": 0,
        "lifetimeWriteBytes": 14012129342144,
        "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"
  },
  {
    "canonicalName": "sda",
    "connected": true,
    "dev": 2048,
    "devPath": "/dev/slot1",
    "driveEncryptionCapability": "fips",
    "driveType": "Slice",
    "lifeRemainingPercent": 98,
    "lifetimeReadBytes": 0,
    "lifetimeWriteBytes": 14112129567184,
    "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": "S1M9NWAG501252",
    "size": 240057409536,
    "slot": 0,
    "uncorrectableErrors": 0,
    "uuid": "789aa05d-e49b-ff4f-f821-f60eed8e43bd",
    "vendor": "Samsung",
    "version": "EXT1303Q"
  }
}
```

GetNodeHardwareInfo（光纤通道节点的输出）

光纤通道节点的 GetNodeHardwareInfo 方法返回类似于以下示例的响应。

```
{
  "id": null,
  "result": {
    "nodeHardwareInfo": {
      "bus": {
        "core_DMI:0200": {
          "description": "Motherboard",
          "physid": "0",
          "product": "0H47HH",
          "serial": "..CN747513AA0541.",
          "version": "A07"
        },
        "fiber:0_PCI:0000:04:00.0": {
```

```
"businfo": "pci@0000:04:00.0",
"clock": "33000000",
"description": "Fibre Channel",
"physid": "0",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
},
"fiber:0_PCI:0000:42:00.0": {
"businfo": "pci@0000:42:00.0",
"clock": "33000000",
"description": "Fibre Channel",
"physid": "0",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
},
"fiber:1_PCI:0000:04:00.1": {
"businfo": "pci@0000:04:00.1",
"clock": "33000000",
"description": "Fibre Channel",
"physid": "0.1",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
},
"fiber:1_PCI:0000:42:00.1": {
"businfo": "pci@0000:42:00.1",
"clock": "33000000",
"description": "Fibre Channel",
"physid": "0.1",
"product": "ISP8324-based 16Gb Fibre Channel to PCI Express Adapter",
"vendor": "QLogic Corp.",
"version": "02",
"width": "64"
}
},
"fans": {
"Fan1A RPM": {
"baseUnit": "RPM",
"threshold": 840,
"value": 3360
},
"Fan1B RPM": {
"baseUnit": "RPM",
"threshold": 840,
"value": 3120
}
},
"fibreChannelPorts": [
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 1,
"internalPortID": 2,
"model": "QLE2672",
"nPortID": "0x060019",
"nodeID": 6,
```

```
"pciSlot": 3,
"serial": "BFE1335E04217",
"speed": "8 Gbit",
"state": "Online",
"switchWwn": "20:01:00:2a:6a:9c:71:01",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:0a"
},
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 2,
"internalPortID": 3,
"model": "QLE2672",
"nPortID": "0xc70019",
"nodeID": 6,
"pciSlot": 3,
"serial": "BFE1335E04217",
"speed": "8 Gbit",
"state": "Online",
"switchWwn": "20:01:00:2a:6a:98:a3:41",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:0b"
},
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 1,
"internalPortID": 0,
"model": "QLE2672",
"nPortID": "0xc70017",
"nodeID": 6,
"pciSlot": 2,
"serial": "BFE1341E09515",
"speed": "8 Gbit",
"state": "Online",
"switchWwn": "20:01:00:2a:6a:98:a3:41",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:08"
},
{
"firmware": "7.04.00 (d0d5)",
"hbaPort": 2,
"internalPortID": 1,
"model": "QLE2672",
"nPortID": "0x060017",
"nodeID": 6,
"pciSlot": 2,
"serial": "BFE1341E09515",
"speed": "8 Gbit",
"state": "Online",
"switchWwn": "20:01:00:2a:6a:9c:71:01",
"wwnn": "5f:47:ac:c8:30:26:c9:00",
"wwpn": "5f:47:ac:c0:30:26:c9:09"
}
],
"memory": {
"firmware_": {
"capacity": "8323072",
"date": "08/29/2013",
"description": "BIOS",
"physid": "0",
```

```
"size": "65536",
"version": "2.0.19"
},
"memory_DMI:1000": {
"description": "System Memory",
"physid": "1000",
"size": "34359738368",
"slot": "System board or motherboard"
},
"network": {
"network:0_": {
"description": "Ethernet interface",
"logicalname": "Bond1G",
"physid": "1",
"serial": "c8:1f:66:df:04:da"
},
"network:0_PCI:0000:01:00.0": {
"businfo": "pci@0000:01:00.0",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth0",
"physid": "0",
"product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
"serial": "c8:1f:66:df:04:d6",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
},
"network:0_PCI:0000:41:00.0": {
"businfo": "pci@0000:41:00.0",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth4",
"physid": "0",
"product": "NetXtreme II BCM57810 10 Gigabit Ethernet",
"serial": "00:0a:f7:41:7a:30",
"vendor": "Broadcom Corporation",
"version": "10",
"width": "64"
},
"network:1_": {
"description": "Ethernet interface",
"logicalname": "Bond10G",
"physid": "2",
"serial": "c8:1f:66:df:04:d6"
},
"network:1_PCI:0000:01:00.1": {
"businfo": "pci@0000:01:00.1",
"capacity": "1000000000",
"clock": "33000000",
"description": "Ethernet interface",
"logicalname": "eth1",
"physid": "0.1",
"product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
"serial": "c8:1f:66:df:04:d8",
"vendor": "Broadcom Corporation",
"version": "10",
```

```
"width": "64"
},
"network:1_PCI:0000:41:00.1": {
  "businfo": "pci@0000:41:00.1",
  "capacity": "1000000000",
  "clock": "33000000",
  "description": "Ethernet interface",
  "logicalname": "eth5",
  "physid": "0.1",
  "product": "NetXtreme II BCM57810 10 Gigabit Ethernet",
  "serial": "00:0a:f7:41:7a:32",
  "vendor": "Broadcom Corporation",
  "version": "10",
  "width": "64"
},
"network:2_PCI:0000:01:00.2": {
  "businfo": "pci@0000:01:00.2",
  "capacity": "1000000000",
  "clock": "33000000",
  "description": "Ethernet interface",
  "logicalname": "eth2",
  "physid": "0.2",
  "product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
  "serial": "c8:1f:66:df:04:da",
  "size": "1000000000",
  "vendor": "Broadcom Corporation",
  "version": "10",
  "width": "64"
},
"network:3_PCI:0000:01:00.3": {
  "businfo": "pci@0000:01:00.3",
  "capacity": "1000000000",
  "clock": "33000000",
  "description": "Ethernet interface",
  "logicalname": "eth3",
  "physid": "0.3",
  "product": "NetXtreme II BCM57800 1/10 Gigabit Ethernet",
  "serial": "c8:1f:66:df:04:dc",
  "size": "1000000000",
  "vendor": "Broadcom Corporation",
  "version": "10",
  "width": "64"
}
},
"networkInterfaces": {
  "Bond10G": {
    "isConfigured": true,
    "isUp": true
  },
  "Bond1G": {
    "isConfigured": true,
    "isUp": true
  },
  "eth0": {
    "isConfigured": true,
    "isUp": true
  },
  "eth1": {
    "isConfigured": true,
    "isUp": true
  }
}
```

```
  },
  "eth2": {
    "isConfigured": true,
    "isUp": true
  },
  "eth3": {
    "isConfigured": true,
    "isUp": true
  },
  "eth4": {
    "isConfigured": true,
    "isUp": true
  },
  "eth5": {
    "isConfigured": true,
    "isUp": true
  }
},
"platform": {
  "chassisType": "R620",
  "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
  "nodeMemoryGB": 32,
  "nodeType": "SFFC"
},
"powerSupplies": {
  "PS1 status": {
    "powerSupplyFailureDetected": false,
    "powerSupplyHasAC": true,
    "powerSupplyPredictiveFailureDetected": false,
    "powerSupplyPresent": true
  },
  "PS2 status": {
    "powerSupplyFailureDetected": false,
    "powerSupplyHasAC": true,
    "powerSupplyPredictiveFailureDetected": false,
    "powerSupplyPresent": true
  }
},
"storage": {
  "storage_PCI:0000:00:1f.2": {
    "businfo": "pci@0000:00:1f.2",
    "clock": "66000000",
    "description": "SATA controller",
    "physid": "1f.2",
    "product": "C600/X79 series chipset 6-Port SATA AHCI Controller",
    "vendor": "Intel Corporation",
    "version": "05",
    "width": "32"
  }
},
"system": {
  "fcv-2_DMI:0100": {
    "description": "Rack Mount Chassis",
    "product": "(SKU=NotProvided;ModelName=)",
    "serial": "HTX1DZ1",
    "width": "64"
  }
},
"temperatures": {
  "Exhaust Temp": {
```

```
"baseUnit": "C",
"threshold": 70,
"value": 38
},
"Inlet Temp": {
"baseUnit": "C",
"threshold": 42,
"value": 13
},
"uuid": "4C4C4544-004D-5310-8052-C4C04F335431"
}
}
}
}
```

GetNvramInfo

GetNvramInfo 方法返回类似于以下示例的响应。

```
{
  id: 1,
  result: {
    nvramInfo: {
      details: {
        errors: {
          numOfErrorLogEntries: "0"
        },
        extended: {
          dialogVersion: "4",
          event: [
            {
              name: "flushToFlash",
              time: "2014-02-24 20:30:28",
              value: "0"
            },
            {
              name: "flushToFlash",
              time: "1946-02-06 17:16:42",
              value: "0"
            },
            {
              name: "flushToFlash",
              time: "2014-02-25 00:48:06",
              value: "0"
            },
            {
              name: "flushToFlash",
              time: "2014-02-25 15:44:07",
              value: "0"
            },
            {
              name: "flushToFlash",
              time: "2014-03-17 17:21:46",
              value: "0"
            },
            {
              name: "flushToFlash",
              time: "2014-03-17 17:59:30",
              value: "0"
            }
          ]
        }
      }
    }
  }
}
```

```
    },
    {
      name: "flushToFlash",
      time: "2014-03-17 18:06:27",
      value: "0"
    },
    {
      name: "flushToFlash",
      time: "2014-03-17 21:43:17",
      value: "0"
    },
    {
      name: "excessiveCurrent",
      time: "2014-02-25 00:00:29",
      value: "39"
    },
    {
      name: "excessiveCurrent",
      time: "2014-03-01 00:00:24",
      value: "23"
    }
  ],
  eventOccurrences: [
    {
      count: "15",
      name: "flushToFlash"
    },
    {
      count: "2",
      name: "excessiveCurrent"
    }
  ],
  initialCapacitance: "6.653 F",
  initialEsr: "0.097 Ohm",
  measurement: [
    {
      level_0: " 0",
      level_1: " 112",
      level_2: " 670919",
      level_3: " 455356",
      level_4: " 90215",
      level_5: " 0",
      level_6: " 0",
      level_7: " 0",
      level_8: " 0",
      level_9: " 0",
      name: "enterpriseFlashControllerTemperature",
      recent: "64 C"
    },
    {
      level_0: " 0",
      level_1: " 27",
      level_2: " 456896",
      level_3: " 717565",
      level_4: " 39422",
      level_5: " 2692",
      level_6: " 0",
      level_7: " 0",
      level_8: " 0",
      level_9: " 0",
      name: "capacitor1And2Temperature",
    }
  ]
}
```

```
    recent: "28.64 C"
  },
  {
    level_0: " 0",
    level_1: " 2080",
    level_2: " 907196",
    level_3: " 280178",
    level_4: " 26539",
    level_5: " 609",
    level_6: " 0",
    level_7: " 0",
    level_8: " 0",
    level_9: " 0",
    name: "capacitor3And4Temperature",
    recent: "28.60 C"
  },
  {
    errorPeriod: {
      duration: "24",
      startTime: "2014-02-06 00:23:54",
      worst: "8"
    },
    level_0: " 0",
    level_1: " 839",
    level_2: " 272794",
    level_3: " 404758",
    level_4: " 35216",
    level_5: " 377818",
    level_6: " 103891",
    level_7: " 21274",
    level_8: " 12",
    level_9: " 0",
    name: "rearVentAmbientTemperature",
    recent: "46.82 C"
  },
  {
    level_0: " 0",
    level_1: " 742749",
    level_2: " 460016",
    level_3: " 13837",
    level_4: " 0",
    level_5: " 0",
    level_6: " 0",
    level_7: " 0",
    level_8: " 0",
    level_9: " 0",
    name: "rms200BoardTemperature",
    recent: "50.62 C"
  },
  {
    name: "voltageOfCapacitor1",
    recent: "2.308 V"
  },
  {
    name: "voltageOfCapacitor2",
    recent: "2.305 V"
  },
  {
    name: "voltageOfCapacitor3",
    recent: "2.314 V"
  },
},
```

```
{
  name: "voltageOfCapacitor4",
  recent: "2.307 V"
},
{
  level_0: " 175052",
  level_1: " 51173",
  level_2: " 435788",
  level_3: " 12766",
  level_4: " 4",
  level_5: " 6",
  level_6: " 541813",
  level_7: " 0",
  level_8: " 0",
  level_9: " 0",
  name: "capacitorPackVoltage",
  recent: "9.233 V"
},
{
  level_0: " 0",
  level_1: " 0",
  level_2: " 0",
  level_3: " 0",
  level_4: " 0",
  level_5: " 0",
  level_6: " 4",
  level_7: " 1",
  level_8: " 4",
  level_9: " 6",
  name: "capacitorPackVoltageAtEndOfFlushToFlash",
  recent: "5.605 V"
},
{
  name: "currentDerivedFromV3V4",
  recent: "0.000 A"
},
{
  level_0: " 7",
  level_1: " 4",
  level_2: " 3",
  level_3: " 1",
  level_4: " 0",
  level_5: " 0",
  level_6: " 0",
  level_7: " 0",
  level_8: " 0",
  level_9: " 0",
  name: "derivedEnergy",
  recent: "175 Joules"
},
{
  level_0: " 0",
  level_1: " 0",
  level_2: " 0",
  level_3: " 0",
  level_4: " 0",
  level_5: " 0",
  level_6: " 0",
  level_7: " 17",
  level_8: " 19",
```

```
    level_9: " 7",
    name: "derivedCapacitanceOfThePack",
    recent: "5.959 F"
  },
  {
    level_0: " 0",
    level_1: " 43",
    level_2: " 0",
    level_3: " 0",
    level_4: " 0",
    level_5: " 0",
    level_6: " 0",
    level_7: " 0",
    level_8: " 0",
    level_9: " 0",
    name: "derivedEsrOfCapacitorPack",
    recent: "0.104 Ohm"
  },
  {
    level_0: " 0",
    level_1: " 0",
    level_2: " 0",
    level_3: " 0",
    level_4: " 15",
    level_5: " 0",
    level_6: " 0",
    level_7: " 0",
    level_8: " 0",
    level_9: " 0",
    name: "timeToRunFlushToFlash",
    recent: "22.40 Seconds"
  },
  {
    level_0: " 0",
    level_1: " 0",
    level_2: " 7",
    level_3: " 0",
    level_4: " 0",
    level_5: " 0",
    level_6: " 0",
    level_7: " 0",
    level_8: " 0",
    level_9: " 0",
    name: "timeToRunRestore",
    recent: "20.44 Seconds"
  },
  {
    level_0: " 0",
    level_1: " 1",
    level_2: " 3",
    level_3: " 2",
    level_4: " 0",
    level_5: " 0",
    level_6: " 0",
    level_7: " 0",
    level_8: " 0",
    level_9: " 1",
    name: "timeToChargeCapacitors",
    recent: "48 Seconds"
  },
},
```

```
{
  level_0: " 448586",
  level_1: " 2998",
  level_2: " 0",
  level_3: " 0",
  level_4: " 0",
  level_5: " 0",
  level_6: " 0",
  level_7: " 0",
  level_8: " 0",
  level_9: " 0",
  name: "correctableBitsInErrorOnReadingAPage"
},
{
  level_0: " 2998",
  level_1: " 0",
  level_2: " 0",
  level_3: " 0",
  level_4: " 0",
  level_5: " 0",
  level_6: " 0",
  level_7: " 0",
  level_8: " 0",
  level_9: " 0",
  name: "correctableBitsInErrorOnReadingTheWorstBchRegionOfAPage"
},
{
  level_0: " 0",
  level_1: " 37",
  level_2: " 280274",
  level_3: " 422999",
  level_4: " 245814",
  level_5: " 242470",
  level_6: " 24447",
  level_7: " 561",
  level_8: " 0",
  level_9: " 0",
  name: "fanInletAmbientTemperature",
  recent: "41.74 C"
},
],
  predictedCapacitanceDepletion: "504328 uF",
  smartCounters: [
    {
      name: "numberOf512ByteBlocksReadFromDdr",
      value: "218284648"
    },
    {
      name: "numberOf512ByteBlocksWrittenToDdr",
      value: "12031567354"
    },
    {
      name: "numberOfHostReadCommands",
      value: "5366315"
    },
    {
      name: "numberOfHostWriteCommands",
      value: "1266099334"
    }
  ]
}
```

```
        name: "controllerBusyTimeMinutes",
        value: "0"
      },
      {
        name: "numberOfPowerCycles",
        value: "13"
      },
      {
        name: "powerOnHours",
        value: "1009"
      },
      {
        name: "unsafeShutdowns",
        value: "5"
      },
      {
        name: "mediaErrors",
        value: "0"
      },
      {
        name: "numberOfErrorLogs",
        value: "2"
      }
    ],
    snapshotTime: "2014-03-20 16:43:49"
  },
  firmware: {
    activeSlotNumber: "2",
    slot1Version: "1e5817bc",
    slot2Version: "1e0d70ac",
    slot3Version: "1e5817bc",
    slot4Version: "1e5817bc"
  },
  smart: {
    availableSpace: "0%",
    availableSpaceThreshold: "0%",
    controllerBusyTimeMinutes: "0",
    criticalErrorVector: "0x0",
    mediaErrors: "0",
    numberOf512ByteBlocksRead: "218284648",
    numberOf512ByteBlocksWritten: "12031567354",
    numberOfErrorInfoLogs: "2",
    numberOfHostReadCommands: "5366315",
    numberOfHostWriteCommands: "1266099334",
    numberOfPowerCycles: "13",
    powerOnHours: "1009",
    temperature: "323 Kelvin",
    unsafeShutdowns: "5"
  }
},
status: "Warning",
statusInfo: {
  warning: [
    "excessiveCurrent (2x)"
  ]
},
type: "RMS-200"
}
```

ListActiveNodes

ListActiveNodes 方法返回类似于以下示例的响应。

```
{
  "id": 1,
  "result": {
    "nodes": [
      {
        "associatedFServiceID": 0,
        "associatedMasterServiceID": 1,
        "attributes": {},
        "cip": "172.27.21.23",
        "cipi": "Bond10G",
        "fibreChannelTargetPortGroup": null,
        "mip": "172.27.1.23",
        "mipi": "Bond1G",
        "name": "PSN-1-23",
        "nodeID": 1,
        "platformInfo": {
          "chassisType": "R620",
          "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
          "nodeMemoryGB": 72,
          "nodeType": "SF3010"
        },
        "sip": "172.27.21.23",
        "sipi": "Bond10G",
        "softwareVersion": "9.0.0.1298",
        "uuid": "4C4C4544-0056-3810-804E-B5C04F4C5631",
        "virtualNetworks": [
          {
            "address": "10.1.2.4",
            "virtualNetworkID": 1
          },
          {
            "address": "10.2.2.10",
            "virtualNetworkID": 2
          }
        ]
      },
      {
        "associatedFServiceID": 0,
        "associatedMasterServiceID": 4,
        "attributes": {},
        "cip": "172.27.21.24",
        "cipi": "Bond10G",
        "fibreChannelTargetPortGroup": null,
        "mip": "172.27.1.24",
        "mipi": "Bond1G",
        "name": "PSN-1-24",
        "nodeID": 2,
        "platformInfo": {
          "chassisType": "R620",
          "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
          "nodeMemoryGB": 72,
          "nodeType": "SF3010"
        },
        "sip": "172.27.21.24",
        "sipi": "Bond10G",
        "softwareVersion": "9.0.0.1298",

```

```
"uuid": "4C4C4544-0042-4210-804E-C3C04F4C5631",
"virtualNetworks": [
  {
    "address": "10.1.2.5",
    "virtualNetworkID": 1
  },
  {
    "address": "10.2.2.11",
    "virtualNetworkID": 2
  }
]
},
{
  "associatedFServiceID": 0,
  "associatedMasterServiceID": 2,
  "attributes": {},
  "cip": "172.27.21.25",
  "cipi": "Bond10G",
  "fibreChannelTargetPortGroup": null,
  "mip": "172.27.1.25",
  "mipi": "Bond1G",
  "name": "PSN-1-25",
  "nodeID": 3,
  "platformInfo": {
    "chassisType": "R620",
    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
    "nodeMemoryGB": 72,
    "nodeType": "SF3010"
  },
  "sip": "172.27.21.25",
  "sipi": "Bond10G",
  "softwareVersion": "9.0.0.1298",
  "uuid": "4C4C4544-0053-4210-8051-C6C04F515631",
  "virtualNetworks": [
    {
      "address": "10.1.2.6",
      "virtualNetworkID": 1
    },
    {
      "address": "10.2.2.12",
      "virtualNetworkID": 2
    }
  ]
},
{
  "associatedFServiceID": 0,
  "associatedMasterServiceID": 3,
  "attributes": {},
  "cip": "172.27.21.26",
  "cipi": "Bond10G",
  "fibreChannelTargetPortGroup": null,
  "mip": "172.27.1.26",
  "mipi": "Bond1G",
  "name": "PSN-1-26",
  "nodeID": 4,
  "platformInfo": {
    "chassisType": "R620",
    "cpuModel": "Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz",
    "nodeMemoryGB": 72,
    "nodeType": "SF3010"
  }
}
```

```
    },
    "sip": "172.27.21.26",
    "sipi": "Bond10G",
    "softwareVersion": "9.0.0.1298",
    "uuid": "4C4C4544-0056-3810-804E-B4C04F4C5631",
    "virtualNetworks": [
      {
        "address": "10.1.2.7",
        "virtualNetworkID": 1
      },
      {
        "address": "10.2.2.13",
        "virtualNetworkID": 2
      }
    ]
  }
}
}
```

ListActiveVolumes

ListActiveVolumes 方法返回类似于以下示例的响应。

```
{
  "id": 1,
  "result": {
    "volumes": [
      {
        "access": "readWrite",
        "accountID": 1,
        "attributes": {},
        "blockSize": 4096,
        "createTime": "2016-06-23T14:19:12Z",
        "deleteTime": "",
        "enable512e": false,
        "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo1.1",
        "name": "HulkDemo1",
        "purgeTime": "",
        "qos": {
          "burstIOPS": 1500,
          "burstTime": 60,
          "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
          },
          "maxIOPS": 1000,
          "minIOPS": 100
        },
        "scsiEUIDeviceID": "306f746f00000001f47acc0100000000",
        "scsiNAADeviceID": "6f47acc100000000306f746f00000001",
        "sliceCount": 1,
        "status": "active",
```

```
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
      1
    ],
    "volumeID": 1,
    "volumePairs": []
  },
  {
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:14Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo6.6",
    "name": "HulkDemo6",
    "purgeTime": "",
    "qos": {
      "burstIOPS": 1500,
      "burstTime": 60,
      "curve": {
        "4096": 100,
        "8192": 160,
        "16384": 270,
        "32768": 500,
        "65536": 1000,
        "131072": 1950,
        "262144": 3900,
        "524288": 7600,
        "1048576": 15000
      },
      "maxIOPS": 1000,
      "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f00000006f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000006",
    "sliceCount": 1,
    "status": "active",
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
      1
    ],
    "volumeID": 6,
    "volumePairs": []
  },
  {
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:14Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo7.7",
    "name": "HulkDemo7",
    "purgeTime": "",
    "qos": {
```

```
        "burstIOPS": 1500,
        "burstTime": 60,
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "maxIOPS": 1000,
        "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f00000007f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000007",
    "sliceCount": 1,
    "status": "active",
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
        1
    ],
    "volumeID": 7,
    "volumePairs": []
},
{
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:15Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo8.8",
    "name": "HulkDemo8",
    "purgeTime": "",
    "qos": {
        "burstIOPS": 1500,
        "burstTime": 60,
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "maxIOPS": 1000,
        "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f00000008f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000008",
    "sliceCount": 1,
    "status": "active",
```

```
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
      1
    ],
    "volumeID": 8,
    "volumePairs": []
  },
  {
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:15Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo9.9",
    "name": "HulkDemo9",
    "purgeTime": "",
    "qos": {
      "burstIOPS": 1500,
      "burstTime": 60,
      "curve": {
        "4096": 100,
        "8192": 160,
        "16384": 270,
        "32768": 500,
        "65536": 1000,
        "131072": 1950,
        "262144": 3900,
        "524288": 7600,
        "1048576": 15000
      },
      "maxIOPS": 1000,
      "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f00000009f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000009",
    "sliceCount": 1,
    "status": "active",
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
      1
    ],
    "volumeID": 9,
    "volumePairs": []
  },
  {
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:16Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo12.12",
    "name": "HulkDemo12",
    "purgeTime": "",
    "qos": {
```

```
        "burstIOPS": 1500,
        "burstTime": 60,
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "maxIOPS": 1000,
        "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f0000000cf47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f0000000c",
    "sliceCount": 1,
    "status": "active",
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
        1
    ],
    "volumeID": 12,
    "volumePairs": []
},
{
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:18Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo16.16",
    "name": "HulkDemo16",
    "purgeTime": "",
    "qos": {
        "burstIOPS": 1500,
        "burstTime": 60,
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "maxIOPS": 1000,
        "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f00000010f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000010",
    "sliceCount": 1,
    "status": "active",
```

```
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
      1
    ],
    "volumeID": 16,
    "volumePairs": []
  },
  {
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:18Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo17.17",
    "name": "HulkDemo17",
    "purgeTime": "",
    "qos": {
      "burstIOPS": 1500,
      "burstTime": 60,
      "curve": {
        "4096": 100,
        "8192": 160,
        "16384": 270,
        "32768": 500,
        "65536": 1000,
        "131072": 1950,
        "262144": 3900,
        "524288": 7600,
        "1048576": 15000
      },
      "maxIOPS": 1000,
      "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f00000011f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000011",
    "sliceCount": 1,
    "status": "active",
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
      1
    ],
    "volumeID": 17,
    "volumePairs": []
  },
  {
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-23T14:19:18Z",
    "deleteTime": "",
    "enable512e": false,
    "iqn": "iqn.2010-01.com.solidfire:0oto.hulkdemo18.18",
    "name": "HulkDemo18",
    "purgeTime": "",
    "qos": {
```

```
        "burstIOPS": 1500,
        "burstTime": 60,
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "maxIOPS": 1000,
        "minIOPS": 100
    },
    "scsiEUIDeviceID": "306f746f00000012f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000012",
    "sliceCount": 1,
    "status": "active",
    "totalSize": 53687091200,
    "virtualVolumeID": null,
    "volumeAccessGroups": [
        1
    ],
    "volumeID": 18,
    "volumePairs": []
},
{
    "access": "readWrite",
    "accountID": 1,
    "attributes": {},
    "blockSize": 4096,
    "createTime": "2016-06-24T15:21:59Z",
    "deleteTime": "",
    "enable512e": true,
    "iqn": "iqn.2010-01.com.solidfire:0oto.bk.24",
    "name": "BK",
    "purgeTime": "",
    "qos": {
        "burstIOPS": 15000,
        "burstTime": 60,
        "curve": {
            "4096": 100,
            "8192": 160,
            "16384": 270,
            "32768": 500,
            "65536": 1000,
            "131072": 1950,
            "262144": 3900,
            "524288": 7600,
            "1048576": 15000
        },
        "maxIOPS": 15000,
        "minIOPS": 50
    },
    "scsiEUIDeviceID": "306f746f00000018f47acc0100000000",
    "scsiNAADeviceID": "6f47acc100000000306f746f00000018",
    "sliceCount": 1,
    "status": "active",
```

```
"totalSize": 10737418240,
"virtualVolumeID": null,
"volumeAccessGroups": [],
"volumeID": 24,
"volumePairs": [
  {
    "clusterPairID": 2,
    "remoteReplication": {
      "mode": "Async",
      "pauseLimit": 3145728000,
      "remoteServiceID": 14,
      "resumeDetails": "",
      "snapshotReplication": {
        "state": "Idle",
        "stateDetails": ""
      }
    },
    "state": "Active",
    "stateDetails": ""
  },
  "remoteSliceID": 8,
  "remoteVolumeID": 8,
  "remoteVolumeName": "PairingDoc",
  "volumePairUUID": "229fcfb3-2d35-4625-865a-d04bb9455cef"
]
}
]
}
}
```

TestHardwareConfig

TestHardwareConfig 方法返回类似于以下示例的响应。

```
{
  "id": 1,
  "result": {
    "nodes": [
      {
        "nodeID": 1,
        "result": {
          "details": {
            "BIOS_REVISION": {
              "Passed": true,
              "actual": "2.0",
              "comparator": ">=",
              "预期": "1.0.0"
            },
            "BIOS_VENDOR": {
              "Passed": true,
              "actual": "SolidFire",
              "comparator": "==",
              "expected": "SolidFire"
            },
            "BIOS_VERSION": {
              "Passed": true,
              "实际": "2.0.19",
              "comparator": ">=",
              "预期": "2.0.19"
            }
          }
        }
      }
    ]
  }
}
```

```
    },
    "CPU_CORES_00": {
      "Passed": true,
      "actual": "6",
      "comparator": "==",
      "expected": "6"
    },
    },
    "CPU_CORES_01": {
      "Passed": true,
      "actual": "6",
      "comparator": "==",
      "expected": "6"
    },
    },
    "CPU_CORES_ENABLED_00": {
      "Passed": true,
      "actual": "6",
      "comparator": "==",
      "expected": "6"
    },
    },
    "CPU_CORES_ENABLED_01": {
      "Passed": true,
      "actual": "6",
      "comparator": "==",
      "expected": "6"
    },
    },
    "CPU_MODEL_00": {
      "Passed": true,
      "实际": "Intel ( R ) Xeon ( R ) CPU E5-2620V2 @2 . 10 GHz",
      "comparator": "==",
      "预期的": "Intel ( R ) Xeon ( R ) CPU E5-2620V2 @2 . 10 GHz"
    },
    },
    "CPU_MODEL_01": {
      "Passed": true,
      "实际": "Intel ( R ) Xeon ( R ) CPU E5-2620V2 @2 . 10 GHz",
      "comparator": "==",
      "预期的": "Intel ( R ) Xeon ( R ) CPU E5-2620V2 @2 . 10 GHz"
    },
    },
    "CPU_THREADS_00": {
      "Passed": true,
      "actual": "12",
      "comparator": "==",
      "expected": "12"
    },
    },
    "CPU_THREADS_01": {
      "Passed": true,
      "actual": "12",
      "comparator": "==",
      "expected": "12"
    },
    },
    "CPU_threads_enabled": {
      "Passed": true,
      "actual": "24",
      "comparator": "==",
      "expected": "24"
    },
    },
    "IDRAC_VERSION": {
```

```
    "Passed": true,
    " 实际 " : "2.41.40.40" 、
    "comparator": ">=",
    "expected": "1.06.06"
  },
  "MEMORY_GB": {
    "Passed": true,
    " 实际 " : "64" 、
    "comparator": ">=",
    " 预期 " : "64"
  },
  "MEMORY_MHZ_00": {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "MEMORY_MHZ_01": {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "MEMORY_MHZ_02": {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "MEMORY_MHZ_03": {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "Memory _MHz _04" : {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "Memory _MHz _05" : {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "Memory _MHz _06" : {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "Memory _MHz _07" : {
    "Passed": true,
    " 实际 " : "1600" 、
    "comparator": ">=",
    "expected": "1333"
  },
  "MPTSAS_BIOS_version" : {
```

```
    "Passed": true,
    " 实际 " : "07.24.01.00" 、
    " 比较器 " : " 任何 " 、
    " 预期 "7.25.0.0"
  },
  "MPTSAS_Firmware_Version" : {
    "Passed": true,
    " 实际 " : "13.00.57.00" 、
    "comparator": "=",
    " 预期的 " : "13.0.57.0"
  },
  "NETWORK_DRIVER_ETH0": {
    "Passed": true,
    "actual": "bnx2x",
    "comparator": "=",
    " 预期 " : "bnx2x "
  },
  "NETWORK_DRIVER_ETH1": {
    "Passed": true,
    "actual": "bnx2x",
    "comparator": "=",
    " 预期 " : "bnx2x "
  },
  "NETWORK_DRIVER_ETH2": {
    "Passed": true,
    "actual": "bnx2x",
    "comparator": "=",
    " 预期 " : "bnx2x "
  },
  "NETWORK_DRIVER_ETH3": {
    "Passed": true,
    "actual": "bnx2x",
    "comparator": "=",
    " 预期 " : "bnx2x "
  },
  "Network_Firmware_version_eth0" : {
    "Passed": true,
    " 实际 " : "7.10.18-SolidFire-5F3CCBC781D53" 、
    "comparator": "=",
    " 预期的 " : "7.10.18- SolidFire-5F3CCBC781D53"
  },
  "Network_Firmware_version_eth1" : {
    "Passed": true,
    " 实际 " : "7.10.18-SolidFire-5F3CCBC781D53" 、
    "comparator": "=",
    " 预期的 " : "7.10.18- SolidFire-5F3CCBC781D53"
  },
  "Network_Firmware_version_eth2" : {
    "Passed": true,
    " 实际 " : "7.10.18-SolidFire-5F3CCBC781D53" 、
    "comparator": "=",
    " 预期的 " : "7.10.18- SolidFire-5F3CCBC781D53"
  },
  "Network_Firmware_version_eth3" : {
    "Passed": true,
    " 实际 " : "7.10.18-SolidFire-5F3CCBC781D53" 、
    "comparator": "=",
    " 预期的 " : "7.10.18- SolidFire-5F3CCBC781D53"
  },
}
```

```
    "NUM_CPU": {
      "Passed": true,
      "actual": "2",
      "comparator": "==",
      "expected": "2"
    },
    "/var/log/sf-bios.info": {
      "Passed": true,
      "actual": "false",
      "comparator": "==",
      "expected": "false"
    }
  },
  "持续时间 " : "00:00:00.195067" 、
  "result": "Passed"
}
]
}
```

联系 NetApp 支持部门

如果您就 NetApp 产品需要帮助或有任何疑问或建议，请联系 NetApp 支持部门。

- 网站: mysupport.netapp.com
- 电话: 00 800 800 80 800 和 +86 400-1203-317

版权

版权所有 © 2020 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。

根据 FAR 2.101 中的定义，本产品所含数据属于商业项目，并归 NetApp, Inc. 专有。美国政府使用此类数据需遵守非独占、不可转让、不可再许可、全球性、有限不可撤消的许可证的要求，且对此类数据的使用仅能出于履行美国政府就此类数据的交付缔结的合约并为此类合约提供支持之目的。除非本文另有规定，否则未经 NetApp, Inc. 事先书面批准，不得使用、披露、再现、修正、执行或显示此类数据。美国国防部享有的美国政府许可权利仅限于 DFARS 第 252.227-7015(b) 条中规定的权利。

商标

NetApp、NetApp 标识以及“NetApp 商标”页面所列的标志是 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。

<http://www.netapp.com/cn/legal/netapptmlist.aspx>

意见反馈及更新通知

您可以向我们发送反馈意见，帮助我们提高文档质量。初次发行生产级 (GA/FCS) 文档或者对现有生产级文档进行重要更改时，您可以收到自动通知。

如果您对改进本文档有任何建议，请发送电子邮件。

ng-gpso-china-documents@netapp.com

为了方便我们将您的意见或建议转发给相关主管部门，请在主题行写明产品名称、版本和操作系统。

您可以通过以下方式与我们联系：

- 北京市朝阳区东大桥路 9 号侨福芳草地 C 座 6 层 606 室 100020
- 电话：86-10-59293000
- 传真：86-10-59293099
- 支持电话：86-10-59293008