安全 API 方法

Element Software

NetApp
April 17, 2024

目录

安全 API 方法

您可以将 Element 软件与外部安全相关服务集成，例如外部密钥管理服务器。通过这些与安全相关的方法，您可以为空闲加密配置外部密钥管理等要素安全功能。

- AddKeyServerToProviderKmip
- CreateKeyProviderKmip
- CreateKeyServerKmip
- CreatePublicPrivateKeyPair
- DeleteKeyProviderKmip
- DeleteKeyServerKmip
- DisableEncryptionAtRest
- EnableEncryptionAtRest
- GetClientCertificateSignRequest
- GetKeyProviderKmip
- GetKeyServerKmip
- ListKeyProvidersKmip
- ListKeyServersKmip
- ModifyKeyServerKmip
- RemoveKeyServerFromProviderKmip
- SignSshKeys
- TestKeyProviderKmip
- TestKeyServerKmip

了解更多信息

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

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，系统会联系服务器以验证功能。如果指定的密钥服务器已分配给指定的密钥提供程序，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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥提供程序 ID	要将密钥服务器分配到的密钥提供程序的 ID 。	整型	无	是的。
密钥服务器 ID	要分配的密钥服务器的 ID 。	整型	无	是的。

返回值

此方法没有返回值。只要不返回错误，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":
      {}
    }
}
```

自版本以来的新增功能

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检索身份验证密钥的机制和位置。创建新的 KMIP 密钥提供程序时，不会为其分配任何 KMIP 密钥服务器。要创建 KMIP 密钥服务器，请使用 `CreateKeyServerKmip` 方法。要将其分配给提供程序，请参见 `AddKeyServerToProviderKmip` 。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
KeyProviderName	要与创建的 KMIP 密钥提供程序关联的名称。此名称仅用于显示目的，不需要唯一。	string	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
kmipKeyProvider	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。	"KeyProviderKmip"

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
    }
}
}
```

自版本以来的新增功能

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，不会联系服务器；使用此方法之前，不需要存在此服务器。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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
kmipCaCertificate	外部密钥服务器根 CA 的公有密钥证书。此操作将用于验证 TLS 通信中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，请提供一个串联字符串，其中包含所有 CA 的根证书。	string	无	是的。

Name	Description	Type	默认值	Required
kmipClientCertificate	SolidFire KMIP 客户端使用的 PEM 格式 Base64 编码 PKCS#10 X.509 证书。	string	无	是的。
kmipKeyServerHostName	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配置时，才必须提供多个主机名或 IP 地址。	string array	无	是的。
kmipKeyServerName	KMIP 密钥服务器的名称。此名称仅用于显示目的，不需要唯一。	string	无	是的。
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：

Name	Description	Type
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
        }
    }
}
```

自版本以来的新增功能

11.7

CreatePublicPrivateKeyPair

您可以使用 `CreatePublicPrivateKeyPair` 方法创建公有和专用 SSL 密钥。您可以使用这些密钥生成证书签名请求。每个存储集群只能使用一个密钥对。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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
commonName	X.509 可分辨名称 * 公用名 * 字段（ CN ）。	string	无	否
国家 / 地区	X.509 可分辨名称 * 国家 / 地区 * 字段（ C ）。	string	无	否
电子邮件地址	X.509 可分辨名称 * 电子邮件地址 * 字段（邮件）。	string	无	否
位置	X.509 可分辨名称 * 位置名称 * 字段（ L ）。	string	无	否
组织	X.509 可分辨名称 * 组织名称 * 字段（ O ）。	string	无	否
organizationalUnit	X.509 可分辨名称 * 组织单位名称 * 字段（ OU ）。	string	无	否
state	X.509 可分辨名称 * 省 / 自治区 / 直辖市 * 或 * 省 / 直辖市名称 * 字段（ ST ， SP 或 S ）。	string	无	否

返回值

此方法没有返回值。如果没有错误，则会认为创建密钥成功。

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":
      {}
}
```

自版本以来的新增功能

11.7

DeleteKeyProviderKmip

您可以使用 `DDeleteKeyProviderKmip` 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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥提供程序 ID	要删除的密钥提供程序的 ID 。	整型	无	是的。

返回值

此方法没有返回值。只要没有错误，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":
      {}
}
```

自版本以来的新增功能

11.7

DeleteKeyServerKmip

您可以使用 `DDeleteKeyServerKmip` 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。您可以删除密钥服务器，除非该密钥服务器是分配给其提供程序的最后一个密钥服务器，并且该提供程序正在提供当前正在使用的密钥。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥服务器 ID	要删除的 KMIP 密钥服务器的 ID 。	整型	无	是的。

返回值

此方法没有返回值。如果没有错误，则删除操作将视为成功。

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":
      {}
}
```

自版本以来的新增功能

11.7

DisableEncryptionAtRest

您可以使用 `DisableEncryptionAtRest` 方法删除先前使用 `EnableEncryptionAtRest` 方法应用于集群的加密。此禁用方法是异步的，会在禁用加密之前返回响应。您可以使用 `GetClusterInfo` 方法轮询系统以查看此过程何时完成。

> 要查看集群上的空闲加密和 / 或空闲软件加密的当前状态，请使用 "获取集群信息方法"。您可以使用 `GetSoftwareEncryptionAtRestInfo` "获取集群用于对空闲数据进行加密的信息的方法"。

> 您不能使用此方法禁用空闲软件加密。要禁用空闲软件加密，您需要 "创建新集群：" 禁用空闲软件加密。

Parameters

此方法没有输入参数。

返回值

此方法没有返回值。

请求示例

此方法的请求类似于以下示例：

```
{
   "method": "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" : {}
}
```

自版本以来的新增功能

9.6

了解更多信息

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

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。

要查看集群上的空闲加密和 / 或空闲软件加密的当前状态，请使用 "获取集群信息方法"。您可以使用 `GetSoftwareEncryptionAtRestInfo` "获取集群用于对空闲数据进行加密的信息的方法"。

此方法不会启用空闲软件加密。只能使用完成此操作 "创建集群方法" 使用 `enableSoftwareEncryptionAtRest` 设置为 `true` 。

启用空闲加密后，集群会自动在内部管理集群中每个节点上的驱动器的加密密钥。

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活动提供程序，并且无法删除，直到使用 `DisableEncryptionAtRest` 方法禁用空闲加密为止。

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 " 不允许加密 " 。集群检测到不可加密的节点 " 。

只有当集群正在运行且运行状况良好时，才应启用或禁用加密。您可以根据需要自由选择启用或禁用加密。

此过程是异步的，在启用加密之前返回响应。您可以使用 `GetClusterInfo` 方法轮询系统以查看此过程何时完成。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥提供程序 ID	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

此方法返回与 EnableEncryptionAtRest 方法中的以下示例类似的响应。没有可报告的结果。

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

在集群上启用空闲加密时， GetClusterInfo 将返回一个结果，其中将空闲加密状态（ "encryptionAtRestState" ）描述为 "enableing" 。完全启用空闲加密后，返回的状态将更改为 "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"
      }
   }
}
```

自版本以来的新增功能

9.6

了解更多信息

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

GetClientCertificateSignRequest

您可以使用 `GetClientCertificateSignRequest` 方法生成证书签名请求，证书颁发机构可以对该请求进行签名，以便为集群生成客户端证书。要建立与外部服务交互的信任关系，需要使用签名证书。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

Name	Description	Type
客户端证书 SignRequest	PEM 格式 Base64 编码 PKCS#10 X.509 客户端证书签名请求。	string

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..."
   }
}
```

自版本以来的新增功能

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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥提供程序 ID	要返回的 KMIP 密钥提供程序对象的 ID 。	整型	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
kmipKeyProvider	包含有关请求的密钥提供程序的详细信息的对象。	"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"
    }
   }
}
```

自版本以来的新增功能

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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥服务器 ID	要返回相关信息的 KMIP 密钥服务器的 ID 。	整型	无	是的。

返回值

此方法具有以下返回值：

Name	Description	Type
kmipKeyServer	包含有关请求的密钥服务器的详细信息的对象。	"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
        }
    }
}
```

自版本以来的新增功能

11.7

GetSoftwareEncryptionAtRestInfo

您可以使用 `GetSoftwareEncryptionAtRestInfo` 方法获取集群用于加密空闲数据的软件空闲加密信息。

Parameters

此方法没有输入参数。

返回值

此方法具有以下返回值：

参数	Description	Type	可选
MasterKeyInfo	有关当前软件空闲加密主密钥的信息。	EncryptionKeyInfo	true
rekeyMasterKeyAsyncResulteID	当前或最近重新设置密钥操作（如果有）的异步结果 ID （如果尚未删除）。`GetAsyncResult` 输出将包含一个 `newkey` 字段，其中包含有关新主密钥的信息，另一个 `keyToDecommission` 字段则包含有关旧密钥的信息。	整型	true
state	当前软件空闲加密状态。可能值为 `d`已标记 或 已启用 。	string	false
version	每次启用空闲软件加密时递增的版本号。	整型	false

请求示例

此方法的请求类似于以下示例：

```
{
 "method": "getsoftwareencryptionatrestinfo"
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
    "id": 1,
    "result": {
        "masterKeyInfo": {
            "keyCreatedTime": "2021-09-20T23:15:56Z",
            "keyID": "4d80a629-a11b-40ab-8b30-d66dd5647cfd",
            "keyManagementType": "internal"
        },
        "state": "enabled",
        "version": 1
    }
}
```

自版本以来的新增功能

12.3

了解更多信息

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

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来筛选此列表。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥提供程序 IsActive	根据返回的 KMIP 密钥服务器对象是否处于活动状态对其进行筛选。可能值： • 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）。 如果省略此参数，则不会根据返回的 KMIP 密钥提供程序是否处于活动状态对其进行筛选。	boolean	无	否
kmipKeyProviderHasServerAssigned	根据是否已分配 KMIP 密钥服务器筛选返回的 KMIP 密钥提供程序。可能值： • 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。 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。	boolean	无	否

返回值

此方法具有以下返回值：

Name	Description	Type

kmipKeyProviders	已创建的 KMIP 密钥提供程序的列表。	"KeyProviderKmip" 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"
        }
    ]
}
}
```

自版本以来的新增功能

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

来筛选结果。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥提供程序 ID	指定后，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，则不会根据是否已将返回的 KMIP 密钥服务器分配给指定的 KMIP 密钥提供程序对其进行筛选。	整型	无	否
kmipAssignedProviderIsActive	根据返回的 KMIP 密钥服务器对象是否处于活动状态对其进行筛选。可能值： • true ： 仅返回处于活动状态的 KMIP 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 KMIP 密钥服务器（未提供任何密钥且可删除）。 如果省略此参数，则不会根据返回的 KMIP 密钥服务器是否处于活动状态对其进行筛选。	boolean	无	否

Name	Description	Type	默认值	Required
kmipHasProviderAssigned	根据是否已分配 KMIP 密钥提供程序对返回的 KMIP 密钥服务器进行筛选。可能值： • true ： 仅返回已分配 KMIP 密钥提供程序的 KMIP 密钥服务器。 • false ： 仅返回未分配 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 KMIP 密钥提供程序对返回的 KMIP 密钥服务器进行筛选。	boolean	无	否

返回值

此方法具有以下返回值：

Name	Description	Type
kmipKeyServer	已创建的 KMIP 密钥服务器的完整列表。	"KeyServerKmip" 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
        }
    ]
}
```

自版本以来的新增功能

11.7

ModifyKeyServerKmip

您可以使用 `MmodifyKeyServerKmip` 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。尽管唯一必需的参数是 keyServerID，但仅包含 keyServerID 的请求不会执行任何操作，也不会返回任何错误。您指定的任何其他参数将使用指定的 keyServerID 替换密钥服务器的现有值。在此操作期间，系统会联系密钥服务器以确保其正常运行。您可以使用 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配置。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥服务器 ID	要修改的 KMIP 密钥服务器的 ID 。	整型	无	是的。

kmipCaCertificate	外部密钥服务器根 CA 的公有密钥证书。此操作将用于验证 TLS 通信中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，请提供一个串联字符串，其中包含所有 CA 的根证书。	string	无	否
kmipClientCertificate	SolidFire KMIP 客户端使用的 PEM 格式 Base64 编码 PKCS#10 X.509 证书。	string	无	否
kmipKeyServerHostName	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配置时，才必须提供多个主机名或 IP 地址。	string array	无	否
kmipKeyServerName	KMIP 密钥服务器的名称。此名称仅用于显示目的，不需要唯一。	string	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：

Name	Description	Type
kmipKeyServer	包含有关新修改的密钥服务器的详细信息的对象。	"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
        }
    }
}
```

自版本以来的新增功能

11.7

RekeySoftwareEncryptionAtRestMasterKey

您可以使用 `RekeySoftwareEncryptionAtRestMasterKey` 方法重新设置用于加密磁盘（数据加密密钥）的软件空闲加密主密钥的密钥。在创建集群期间，系统会将空闲软件加密配置为使用内部密钥管理（ InInternal Key Management， KKM ）。创建集群后，可以使用此重新设置密钥方法来使用 IKM 或外部密钥管理（ External Key Management， EKM ）。

Parameters

此方法具有以下输入参数。如果未指定 `keyManagementType` 参数，则使用现有密钥管理配置执行重新设置密钥操作。如果指定了 `keyManagementType` ，并且密钥提供程序是外部的，则还必须使用 `keyProviderID` 参数。

参数	Description	Type	可选
KeyManagementType	用于管理主密钥的密钥管理类型。可能值包括：`Internal`：使用内部密钥管理重新设置密钥。外部：使用外部密钥管理重新设置密钥。如果未指定此参数，则使用现有密钥管理配置执行重新设置密钥操作。	string	true
密钥提供程序 ID	要使用的密钥提供程序的 ID 。这是作为 `CreateKeyProvider` 方法之一返回的唯一值。只有当 `keyManagementType` 为 `External` 且在其他情况下无效时，才需要此 ID 。	整型	true

返回值

此方法具有以下返回值：

参数	Description	Type	可选
异步处理	使用 `asyncHandle` 值 with `GetAsyncResult` 确定重新设置密钥操作的状态。`GetAsyncResult` 输出将包含一个 `newkey` 字段，其中包含有关新主密钥的信息，另一个 `keyToDecommission` 字段则包含有关旧密钥的信息。	整型	false

请求示例

此方法的请求类似于以下示例：

```
{
 "method": "rekeysoftwareencryptionatrestmasterkey",
 "params": {
   "keyManagementType": "external",
   "keyProviderID": "<ID number>"
 }
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
    "asyncHandle": 1
}
```

自版本以来的新增功能

12.3

了解更多信息

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

RemoveKeyServerFromProviderKmip

您可以使用 `RemoveKeyServerFromProviderKmip` 方法从分配给它的提供程序中取消分配指定的密钥管理互操作性协议（ Key Management Interoperability Protocol ， KMIP ）密钥服务器。除非密钥服务器是最后一个密钥服务器且其提供程序处于活动状态（提供当前正在使用的密钥），否则您可以从其提供程序中取消分配密钥服务器。如果指定的密钥服务器未分配给提供程序，则不会执行任何操作，也不会返回任何错误。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥服务器 ID	要取消分配的 KMIP 密钥服务器的 ID 。	整型	无	是的。

返回值

此方法没有返回值。只要不返回任何错误，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": "RemoveKeyServerFromProviderKmip",
  "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":
      {}
    }
}
```

自版本以来的新增功能

11.7

SignSshKeys

使用在集群上启用SSH后 "EnableSSH方法"、您可以使用`SignSshKeys`方法访问节点上的Shell。

从Element 12.5开始、`sfreadonly`是一个新的系统帐户、可用于对节点进行基本故障排除。此`API`可在集群中的所有节点上使用`sfreadonly` system帐户启用SSH访问。

除非NetApp支持部门建议、否则不支持对系统进行任何更改、从而使您的支持合同失效、并可能导致数据不稳定或无法访问。

使用方法后、您必须从响应中复制密钥链、将其保存到要启动SSH连接的系统、然后运行以下命令：

```
ssh -i <identity_file> sfreadonly@<node_ip>
```

`identity_file`是一个文件、可从中读取公有 密钥身份验证的身份(私钥)、而`node_IP`是节点的IP地址。有关`identity_file`的详细信息、请参见SSH手册页。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
Duration	1到24之间的整数、反映签名密钥有效的小时数。如果未指定持续时间、则使用默认值。	整型	1.	否
公共密钥	如果提供、则此参数将仅返回signed_public_key、而不是为用户创建完整的密钥链。 在使用`+`的浏览器中使用URL栏提交的公有密钥将被解释为间隔签名和中断签名。	string	空	否

Name	Description	Type	默认值	Required
sfadmin	允许在您使用supportAdmin集群访问权限进行API调用时或节点不在集群中时访问sfadmin shell帐户。	boolean	false	否

返回值

此方法具有以下返回值：

Name	Description	Type
keygen_status	包含签名密钥中的标识、允许的主体以及该密钥的有效开始和结束日期。	string
private_key	只有当API为最终用户生成完整的密钥链时、才会返回专用SSH密钥值。 此值采用Base64编码；您必须在将此值写入文件时对其进行解码、以确保将其读取为有效的私钥。	string
公共密钥	只有当公有 为最终用户生成完整的密钥链时、才会返回SSH密钥值。 将public_key参数传递到API方法时、响应中仅返回`sUG_public_key`值。	string
signed_public_key	对公有 密钥签名后产生的SSH公有密钥、无论此密钥是用户提供的、还是由API生成的。	string

请求示例

此方法的请求类似于以下示例：

```
{
    "method": "SignSshKeys",
    "params": {
        "duration": 2,
        "publicKey":<string>
    },
    "id": 1
}
```

响应示例

此方法返回类似于以下示例的响应：

```
{
  "id": null,
  "result": {
    "signedKeys": {
      "keygen_status": <keygen_status>,
      "signed_public_key": <signed_public_key>
    }
  }
}
```

在此示例中、系统将对公有 密钥进行签名并返回有效期(1-24小时)。

自版本以来的新增功能

12.5.

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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥提供程序 ID	要测试的密钥提供程序的 ID 。	整型	无	是的。

返回值

此方法没有返回值。只要未返回错误，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":
      {}
    }
}
```

自版本以来的新增功能

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。

Parameters

此方法具有以下输入参数：

Name	Description	Type	默认值	Required
密钥服务器 ID	要测试的 KMIP 密钥服务器的 ID 。	整型	无	是的。

返回值

此方法没有返回值。如果未返回任何错误，则此测试将视为成功。

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":
      {}
    }
```

自版本以来的新增功能

11.7