Métodos de API de seguridad

Element Software

NetApp
October 01, 2024

Tabla de contenidos

Métodos de API de seguridad

Es posible integrar el software Element con servicios relacionados con la seguridad externos, como un servidor de gestión de claves externo. Estos métodos relacionados con la seguridad permiten configurar funciones de seguridad de Element, como la gestión de claves externa para el cifrado en reposo.

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

Obtenga más información

- "Documentación de SolidFire y el software Element"
- "Documentación para versiones anteriores de SolidFire de NetApp y los productos Element"

AddKeyServerToProviderKmip

Puede usar `AddKeyServerToProviderKmip` el método para asignar un servidor de claves de protocolo de interoperabilidad de gestión de claves (KMIP) al proveedor de claves especificado. Durante la asignación, se contacta con el servidor para verificar la funcionalidad. Si el servidor de claves especificado ya está asignado al proveedor de claves especificado, no se realiza ninguna acción y no se devuelve ningún error. Puede eliminar la asignación mediante el `RemoveKeyServerFromProviderKmip` método.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderID	El ID del proveedor de claves al que se asignará el servidor de claves.	entero	Ninguno	Sí
KeyServerID	El ID del servidor de claves que se asignará.	entero	Ninguno	Sí

Valores devueltos

Este método no tiene ningún valor devuelto. La asignación se considera correcta siempre que no se devuelva ningún error.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "AddKeyServerToProviderKmip",
  "params": {
    "keyProviderID": 1,
    "keyServerID": 15
    },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

11,7

CreateKeyProviderKmip

Puede usar este `CreateKeyProviderKmip` método para crear un proveedor de claves de protocolo de interoperabilidad de gestión de claves (KMIP) con el nombre especificado. Un proveedor de claves define un mecanismo y una ubicación para recuperar claves de autenticación. Cuando se crea un proveedor de claves KMIP nuevo, no tiene ningún servidor de claves KMIP asignado. Para crear un servidor de claves KMIP, utilice `CreateKeyServerKmip` el método. Para asignarla a un proveedor, consulte `AddKeyServerToProviderKmip`.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderName	El nombre que se asignará al proveedor de claves KMIP creado. Este nombre sólo se utiliza con fines de visualización y no necesita ser único.	cadena	Ninguno	Sí

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
KmipKeyProvider	Objeto que contiene detalles acerca del proveedor de claves recién creado.	"KeyProviderKmip"

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "CreateKeyProviderKmip",
  "params": {
    "keyProviderName": "ProviderName",
    },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
   "id": 1,
   "result":
      {
    "kmipKeyProvider": {
        "keyProviderName": "ProviderName",
        "keyProviderIsActive": true,
        "kmipCapabilities": "SSL",
        "keyServerIDs": [
            15
        ],
        "keyProviderID": 1
    }
}
}
```

Nuevo desde la versión

11,7

CreateKeyServerKmip

Puede usar este `CreateKeyServerKmip` método para crear un servidor de claves de protocolo de interoperabilidad de gestión de claves (KMIP) con los atributos especificados. Durante la creación, no se contacta con el servidor; no es necesario que exista antes de utilizar este método. Para configuraciones de servidor de claves en clúster, debe proporcionar los nombres de host o direcciones IP de todos los nodos de servidor en el parámetro kmipKeyServerHostnames. Puede usar el `TestKeyServerKmip` método para probar un servidor de claves.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KmipCaCertificate	El certificado de clave pública de la CA raíz del servidor de claves externo. Esto se utilizará para verificar el certificado presentado por el servidor de claves externo en la comunicación TLS. Para los clústeres de servidores de claves en los que los servidores individuales utilizan distintas CA, proporcione una cadena concatenada que contenga los certificados raíz de todas las CA.	cadena	Ninguno	Sí
KmipClientCertificate	Un certificado PKCS#10 X.509 codificado en Base64 con formato PEM que utiliza el cliente KMIP de SolidFire.	cadena	Ninguno	Sí
KmipKeyServerHostnames	Cabina de los nombres de host o las direcciones IP asociadas con este servidor de claves KMIP. Sólo se deben proporcionar varios nombres de host o direcciones IP si los servidores de claves se encuentran en una configuración en clúster.	matriz de cadenas	Ninguno	Sí

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KmipKeyServerName	El nombre del servidor de claves KMIP. Este nombre sólo se utiliza con fines de visualización y no necesita ser único.	cadena	Ninguno	Sí
KmipKeyServerPort	El número de puerto asociado con este servidor de claves KMIP (por lo general, 5696).	entero	Ninguno	No

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
KmipKeyServer	Objeto que contiene detalles acerca del servidor de claves recién creado.	"KeyServerKmip"

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "CreateKeyServerKmip",
  "params": {
    "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
    "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
    "kmipKeyServerHostnames" : ["server1.hostname.com",
"server2.hostname.com"],
    "kmipKeyServerName" : "keyserverName",
    "kmipKeyServerPort" : 5696
  },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
   "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
        }
     }
}
```

Nuevo desde la versión

11,7

CreatePublicPrivateKeyPair

Puede utilizar el `CreatePublicPrivateKeyPair` método para crear claves SSL públicas y privadas. Es posible usar estas claves para generar solicitudes de firma de certificados. Solo puede haber una pareja de claves en uso para cada clúster de almacenamiento. Antes de utilizar este método para reemplazar las claves existentes, asegúrese de que ningún proveedor ya utilice las claves.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
CommonName	El campo X.509 Nombre distintivo **Nombre común** (CN).	cadena	Ninguno	No
país	El campo X 509 de nombre completo **país** ©.	cadena	Ninguno	No

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
Dirección de correo electrónico	El campo X 509 Nombre distintivo **Dirección de correo electrónico** (CORREO).	cadena	Ninguno	No
localidad	El campo X 509 Nombre distintivo **Nombre de localidad** (L).	cadena	Ninguno	No
organización	El campo X 509 Nombre distintivo **Nombre de organización** (o).	cadena	Ninguno	No
Unidad organizativa	El campo X.509 Nombre distintivo **Nombre de unidad organizativa** (OU).	cadena	Ninguno	No
estado	El campo X 509 Nombre distinguido **Estado** o **Nombre de provincia** (ST o SP o S).	cadena	Ninguno	No

Valores devueltos

Este método no tiene valores devueltos. Si no hay ningún error, la creación de claves se considera correcta.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "CreatePublicPrivateKeyPair",
  "params": {
    "commonName": "Name",
    "country": "US",
    "emailAddress" : "email@domain.com"
   },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

11,7

DeleteKeyProviderKmip

Es posible usar `DeleteKeyProviderKmip` el método para eliminar el proveedor de claves especificado del protocolo de interoperabilidad de gestión de claves inactivo (KMIP).

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderID	El ID del proveedor de claves que se eliminará.	entero	Ninguno	Sí

Valores devueltos

Este método no tiene valores devueltos. La operación de eliminación se considera correcta siempre que no haya error.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "DeleteKeyProviderKmip",
  "params": {
    "keyProviderID": "1"
    },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

11,7

DeleteKeyServerKmip

Es posible usar este `DeleteKeyServerKmip` método para eliminar un servidor de claves existente del protocolo de interoperabilidad de gestión de claves (KMIP). Puede eliminar un servidor de claves a menos que sea el último asignado a su proveedor, y ese proveedor proporciona claves que están en uso actualmente.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyServerID	El ID del servidor de claves KMIP que se desea eliminar.	entero	Ninguno	Sí

Valores devueltos

Este método tiene los valores no return. La operación de eliminación se considera correcta si no hay errores.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "DeleteKeyServerKmip",
  "params": {
     "keyServerID": 15
  },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

11,7

DisableEncryptionAtest

Puede usar el `DisableEncryptionAtRest` método para quitar el cifrado que se había aplicado previamente al clúster mediante `EnableEncryptionAtRest` el método. Este método disable es asíncrono y devuelve una respuesta antes de que se deshabilite el cifrado. Puede utilizar el `GetClusterInfo` método para sondear el sistema para ver cuándo se ha completado el proceso.

Para ver el estado actual del cifrado en reposo o el cifrado de software en reposo en el clúster, utilice el "obtenga el método de información del clúster". Puede utilizar el `GetSoftwareEncryptionAtRestInfo` "método para obtener información que utiliza el clúster para cifrar datos en reposo".

No se puede usar este método para deshabilitar el cifrado de software en reposo. Para deshabilitar el cifrado de software en reposo, debe "cree un nuevo clúster"desactivar el cifrado de software en reposo.

Parámetros

Este método no tiene parámetros de entrada.

Valores devueltos

Este método no tiene valores devueltos.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

9,6

Obtenga más información

- "GetClusterInfo"
- "Documentación de SolidFire y el software Element"
- "Documentación para versiones anteriores de SolidFire de NetApp y los productos Element"

EnableEncryptionAtest

Puede usar el `EnableEncryptionAtRest` método para habilitar el cifrado estándar de cifrado avanzado (AES) de 256 bits en reposo en el clúster de modo que el clúster pueda gestionar la clave de cifrado utilizada para las unidades en cada nodo. Esta función no está habilitada de forma predeterminada.

Para ver el estado actual del cifrado en reposo o el cifrado de software en reposo en el clúster, utilice el "obtenga el método de información del clúster". Puede utilizar el `GetSoftwareEncryptionAtRestInfo` "método para obtener información que utiliza el clúster para cifrar datos en reposo".

Este método no habilita el cifrado de software en reposo. Esto sólo se puede hacer utilizando el "cree el método de clúster" con `enableSoftwareEncryptionAtRest` establecido en `true`.

Cuando habilita el cifrado en reposo, el clúster gestiona automáticamente las claves de cifrado internamente para las unidades de cada nodo del clúster.

Si se especifica un keyProviderID, la contraseña se genera y recupera según el tipo de proveedor de claves. Esto suele realizarse mediante un servidor de claves de protocolo de interoperabilidad de gestión de claves (KMIP) en el caso de un proveedor de claves KMIP. Después de esta operación, el proveedor especificado se considera activo y no se puede eliminar hasta que se desactive Cifrado en reposo mediante el `DisableEncryptionAtRest` método.

Si tiene un tipo de nodo con un número de modelo que termina en “-NE”, la `EnableEncryptionAtRest` llamada al método fallará con una respuesta de “Cifrado no permitido. Cluster detectado nodo no encriptable".

Solo tendrá que habilitar o deshabilitar el cifrado cuando el clúster se ejecute y esté en buen estado. Puede activar o desactivar el cifrado a su discreción y con la frecuencia que necesite.

Este proceso es asíncrono y devuelve una respuesta antes de activar el cifrado. Puede utilizar el `GetClusterInfo` método para sondear el sistema para ver cuándo se ha completado el proceso.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderID	El ID de un proveedor de claves KMIP que se debe usar.	entero	Ninguno	No

Valores devueltos

Este método no tiene valores devueltos.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplos de respuestas

Este método devuelve una respuesta similar al siguiente ejemplo del método EnableEncryptionAtRest. No hay resultados para informar.

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

Mientras que el cifrado en reposo se está habilitado en un clúster, GetClusterInfo muestra un resultado que describe el estado del cifrado en reposo ("cifrado AtRestState") como "habilitado". Una vez que el cifrado en reposo está completamente habilitado, el estado devuelto cambia a "habilitado".

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

Nuevo desde la versión

9,6

Obtenga más información

- "SecureEraseDrives"
- "GetClusterInfo"
- "Documentación de SolidFire y el software Element"
- "Documentación para versiones anteriores de SolidFire de NetApp y los productos Element"

GetClientCertificateSignRequest

Puede utilizar `GetClientCertificateSignRequest` el método para generar una solicitud de firma de certificación que puede estar firmada por una entidad de certificación a fin de generar un certificado de cliente para el clúster. Los certificados firmados son necesarios para establecer una relación de confianza para interactuar con servicios externos.

Parámetros

Este método no tiene parámetros de entrada.

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
ClientCertificadosSignRequest	Una solicitud de firma de certificado de cliente PKCS#10 X.509 codificada con PEM Base64.	cadena

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
   "id": 1,
   "result":
      {
        "clientCertificateSignRequest":
"MIIByjCCATMCAQAwgYkxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybm..."
   }
}
```

Nuevo desde la versión

11,7

GetKeyProviderKmip

Es posible usar `GetKeyProviderKmip` el método para recuperar información sobre el proveedor de claves del protocolo de interoperabilidad de gestión de claves (KMIP) especificado.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderID	El ID del objeto de proveedor de claves KMIP que se va a devolver.	entero	Ninguno	Sí

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
KmipKeyProvider	Objeto que contiene detalles sobre el proveedor de claves solicitado.	"KeyProviderKmip"

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "GetKeyProviderKmip",
  "params": {
    "keyProviderID": 15
    },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
   "id": 1,
   "result":
    {
    "kmipKeyProvider": {
        "keyProviderID": 15,
        "kmipCapabilities": "SSL",
        "keyProviderIsActive": true,
        "keyServerIDs": [
            1
        ],
        "keyProviderName": "ProviderName"
    }
   }
}
```

Nuevo desde la versión

11,7

GetKeyServerKmip

El `GetKeyServerKmip` método permite obtener información sobre el servidor de claves del protocolo de interoperabilidad de gestión de claves (KMIP) especificado.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyServerID	El ID del servidor de claves KMIP acerca de la cual se desea obtener información.	entero	Ninguno	Sí

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
KmipKeyServer	Objeto que contiene detalles acerca del servidor de claves solicitado.	"KeyServerKmip"

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "GetKeyServerKmip",
  "params": {
     "keyServerID": 15
  },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
   "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
        }
     }
}
```


Nuevo desde la versión

11,7

GetSoftwareEncryptionAtRestInfo

Puede utilizar el `GetSoftwareEncryptionAtRestInfo` método para obtener información sobre cifrado por software en reposo que el clúster utiliza para cifrar los datos en reposo.

Parámetros

Este método no tiene parámetros de entrada.

Valores devueltos

Este método tiene los siguientes valores devueltos:

Parámetro	Descripción	Tipo	Opcional
MasterKeyInfo	Información acerca de la clave maestra actual de cifrado en reposo de software.	Cifrar KeyInfo	Verdadero

Parámetro	Descripción	Tipo	Opcional
RekeyMasterKeyAsyncResultID	ID de resultado asíncrono de la operación de nueva clave actual o más reciente (si la hay), si aún no se ha suprimido. `GetAsyncResult` la salida incluirá un `newKey` campo que contiene información sobre la nueva clave maestra y un `keyToDecommission` campo que contiene información sobre la clave antigua.	entero	Verdadero
estado	El estado actual del cifrado de software en reposo. Los valores posibles son `disabled` o. `enabled`	cadena	Falso
versión	Un número de versión que aumenta cada vez que se habilita el cifrado de software en reposo.	entero	Falso

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

12,3

Obtenga más información

- "Documentación de SolidFire y el software Element"
- "Documentación para versiones anteriores de SolidFire de NetApp y los productos Element"

ListKeyProvidersKmip

Puede usar el `ListKeyProvidersKmip` método para recuperar una lista de todos los proveedores de claves del protocolo de interoperabilidad de gestión de claves (KMIP) existentes. Puede filtrar la lista especificando parámetros adicionales.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderIsActive	Los filtros regresaron objetos del servidor de claves KMIP en función de si están activos. Los posibles valores son los siguientes: • True: Devuelve solo los proveedores de claves KMIP activos (que proporcionan claves en uso actualmente). • False: Devuelve solo los proveedores de claves KMIP inactivos (sin ofrecer ninguna clave y sin poder eliminarla). Si se omite, los proveedores de claves KMIP que se devuelven no se filtran en función de si están activos.	booleano	Ninguno	No

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KmipKeyProviderHasServerAssigned	Los filtros devuelven proveedores de claves KMIP en función de si tienen asignado un servidor de claves KMIP. Los posibles valores son los siguientes: • True: Solo devuelve los proveedores de claves KMIP que tienen asignado un servidor de claves KMIP. • False: Devuelve solo los proveedores de claves KMIP que no tienen asignado un servidor de claves KMIP. Si se omite, los proveedores de claves KMIP que se devuelven no se filtran en función de si tienen asignado un servidor de claves KMIP.	booleano	Ninguno	No

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
KmipKeyProviders	Una lista de los proveedores de claves KMIP que se hayan creado.	"KeyProviderKmip" cabina

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
   "id": 1,
   "result":
    {
    "kmipKeyProviders": [
        {
            "keyProviderID": 15,
            "kmipCapabilities": "SSL",
            "keyProviderIsActive": true,
            "keyServerIDs": [
                1
            ],
            "keyProviderName": "KeyProvider1"
        }
    ]
}
}
```

Nuevo desde la versión

11,7

ListKeyServersKmip

Puede usar el `ListKeyServersKmip` método para incluir todos los servidores de claves del protocolo de interoperabilidad de gestión de claves (KMIP) que se crearon. Los resultados se pueden filtrar especificando parámetros adicionales.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderID	Cuando se especifica, el método solo devuelve los servidores de claves KMIP asignados al proveedor de claves KMIP especificado. Si se omite, los servidores de claves KMIP devueltos no se filtrarán en función de si se asignan al proveedor de claves KMIP especificado.	entero	Ninguno	No
KmipAssignedProviderIsActive	Los filtros regresaron objetos del servidor de claves KMIP en función de si están activos. Los posibles valores son los siguientes: • True: Devuelve solo los servidores de claves KMIP activos (que proporcionan claves en uso actualmente). • False: Devuelve solo los servidores de claves KMIP inactivos (sin proporcionar ninguna clave y sin poder eliminarse). Si se omite, los servidores de claves KMIP devueltos no se filtran en función de si están activos.	booleano	Ninguno	No

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KmipHasProviderAssigned	Los filtros devuelven servidores de claves KMIP en función de si tienen asignado un proveedor de claves KMIP. Los posibles valores son los siguientes: • True: Solo devuelve los servidores de claves KMIP que tienen asignado un proveedor de claves KMIP. • False: Devuelve solo los servidores de claves KMIP que no tienen asignado un proveedor de claves KMIP. Si se omite, los servidores de claves KMIP que se devuelven no se filtran en función de si tienen asignado un proveedor de claves KMIP.	booleano	Ninguno	No

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
KmipKeyServers	La lista completa de los servidores de claves KMIP que se crearon.	"KeyServerKmip" cabina

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
    "kmipKeyServers": [
        {
            "kmipKeyServerName": "keyserverName",
            "kmipClientCertificate": "dKkkirWmnWXbj9T/UWZYB2oK0z5...",
            "keyServerID": 15,
            "kmipAssignedProviderIsActive": true,
            "kmipKeyServerPort": 5696,
            "kmipCaCertificate": "MIICPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
            "kmipKeyServerHostnames": [
                "server1.hostname.com", "server2.hostname.com"
            ],
            "keyProviderID": 1
        }
    ]
}
```

Nuevo desde la versión

11,7

ModifyKeyServerKmip

Puede usar este `ModifyKeyServerKmip` método para modificar un servidor de claves existente de protocolo de interoperabilidad de gestión de claves (KMIP) a los atributos especificados. Aunque el único parámetro requerido es keyServerID, una solicitud que contiene sólo el keyServerID no realizará ninguna acción y no devolverá ningún error. Cualquier otro parámetro que especifique reemplazará los valores existentes para el servidor de claves con el keyServerID especificado. Se contacta con el servidor de claves durante la operación para garantizar que funciona. Puede proporcionar varios nombres de host o direcciones IP con el parámetro kmipKeyServerHostnames, pero sólo si los servidores de claves están en una configuración en clúster.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyServerID	El ID del servidor de claves KMIP que se desea modificar.	entero	Ninguno	Sí
KmipCaCertificate	El certificado de clave pública de la CA raíz del servidor de claves externo. Esto se utilizará para verificar el certificado presentado por el servidor de claves externo en la comunicación TLS. Para los clústeres de servidores de claves en los que los servidores individuales utilizan distintas CA, proporcione una cadena concatenada que contenga los certificados raíz de todas las CA.	cadena	Ninguno	No
KmipClientCertificate	Un certificado PKCS#10 X.509 codificado en Base64 con formato PEM que utiliza el cliente KMIP de SolidFire.	cadena	Ninguno	No

KmipKeyServerHostnames	Cabina de los nombres de host o las direcciones IP asociadas con este servidor de claves KMIP. Sólo se deben proporcionar varios nombres de host o direcciones IP si los servidores de claves se encuentran en una configuración en clúster.	matriz de cadenas	Ninguno	No
KmipKeyServerName	El nombre del servidor de claves KMIP. Este nombre sólo se utiliza con fines de visualización y no necesita ser único.	cadena	Ninguno	No
KmipKeyServerPort	El número de puerto asociado con este servidor de claves KMIP (por lo general, 5696).	entero	Ninguno	No

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
KmipKeyServer	Un objeto que contiene detalles acerca del servidor de claves recién modificado.	"KeyServerKmip"

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "ModifyKeyServerKmip",
  "params": {
    "keyServerID": 15
    "kmipCaCertificate": "CPDCCAaUCEDyRMcsf9tAbDpq40ES/E...",
    "kmipClientCertificate": "kirWmnWXbj9T/UWZYB2oK0z5...",
    "kmipKeyServerHostnames" : ["server1.hostname.com",
"server2.hostname.com"],
    "kmipKeyServerName" : "keyserverName",
    "kmipKeyServerPort" : 5696
  },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

```
{
   "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
        }
    }
}
```

Nuevo desde la versión

11,7

RekeySoftwareEncryptionAtRestMasterKey

Puede utilizar el `RekeySoftwareEncryptionAtRestMasterKey` método para volver

a introducir la clave maestra de cifrado de software en reposo utilizada para cifrar deks (claves de cifrado de datos). Durante la creación de clústeres, el cifrado de software en reposo se configura para utilizar Internal Key Management (IKM). Este método de nueva clave se puede utilizar después de la creación de un clúster para utilizar IKM o Gestión de claves externas (EKM).

Parámetros

Este método tiene los siguientes parámetros de entrada. Si no se especifica el `keyManagementType` parámetro, la operación de regeneración de claves se realiza mediante la configuración de gestión de claves existente. Si se especifica el `keyManagementType` y el proveedor de claves es externo, `keyProviderID` también se debe utilizar el parámetro.

Parámetro	Descripción	Tipo	Opcional
Tipo de material de la columna	El tipo de gestión de claves utilizado para gestionar la clave maestra. Los valores posibles son `Internal`: Volver a introducir claves mediante la gestión de claves interna. `External`: Rekey usando la gestión de claves externa. Si no se especifica este parámetro, se ejecuta la operación de nueva clave mediante la configuración de gestión de claves existente.	cadena	Verdadero
KeyProviderID	El ID del proveedor de claves que se utilizará. Este es un valor único devuelto como parte de uno de los `CreateKeyProvider` métodos. El ID solo es necesario cuando `keyManagementType` es `External` y no es válido.	entero	Verdadero

Valores devueltos

Este método tiene los siguientes valores devueltos:

Parámetro	Descripción	Tipo	Opcional
Establish asyncHandle	Determine el estado de la operación de nueva clave mediante este `asyncHandle` valor con `GetAsyncResult`. `GetAsyncResult` la salida incluirá un `newKey` campo que contiene información sobre la nueva clave maestra y un `keyToDecommission` campo que contiene información sobre la clave antigua.	entero	Falso

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

12,3

Obtenga más información

- "Documentación de SolidFire y el software Element"
- "Documentación para versiones anteriores de SolidFire de NetApp y los productos Element"

RemoveKeyServerFromProviderKmip

Puede usar el `RemoveKeyServerFromProviderKmip` método para anular la asignación de claves del servidor de protocolo de interoperabilidad de gestión de claves (KMIP) especificado del proveedor al que se le asignó. Puede anular la asignación de un servidor de claves de su proveedor a menos que sea el último y su proveedor esté activo (proporcionando claves que estén en uso actualmente). Si el servidor de claves especificado no está asignado a un proveedor, no se realiza ninguna acción y no se devuelve ningún error.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyServerID	El ID del servidor de claves KMIP para anular la asignación.	entero	Ninguno	Sí

Valores devueltos

Este método no tiene valores devueltos. La eliminación se considera correcta siempre que no se devuelva ningún error.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "RemoveKeyServerFromProviderKmip",
  "params": {
    "keyServerID": 1
   },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

11,7

SignSshKeys

Después de habilitar SSH en el clúster mediante el "Método EnableSSH", puede usar el `SignSshKeys` método para obtener acceso a un shell en un nodo.

A partir de Element 12,5, `sfreadonly` se ofrece una nueva cuenta del sistema que permite solucionar los problemas básicos en un nodo. Esta API permite el acceso de SSH mediante `sfreadonly` la cuenta del sistema en todos los nodos del clúster.

A menos que el soporte de NetApp lo indique, cualquier modificación del sistema no será compatible, anulando su contrato de soporte y podría dar lugar a inestabilidad o inaccesibilidad a los datos.

Después de utilizar el método, debe copiar la cadena de claves de la respuesta, guardarla en el sistema que iniciará la conexión SSH y, a continuación, ejecutar el siguiente comando:

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

```
`identity_file` Es un archivo desde el que se lee la identidad (clave
privada) para la autenticación de clave pública y `node_ip` es la
dirección IP del nodo. Para obtener más información sobre `identity_file`,
consulte la página del comando man SSH.
```

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
duración	El número entero de 1 a 24 refleja el número de horas de la clave firmada para que sea válida. Si no se especifica la duración, se utiliza el valor predeterminado.	entero	1	No
Publickey	Si se proporciona, este parámetro sólo devolverá la clave_pública_firmada en lugar de crear una cadena de claves completa al usuario.  Las claves públicas enviadas mediante la barra de URL en un navegador con + se interpretan como espaciadas y como firma de rotura.	cadena	Nulo	No

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
sfadmin	Permite acceder a la cuenta de shell sfadmin cuando realiza la llamada API con acceso a clústeres supportAdmin, o cuando el nodo no está en un clúster.	booleano	Falso	No

Valores devueltos

Este método tiene los siguientes valores devueltos:

Nombre	Descripción	Tipo
estado_keygen	Contiene el código de la clave firmada, los principales permitidos y las fechas de inicio y finalización válidas de la clave.	cadena
clave_privada	Un valor de clave SSH privada solo se devuelve si la API genera una cadena de claves completa para el usuario final. El valor está codificado en Base64; debe descodificar el valor cuando se escribe en un archivo para asegurarse de que se lee como clave privada válida.	cadena

Nombre	Descripción	Tipo
public_key	Un valor de clave SSH pública solo se devuelve si la API genera una cadena de claves completa para el usuario final. Cuando pasa un parámetro PUBLIC_KEY al método API, solo `signed_public_key` se devuelve el valor en la respuesta.	cadena
clave_pública_firmada	La clave pública SSH que resulta de la firma de la clave pública, ya sea proporcionada por el usuario o generada por la API.	cadena

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

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

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

En este ejemplo, se firma una clave pública y se devuelve que es válida durante el tiempo (1-24 horas).

Nuevo desde la versión

12,5

TestKeyProviderKmip

Puede usar el `TestKeyProviderKmip` método para probar si el proveedor de claves del protocolo de interoperabilidad de gestión de claves (KMIP) especificado es accesible y funciona con normalidad.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyProviderID	El ID del proveedor de claves que se probará.	entero	Ninguno	Sí

Valores devueltos

Este método no tiene valores devueltos. La prueba se considera correcta mientras no se devuelve ningún error.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "TestKeyProviderKmip",
  "params": {
    "keyProviderID": 15
  },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

11,7

TestKeyServerKmip

Puede usar este `TestKeyServerKmip` método para probar si el servidor de claves de protocolo de interoperabilidad de gestión de claves (KMIP) especificado es accesible y funciona normalmente.

Parámetros

Este método tiene los siguientes parámetros de entrada:

Nombre	Descripción	Tipo	Valor predeterminado	Obligatorio
KeyServerID	El ID del servidor de claves KMIP que se probará.	entero	Ninguno	Sí

Valores devueltos

Este método no tiene valores devueltos. La prueba se considera correcta si no se devuelve ningún error.

Ejemplo de solicitud

Las solicitudes de este método son similares al ejemplo siguiente:

```
{
  "method": "TestKeyServerKmip",
  "params": {
    "keyServerID": 15
  },
"id": 1
}
```

Ejemplo de respuesta

Este método devuelve una respuesta similar al siguiente ejemplo:

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

Nuevo desde la versión

11,7