Comece a usar o gerenciamento de chaves externas

Element Software

NetApp
November 21, 2024

Índice

Comece a usar o gerenciamento de chaves externas 1
Configurar o gerenciamento de chaves externas 1
Rechavear criptografia de software na chave mestra em repouso 2
Recuperar chaves de autenticação inacessíveis ou inválidas 5
Comandos externos da API de gerenciamento de chaves 5

Comece a usar o gerenciamento de chaves externas

O gerenciamento de chaves externas (EKM) fornece gerenciamento seguro de chaves de autenticação (AK) em conjunto com um servidor de chaves externas (EKS) fora do cluster. Os AKs são utilizados para bloquear e desbloquear unidades de encriptação automática (SEDs) quando "criptografia em repouso" o está ativado no cluster. O EKS fornece geração e armazenamento seguros dos AKs. O cluster utiliza o Key Management Interoperability Protocol (KMIP), um protocolo padrão definido PELA OASIS, para se comunicar com o EKS.

- "Configurar o gerenciamento externo"
- "Rechavear criptografia de software na chave mestra em repouso"
- "Recuperar chaves de autenticação inacessíveis ou inválidas"
- "Comandos externos da API de gerenciamento de chaves"

Encontre mais informações

- "CreateCluster API que pode ser usada para habilitar a criptografia de software em repouso"
- "Documentação do software SolidFire e Element"
- "Documentação para versões anteriores dos produtos NetApp SolidFire e Element"

Configurar o gerenciamento de chaves externas

Você pode seguir estas etapas e usar os métodos da API Element listados para configurar seu recurso de gerenciamento de chaves externas.

O que você vai precisar

- Se você estiver configurando o gerenciamento de chaves externas em combinação com a criptografia de software em repouso, habilitou a criptografia de software em repouso usando o "CreateCluster" método em um novo cluster que não contém volumes.

Passos

1. Estabeleça uma relação de confiança com o servidor de chave externa (EKS).
   a. Crie um par de chaves públicas/privadas para o cluster de elementos que é usado para estabelecer uma relação de confiança com o servidor de chaves chamando o seguinte método de API: "CreatePublicPrivateKeyPair"
   b. Obtenha o pedido de assinatura de certificado (CSR) que a Autoridade de Certificação precisa assinar. O CSR permite que o servidor de chaves verifique se o cluster do elemento que vai acessar as chaves é autenticado como o cluster do elemento. Chame o seguinte método API: "GetClientCertificateSignRequest"
   c. Utilize a EKS/Certificate Authority para assinar a CSR recuperada. Consulte a documentação de terceiros para obter mais informações.
2. Crie um servidor e um provedor no cluster para se comunicar com o EKS. Um provedor de chaves define onde uma chave deve ser obtida, e um servidor define os atributos específicos do EKS que serão

comunicados.

a. Crie um provedor de chaves onde os detalhes do servidor de chaves residirão chamando o seguinte método de API: "CreateKeyProviderKmip"

b. Crie um servidor de chaves fornecendo o certificado assinado e o certificado de chave pública da Autoridade de Certificação chamando os seguintes métodos de API: "CreateKeyServerKmip" "TestKeyServerKmip"

   Se o teste falhar, verifique a conetividade e a configuração do servidor. Em seguida, repita o teste.

c. Adicione o servidor de chaves ao contentor do provedor de chaves chamando os seguintes métodos de API: "AddKeyServerToProviderKmip" "TestKeyProviderKmip"

   Se o teste falhar, verifique a conetividade e a configuração do servidor. Em seguida, repita o teste.

3. Execute uma das seguintes ações como próxima etapa para criptografia em repouso:

   a. (Para criptografia de hardware em repouso) ative "criptografia de hardware em repouso"fornecendo a ID do provedor de chaves que contém o servidor de chaves usado para armazenar as chaves chamando o "EnableEncryptionAtRest"método API.

      É necessário habilitar a criptografia em repouso por meio do "API". Ativar a criptografia em repouso usando o botão UI do elemento existente fará com que o recurso reverta para o uso de chaves geradas internamente.

   b. (Para criptografia de software em repouso) para "criptografia de software em repouso"utilizar o provedor de chaves recém-criado, passe o ID do provedor de chaves para o "RekeySoftwareEncryptionAtRestMasterKey"método API.

Encontre mais informações

- "Ativar e desativar a encriptação para um cluster"
- "Documentação do software SolidFire e Element"
- "Documentação para versões anteriores dos produtos NetApp SolidFire e Element"

Rechavear criptografia de software na chave mestra em repouso

Você pode usar a API Element para rechavear uma chave existente. Esse processo cria uma nova chave mestra de substituição para o servidor de gerenciamento de chaves externo. As chaves mestras são sempre substituídas por novas chaves mestras e nunca duplicadas ou substituídas.

Você pode precisar de rechavear como parte de um dos seguintes procedimentos:

- Crie uma nova chave como parte de uma alteração do gerenciamento de chaves internas para o gerenciamento de chaves externas.
- Crie uma nova chave como reação ou como proteção contra um evento relacionado à segurança.

Este processo é assíncrono e retorna uma resposta antes que a operação de rechavear esteja concluída. Você pode usar o "GetAsyncResult" método para poll o sistema para ver quando o processo foi concluído.

O que você vai precisar

- Você ativou a criptografia de software em repouso usando o "CreateCluster" método em um novo cluster que não contém volumes e não tem e/S Use `GetSoftwareEncryptionatRestInfo` para confirmar que o estado está `enabled` antes de prosseguir.
- Você tem "estabeleceu uma relação de confiança" entre o cluster SolidFire e um servidor de chave externa (EKS). Execute o "TestKeyProviderKmip" método para verificar se uma conexão com o provedor de chaves está estabelecida.

Passos

1. Execute o "ListKeyProvidersKmip" comando e copie o ID do provedor de chaves (`keyProviderID`).
2. Execute o "RekeySoftwareEncryptionAtRestMasterKey" com o `keyManagementType` parâmetro como `external` e `keyProviderID` como o número de ID do provedor de chaves da etapa anterior:

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

3. Copie o `asyncHandle` valor da `RekeySoftwareEncryptionAtRestMasterKey` resposta do comando.
4. Execute o "GetAsyncResult" comando com o `asyncHandle` valor da etapa anterior para confirmar a alteração na configuração. A partir da resposta do comando, você deve ver que a configuração de chave mestra mais antiga foi atualizada com novas informações de chave. Copie a nova ID do provedor de chaves para uso em uma etapa posterior.

```
{
   "id": null,
   "result": {
     "createTime": "2021-01-01T22:29:18Z",
     "lastUpdateTime": "2021-01-01T22:45:51Z",
     "result": {
       "keyToDecommission": {
         "keyID": "<value>",
         "keyManagementType": "internal"
     },
     "newKey": {
       "keyID": "<value>",
       "keyManagementType": "external",
       "keyProviderID": <value>
     },
     "operation": "Rekeying Master Key. Master Key management being
transferred from Internal Key Management to External Key Management with
keyProviderID=<value>",
     "state": "Ready"
   },
   "resultType": "RekeySoftwareEncryptionAtRestMasterKey",
   "status": "complete"
}
```

5. Execute o `GetSoftwareEncryptionatRestInfo` comando para confirmar que os novos detalhes da chave, incluindo o `keyProviderID`, foram atualizados.

```
{
   "id": null,
   "result": {
     "masterKeyInfo": {
       "keyCreatedTime": "2021-01-01T22:29:18Z",
       "keyID": "<updated value>",
       "keyManagementType": "external",
       "keyProviderID": <value>
     },
     "rekeyMasterKeyAsyncResultID": <value>
     "status": "enabled",
     "version": 1
   },
}
```

Encontre mais informações

- "Gerencie o storage com a API Element"
- "Documentação do software SolidFire e Element"
- "Documentação para versões anteriores dos produtos NetApp SolidFire e Element"

Recuperar chaves de autenticação inacessíveis ou inválidas

Ocasionalmente, pode ocorrer um erro que requer a intervenção do utilizador. Em caso de erro, será gerada uma avaria no cluster (designada por código de avaria do cluster). Os dois casos mais prováveis são descritos aqui.

O cluster não consegue desbloquear as unidades devido a uma falha do cluster KmipServerFault.

Isso pode ocorrer quando o cluster inicializa pela primeira vez e o servidor de chaves está inacessível ou a chave necessária não está disponível.

1. Siga as etapas de recuperação nos códigos de falha do cluster (se houver).

Uma falha sliceServiceUnHealthy pode ser definida porque as unidades de metadados foram marcadas como com falha e colocadas no estado "disponível".

Passos para limpar:

1. Adicione as unidades novamente.
2. Após 3 a 4 minutos, verificar se a `sliceServiceUnhealthy` avaria foi apagada.

Consulte "códigos de falha do cluster" para obter mais informações.

Comandos externos da API de gerenciamento de chaves

Lista de todas as APIs disponíveis para gerenciar e configurar EKM.

Usado para estabelecer uma relação de confiança entre o cluster e servidores externos de propriedade do cliente:

- CreatePublicPrivateKeyPair
- GetClientCertificateSignRequest

Usado para definir os detalhes específicos de servidores externos de propriedade do cliente:

- CreateKeyServerKmip
- ModifyKeyServerKmip
- DeleteKeyServerKmip
- GetKeyServerKmip
- ListKeyServersKmip
- TestKeyServerKmip

Usado para criar e manter provedores-chave que gerenciam servidores de chave externos:

- CreateKeyProviderKmip
- DeleteKeyProviderKmip
- AddKeyServerToProviderKmip
- RemoveKeyServerFromProviderKmip
- GetKeyProviderKmip
- ListKeyProvidersKmip
- RekeySoftwareEncryptionAtRestMasterKey
- TestKeyProviderKmip

Para obter informações sobre os métodos de API, "Informações de referência da API"consulte .

Informações sobre direitos autorais

Informações sobre marcas comerciais