보안 **Astra** 데이터 저장소

Astra Data Store

NetApp
June 15, 2022

목차

보안 **Astra** 데이터 저장소

보안 인증서를 관리합니다

Astra Data Store는 클러스터의 소프트웨어 구성 요소 간에 MTL(Mutual Transport Layer Security) 암호화를 사용합니다. 각 Astra Data Store 클러스터에는 자체 서명된 루트 CA 인증서("astrads-cert-root")와 중간 CA 인증서("astrads-cert-<cluster_name>")가 있습니다. 이 인증서는 Astra Data Store 운영자가 관리합니다. 운영자는 만료 날짜 7일 전에 각 인증서를 자동으로 갱신합니다. 인증서를 수동으로 취소할 수도 있습니다.

인증서를 해지합니다

Astra Data Store 컨트롤러, 노드 또는 CA 인증서가 손상된 경우 MTL 암호를 삭제하여 이를 취소할 수 있습니다. 이렇게 하면 Astra Data Store 운영자가 자동으로 새 인증서를 발급합니다. 언제든지 Astra Data Store 인증서를 해지할 수 있습니다.

CA 인증서를 해지하면 해당 CA에서 서명한 인증서가 해지됩니다.

단계

1. Astra Data Store 클러스터의 컨트롤러 노드에 로그인합니다.
2. 시스템에 있는 기존 인증서를 나열합니다. 예를 들면 다음과 같습니다.

```
kubectl get secrets -n astrads-system | grep astrads-cert
```

출력은 다음과 비슷해야 합니다.

```
astrads-cert-astrads-cluster-controller
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-ds-dms-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-ds-support-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-astrads-support-astrads-cluster-f23d158
kubernetes.io/tls     4      6d6h
astrads-cert-root
kubernetes.io/tls     4      6d6h
astrads-cert-sti-net-com
kubernetes.io/tls     5      6d6h
```

3. 출력에서 취소할 인증서의 이름을 기록합니다.
4. kubeck 유틸리티를 사용하여 인증서를 해지하고 certificate_name을 인증서 이름으로 바꿉니다. 예를 들면 다음과 같습니다.

```
kubectl delete secret CERTIFICATE_NAME -n astrads-system
```

기존 인증서가 해지되고 대신 새 인증서가 자동으로 생성됩니다.

외부 키 관리

하나 이상의 외부 키 관리 서버를 사용하여 클러스터가 암호화된 데이터에 액세스하는 데 사용하는 키를 보호할 수 있습니다. 외부 키 관리 서버는 KMIP(Key Management Interoperability Protocol)를 사용하여 노드에 키를 제공하는 스토리지 환경의 타사 시스템입니다.

Astra Data Store는 Astra Data Store 클러스터를 생성할 때 기본적으로 내부 키 제공업체와 함께 Software Encryption at Rest(sear)를 활성화합니다.

키 관리에는 다음과 같은 사용자 정의 리소스 정의(CRD)가 포함됩니다.

- * AstraDSKeyProvider *: 외부 KMIP 서버를 구성합니다. 이는 서버 클러스터일 수 있습니다.
- * AstraDSSEARKeyRotate *: 키 공급자에서 새 키 암호화 키를 가져옵니다. 이 키를 Astra Data Store에 제공합니다.

외부 키 관리와 관련된 다음 작업을 수행할 수 있습니다.

- [Set up external key management]
- [Check the software encryption at rest status]
- [Change external to internal key management]
- [Rotate keys for security]

외부 키 관리를 설정합니다

Astra Data Store에서 외부 키 관리를 설정하는 것은 kubeck asts 명령어를 사용한다.

예를 들어, OpenSSL을 사용하여 외부 키를 설정할 수 있도록 클러스터 또는 KMIP 서버에 SSL 인증서가 필요합니다.

단계

1. 키 공급자 클라이언트에 대한 인증서를 준비합니다. 클라이언트 인증서, 클라이언트 개인 키 및 신뢰 CA 번들을 포함합니다.

   예를 들어, OpenSSL을 사용하여 외부 키를 설정할 수 있도록 클러스터 또는 KMIP 서버에 SSL 인증서를 준비합니다.

2. Astra Data Store 클러스터의 노드 중 하나에 로그인합니다.
3. 다음 kubeck extension 명령을 입력하여 Astra Data Store 클러스터의 주요 공급자를 구성합니다.

```
kubectl-astrads key-provider certs --key key.pem
--client-cert client_cert.pem --ca-cert server_ca.pem
--hostnames=<kmip_server_ip> <key_provider_cr_name>
--namespace astrads-system --cluster <ads_cluster_name>
```

다음 예에서는 ADS 클러스터 "astrads-cluster-f23d158"에 대해 "hashicorp"라는 외부 키 공급자를 구성합니다.

```
kubectl-astrads key-provider certs --key key.pem
--client-cert client_cert.pem --ca-cert server_ca.pem
--hostnames=10.235.nnn.nnn hashicorp
--namespace astrads-system --cluster astrads-cluster-f23d158
```

1. AstraDSCluster CR을 통해 sear에 외부 키 관리자를 사용하도록 Astra Data Store 클러스터를 구성합니다. 도움말을 표시합니다.

   ```
   kubectl-astrads clusters sears -h
   ```

   응답:

```
Configure SEARS in AstraDS cluster

Usage:
  astrads clusters sears [flags]

Flags:
  -d, --duration string   Duration for key rotation (default "2160h")
  -h, --help              help for sears


Global Flags:
      --ads-cluster-name string        Name of the ADS Cluster
      --ads-cluster-namespace string   Namespace of the ADS Cluster
      ...
```

다음 명령을 실행하면 Astra Data Store 클러스터가 sear의 Key Manager로 "AstraDSKeyProvider hashicorp"를 사용하도록 구성됩니다. 이 명령은 또한 기본값 90일(2160시간)의 키 회전 시간을 사용합니다.

```
kubectl-astrads clusters sears -d 500h hashicorp
--ads-cluster-name=astrads-cluster-f23d158
--ads-cluster-namespace=astrads-system
```

소프트웨어 암호화 유휴 상태를 확인합니다

저장된 소프트웨어 암호화 구성을 확인할 수 있습니다.

단계

1. AstraDSCluster CR을 검사합니다.

```
Name:         astrads-cluster-f23d158
Namespace:    astrads-system
Labels:       <none>
Annotations:  <none>
API Version:  astrads.netapp.io/v1beta1
Kind:         AstraDSCluster
...
Spec:
...
  Software Encryption At Rest:
    Ads Key Provider:     hashicorp
    Key Rotation Period:  500h0m0s
...
Status:
...
  Software Encryption At Rest Status:
    Key Active Time:       2022-05-16T15:53:47Z
    Key Provider Name:     hashicorp
    Key Provider UUID:     ccfc2b0b-dd98-5ca4-b778-99debef83550
    Key UUID:              nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn
```

내부 키 관리를 외부로 변경합니다

현재 외부 키 관리자를 사용 중인 경우 내부 키 관리자로 변경할 수 있습니다.

단계

1. SoftwareEncryptionAtRest 구성을 제거하여 AstraDSCluster CR을 변경합니다.
2. (선택 사항) 이전 AstraDSKeyProvider 및 관련 암호를 삭제합니다.

이전 키 공급자와 암호는 자동으로 제거되지 않습니다.

보안을 위해 키를 회전합니다

키 로테이션을 통해 보안이 강화됩니다. 기본적으로 Astra Data Store는 90일마다 자동으로 키를 순환합니다. 기본 설정을 변경할 수 있습니다. 또한 필요할 때 키를 회전할 수도 있습니다.

자동 키 회전을 구성합니다

1. CRD에서 AstraDSSEARKeyRotate 매개변수를 업데이트합니다.

```
kubectl patch astradscluster astrads-cluster-f23d158
-n astrads-system
--type=merge -p '{"spec": {"softwareEncryptionAtRest": {
"keyRotationPeriod": "3000h"}}}'
```

주문형 키 회전을 구성합니다

1. 키를 회전하기 위해 AstraDSSEARKeyRotateRequest CR을 생성합니다.

```
cat << EOF | kubectl apply -f  -
apiVersion: astrads.netapp.io/v1beta1
kind: AstraDSSEARKeyRotateRequest
metadata:
  name: manual
  namespace: astrads-system
spec:
  cluster: astrads-cluster-f23d158
EOF
```