7.5 시크릿으로 민감한 데이터를 컨테이너에 전달

생성일	@2021년 4월 25일 오전 2:24
태그	

1. 시크릿 소개

쿠버네티스는 시크릿이라는 별도 오브젝트를 제공해서 설정안에 보안을 유지하는 방법을 제공

- 환경변수로 시크릿 항목을 컨테이너에 전달
- 시크릿 항목을 볼륨파일로 노출

쿠버네티스는 시크릿에 접근해야하는 파드가 실행되고 있는 노드에만 개별 시크릿을 배포해 시크릿을 안전하게 유지

+) 노드 자체적으로 시크릿을 항상 메모리에만 저장햐고 물리저장소에 기록되지 않게 함

마스터 노드를 보호하는 것이 필요

마스터 노드(etcd) : 시크릿을 암호화되지 않은 형식으로 저장

→ 이 데이터를 보호하려면 마스터 노드를 보호해야함

→ 권한없는 사용자가 api서버를 이용하지 못하게 하는 것도 포함된다

- etcdd가 시크릿을 암호화된 형태로 저장해서 시스템을 좀 더 안전하게 만듦 → `언제 시크릿`을 사용하고 `어떤 컨피그맵`을 사용할지 선택하는 것이 필요

[방법]

1. 민감하지 않고, 일반 설정 데이터 → 컨피그맵
2. 민감한 데이터 → 시크릿 (을 사용해서 키 아래에 보관)
3. 설정파일이 둘 다 갖고 있다면? 설정파일을 시크릿 안에 저장

2. 기본 토큰 시크릿 소개

모든 실행 컨테이너가 마운트해서 갖고 있는 시크릿을 살펴보면...

`kubectl get secrets`

`kubectl describe secrets` → 시크릿 볼륨이 마운트된것을 보여줌

시크릿은 컨피그맵과 비슷

- 시크릿 볼륨이 마운트된 디렉터리의 세 개의 파일 볼 수 있다
- kubectl exec mypod ls /var/run/secrets/kubernetes.io/serviceaccount

  ca.crt

  namespace

  token

3. 시크릿 생성

먼저 필요한것...

- 개인 키 파일
  - openssl genrsa -out https.key 2048
- 인증서
  - openssl req 0new -x509 -key https.key -out https.cert -days 3650 -subj /CN=www.kubia-example.com

echo bar > foo 로 추가 더미파일에 bar문자열 저장..

```
kubectl create secret generic fortune-https --from-file=https.key --from-file=https.cert --from-file=foo
```

위 명령으로 세가지 파일에서 시크릿 생성

`--from-file=fortune-https` 옵션을 이용해 개별 파일을 지정하는 대신 디렉터리 전체 포함시킬 수 있음

4. 컨피그맵과 시크릿 비교

`kubectl get <이름>` 으로 갖고오는 정보를 비교해보면..

시크릿 항목의 내용

- base64 인코딩 문자열로 표시
- 시크릿 항목에 일반 텍스트 뿐만 아니라 바이너리 값도 담을 수 있기때문에 base64사용
  - 바이너리 데이터를 yaml, json 안에 base64 인코딩으로 넣을 수 있음

컨피그맵의 내용

- 일반 텍스트

stringData 필드 소개

모든 민감한 데이터가 바이너리 형태는 아님 → 시크릿의 값을 stringData 필드로도 설정 가능

```
kind: Secret
apiVersion: v1
stringData:
  foo: plain text
data:
  https.cert ......
```

plain text는 base64로 인코딩 되지 않는다

5. 파드에서 시크릿 사용

인증서와 키 파일 모두 포함하는 fortune-https 시크릿을 nginX에서 사용할 수 있도록 설정하는 것이 필요

컨피그맵 수정

https활성화 위해 fortune-config 컨피그맵 수정

```
kubectl edit configmap fortune-config
```

`data:` 하위의 `my-nginx-config.conf :` 를 서버 목록(이름,경로,포트)으로 수정

fortune-https 시크릿을 파드에 마운트

인증서와 키를 갖고 있는 시크릿 볼륨을 web-server 컨테이너 안에 적당한 위치에 마운트 한다