챕터3 쿠버네티스 유저 관리

Created 2021년 11월 6일 오후 2:48

Tags 비어 있음

▼ 목차

유저와 서비스어카운트

스태틱토큰 연습문제

- 다음 csv 파일을 생성하고 apiserver에 토큰을 등록하여 재시작하자.

  vim somefile.csv

  ```
  password1,user1,uid001,"group1" password2,user2,uid002
  password3,user3,uid003 password4,user4,uid004
  ```

- 마스터 노드에 문제가 발생하는가? 어떤 서비스가 문제가 발생하는가?문제가 발생하는 서비스의 컨테이너를 확인하기 위해 docker logs를 사용하고 그 해결 방법을 찾으라.

  ▼ 풀이

  cp somefile.csv /etc/kubernetes/pki/

  vim /etc/kubernetes/manifests/kube-apiserver.yaml

  kubectl get pod

- 정상적으로 서비스가 시작됐다면 kubectl에 유저 정보를 등록하고 등록한 유저 권한으로 kubectl get pod 요청을 수행하라.
  - 반드시 Forbidden이라는 결과가 나와야 함 (유저는 생성됐으나 권한은 없다)

  ▼ 풀이

  ```
  kubectl config set-credentials user1 --token=password1 kubectl config
  set-context user1-context --cluster=kubernetes \ --namespace=frontend -
  -user=user1 kubectl get pod --user user1
  ```

서비스어카운트 생성과 사용

서비스어카운트 생성

```
kubectl create serviceaccount sa1
```

서비스 어카운트와 함께 시크릿을 사용해 토큰이 생성됨

```
kubectl get sa,secret
```

새로 생성한 서비스어카운트를 사용하는 파드를 다음과 같이 구성

```
cat <<EOF | kubectl apply -f - apiVersion: v1 kind: Pod metadata: name: nx
spec: serviceAccountName: sa1 containers: - image: nginx name: nx
imagePullPolicy: IfNotPresent EOF
```

서비스 어카운트 정보는 다음 디렉토리에 자동으로 마운트됨 - 오토마운트 옵션을 비활성화가 가능

```
kubectl exec -it nx -- bash # ls /var/run/secrets/kubernetes.io/serviceaccount
```

토큰을 사용해 요청을 수행하면 정상적으로 통신이 가능하며 토큰을 사용하지 않으면 403이 나타남

```
TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token) curl -X GET h
ttps://$KUBERNETES_SERVICE_HOST/api --header "Authorization: Bearer $TOKEN" --
insecure
```

네임스페이스에 있는 POD 조회하는 명령어 실행

```
curl -X GET https://$KUBERNETES_SERVICE_HOST/api/v1/namespaces/default/pods --
header "Authorization: Bearer $TOKEN" -k
```

서비스어카운트 연습문제

- http-go라는 이름을 가진 ServiceAccount를 생성하고 http-go 파드를 생성해 http-go 서비스 어카운트를 사용하도록 설정하라.
- 서비스어카운트를 사용해 디플로이먼트를 조회해보자.

▼ 풀이

파드와 서비스어카운트를 구성한다.

```
cat <<EOF | kubectl apply -f - --- # kubectl create sa http-go --dry-run=c
lient -o yaml apiVersion: v1 kind: ServiceAccount metadata: creationTimest
amp: null name: http-go --- apiVersion: v1 kind: Pod metadata: name: http-
go spec: serviceAccountName: http-go containers: - image: gasbugs/http-go
name: http-go imagePullPolicy: IfNotPresent EOF
```

생성된 파드를 사용해서 디플로이먼트 리스트 요청도 수행해본다.

```
kubectl exec -it http-go -- bash # GET /apis/apps/v1/namespaces/{namespac
e}/deployments TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/t
oken) curl -X GET https://$KUBERNETES_SERVICE_HOST/api/v1/namespaces/defau
lt/deployments --header "Authorization: Bearer $TOKEN" -k
```

TLS를 활용한 유저 인증서 발급

개인키, CSR 생성 후 전자서명

```
# 개인키 생성 openssl genrsa -out gasbugs.key 2048 # csr 생성 openssl req -new -ke
y gasbugs.key -out gasbugs.csr -subj "/CN=gasbugs/O=boanproject" # csr을 사용해
CA 권한으로 전자서명해 CRT 발급 openssl x509 -req -in gasbugs.csr -CA /etc/kubernete
s/pki/ca.crt -CAkey /etc/kubernetes/pki/ca.key -CAcreateserial -out gasbugs.cr
t -days 365 # csr은 더이상 필요하지 않으므로 삭제 rm gasbugs.csr
```

kubectl 설정에서 유저 정보 등록

```
# 유저 정보 생성 kubectl config set-credentials gasbugs --client-certificate=gasb
ugs.crt --client-key=gasbugs.key # 유저와 클러스터 정보 연결 kubectl config set-cont
ext gasbugs@kubernetes --cluster=kubernetes --namespace=office --user=gasbugs
# 컨텍스트를 사용해 접속 kubectl --context=gasbugs@kubernetes get pods # 컨텍스트 스위
칭을 사용 kubectl config use-context gasbugs@kubernetes # 관리자 권한으로 돌아오기 kub
ectl config use-context kubernetes-admin@kubernetes
```

연습문제

dev1팀에 john이 참여했다. john을 위한 인증서를 만들고 승인해보자.

RBAC를 활용한 권한 할당

gasbugs에게 파드, 디플로이먼트 읽기 할당하기

```
kubectl create ns office cat <<EOF | kubectl apply -f - apiVersion: rbac.autho
rization.k8s.io/v1 kind: Role metadata: namespace: office name: pod-deploy-rea
der rules: - apiGroups: ["apps", ""] # "" indicates the core API group resourc
es: ["pods", "deployments"] verbs: ["get", "watch", "list"] --- apiVersion: rb
ac.authorization.k8s.io/v1 kind: RoleBinding metadata: name: read-pods-deploys
namespace: office subjects: - kind: User name: gasbugs apiGroup: rbac.authoriz
ation.k8s.io roleRef: kind: Role name: pod-deploy-reader apiGroup: rbac.author
ization.k8s.io EOF
```

연습문제

john 유저에게 dev1 네임스페이스에 대한 파드, 레플리카셋, 디플로이먼트를 조회, 생성하고 삭제할 수 있는 권한을 부여하라. deploy-manager

```
kubectl create ns dev1 cat <<EOF | kubectl apply -f - apiVersion: rbac.authori
zation.k8s.io/v1 kind: Role metadata: namespace: dev1 name: deploy-manager rul
es: - apiGroups: ["apps", ""] # "" indicates the core API group resources: ["p
ods", "deployments", "replicasets"] verbs: ["get", "watch", "list", "create",
"delete"] --- apiVersion: rbac.authorization.k8s.io/v1 kind: RoleBinding metad
ata: name: manage-deploys namespace: dev1 subjects: - kind: User name: john ap
iGroup: rbac.authorization.k8s.io roleRef: kind: Role name: deploy-manager api
Group: rbac.authorization.k8s.io EOF
```