нетология

Создание и использование секретов

Сергей
Андрюнин

Сергей Андрюнин

DevOps-инженер

RTLabs

Сергей Андрюнин

План занятия

1. Общие сведения о секретах
2. Создание
3. Хранение
4. Доступ
5. Использование
6. Итоги
7. Домашнее задание

Общие сведения

Общие сведения о секретах

- предназначены для хранения конфиденциальной информации;
- неопределенный размер;
- неопределенный состав;
- требования безопасности;
- ролевая модель доступа к различным секретам.

Хранение конфиденциальной информации

- пароли;
- ключи;
- токены;
- сертификаты.

Неопределенный размер

От двоичного типа размером в один бит, до большого файла.

Неопределенный состав

- число;
- строка;
- список;
- словарь;
- непечатаемая последовательность символов, закодированная, например, в base64;
- файл.

Требования безопасности

- общие требования;
- руководящие документы;
- локальные правила и распоряжения.

Ролевая модель доступа к различным секретам

- владелец или суперпользователь;
- системный администратор;
- верхнеуровневый менеджер;
- менеджер отдела, производства, региона и т.д.;
- работники, программисты, субподрядные организации и многие другие.

Создание

Создание

- человеком: логины, пароли;
- системой: генераторы ключей, токенов, url, файлы;
- от одной системы к другой: OAuth2, OpenID, CSR.

Типы секретов

- **generic** – создание секрета из литеральных значений, файла или каталога;
- **tls** – секрет из пары открытого и закрытого ключей;
- **docker-registry** – секрет для доступа к хранилищу образов Docker.

Создание из командной строки

```
echo ‘admin’  > username.txt
echo ‘password’ > password.txt
kubectl create secret generic user-cred \
--from-file=./username.txt --from-file=./password.txt
```

Создание из файла

```
$ echo 'admin' | base64
YWRtaW4K
$ echo 'password' | base64
cGFzc3dvcmQK
```

Создаем файл secret.yml

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  superuser: YWRtaW4K
  password: cGFzc3dvcmQK
```

Импортируем в Kubernetes

```
kubectl apply -f secret.yml
```

Создание tls

```
$ openssl genrsa -out cert.key 4096
$ openssl req -x509 -new -key cert.key -days 3650 -out cert.crt \
-subj '/C=RU/ST=Moscow/L=Moscow/CN=server.local’
$ kubectl create secret tls domain-cert --cert=cert.crt
--key=cert.key
```

Хранение

Хранение

- хранение в кодировке base64;
- размер не более 1MB, а с учётом base64 не более 750kB;
- создание множества мелких секретов может истощить память;
- секреты находятся в пространстве имён, что означает доступ модулей только из того же пространства имён.

Доступ

Особенности

- секреты в kubernetes создаются в виде объектов типа secret;
- секреты очень похожи на словари конфигурации configMap (ассоциативные массивы), но при этом для работы с манифестами YAML и JSON для секретов нужна конвертация в base64.

Просмотр tls

```
$ kubectl get secret domain-cert -o yaml
```

```
apiVersion: v1
data:
 tls.crt: LS0t...
 tls.key: LS0t...
kind: Secret
metadata:
 creationTimestamp: "2021-06-01T18:29:43Z"
 managedFields:
 - apiVersion: v1
   fieldsType: FieldsV1
   fieldsV1:
     f:data:
       .: {}
       f:tls.crt: {}
       f:tls.key: {}
     f:type: {}
   manager: kubectl-create
   operation: Update
   time: "2021-06-01T18:29:43Z"
 name: domain-cert
 namespace: default
 resourceVersion: "1428"
 uid: d5188e5b-1d41-407e-8a53-57adf2331ee1
type: kubernetes.io/tls
```

Использование

Особенности

- секреты должны быть созданы до того, как они будут использованы в модулях. Ссылки на несуществующие секреты предотвратят запуск Pod;
- секрет можно использовать как файлы в смонтированном томе;
- секрет можно использовать как переменную среды.

Пример монтирования сертификатов

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
  - name: nginx
    image: nginx:latest
    ports:
    - containerPort: 80
      protocol: TCP
    - containerPort: 443
      protocol: TCP
    volumeMounts:
    - name: certs
      mountPath: "/etc/nginx/ssl"
      readOnly: true
    - name: config
      mountPath: /etc/nginx/conf.d
      readOnly: true
  volumes:
  - name: certs
    secret:
      secretName: domain-cert
  - name: config
    configMap:
      name: nginx-config
```

Итоги

Сегодня мы изучили:

- что такое секреты;
- как их создавать и использовать в kubernetes.

Домашнее задание

Давайте посмотрим ваше домашнее задание.

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

нетология

Задавайте вопросы и пишите отзыв о лекции!

Сергей Андрюнин

Сергей Андрюнин