

**Национальный исследовательский Университет
ИТМО**

Факультет инфокоммуникационных технологий

Дисциплина: «Введение в распределенные технологии»

Лабораторная работа №1

"Установка Docker и Minikube, мой первый манифест."

Выполнил студент:

Сагайдак Алина Алексеевна,

K4111c

Проверил преподаватель:

Филянин Иван Викторович

Санкт-Петербург, 2024 год

Описание

Это первая лабораторная работа в которой вы сможете протестировать Docker, установить Minikube и развернуть свой первый "под".

Цель работы

Ознакомиться с инструментами Minikube и Docker, развернуть свой первый "под".

Ход работы

1. Установить Docker на рабочий компьютер;
2. Установить Minikube;
3. Развернуть minikube cluster;
4. Скачать образ HashiCorp Vault;
5. Написать manifest для развертывания пода с образом HashiCorp Vault;
6. Создать сервис для доступа к данному контейнеру;
7. Прокинуть порт для доступа к контейнеру;
8. Найти токен для доступа к vault
9. Войти, используя токен.

Выполнение работы

1. Установка Docker

Так как Docker у меня был установлен заранее, то его наличие подтверждаю следующей командой:

```
[asagaydak@alinas-air lab1 % docker --version  
Docker version 24.0.6, build ed223bc
```

2. Установка Minikube

Аналогично и с Minikube

```
[asagaydak@alinas-air lab1 % minikube version  
minikube version: v1.34.0  
commit: 210b148df93a80eb872ecbeb7e35281b3c582c61
```

3. Развернуть minikube cluster

При помощи команды `minikube start`, можем запустить `minikube cluster` в рамках которого будут располагаться поды

```
[asagaydak@alinas-air lab1 % minikube start
🐶 minikube v1.34.0 on Darwin 13.5.2 (arm64)
🌟 Using the docker driver based on existing profile
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🚚 Pulling base image v0.0.45 ...
🔄 Restarting existing docker container for "minikube" ...
🐳 Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
🔍 Verifying Kubernetes components...
   ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🏁 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

4. Скачать образ HashiCorp Vault;

Для того, чтобы правильно описать под, нужно предварительно скачать образ HashiCorp Vault с помощью команды `docker pull vault:1.13.3`, где версия 1.13.3 Vault - это последняя актуальная версия

```
[asagaydak@alinas-air lab1 % docker pull vault:1.13.3
1.13.3: Pulling from library/vault
Digest: sha256:f98ac9dd97b0612746630033771bc7a8c86408a44a056f3f4be47fc576ec3744
Status: Image is up to date for vault:1.13.3
docker.io/library/vault:1.13.3
```

5. Написать manifest для развертывания пода с образом HashiCorp Vault;

После того, как было определено какой образ какой версии будем запускать и на каком порту, можно приступить к написанию manifest, описывающего сущность пода vault для кластера

```

1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: vault
5    labels:
6      name: vault
7  spec:
8    containers:
9      - name: vault
10       image: vault:1.13.3
11       ports:
12       - containerPort: 8200

```

здесь apiVersion - это версия api

kind - абстракция, которую мы запускаем (например, под)

metadata.name - имя пода, который запускаем (например, есть 3 пода с разными именами, внутри которых запущено одно и то же приложение)

metadata.labels.name - метка, по которой можно обратиться к любому из этих 3х подов или что-то с ними сделать

spec - описание каким должно быть приложение

containers - нужно запускать контейнеры в рамках этого пода

name - название изображения

image - название докер изображения с указанием версии

ports - порт на котором будет запущено изображение

6. Создать сервис для доступа к данному контейнеру;

Применяем манифест пода vault в нашем кластере: minikube

kubectl -- apply -f pods.yaml . Тем самым создается под

```

[asagaydak@alinas-air lab1 % minikube kubectl -- apply -f pods.yaml
pod/vault created

```

Данная команда создает сервис для пода vault. Сервис будет перенаправлять трафик на под vault через порт 8200.

```
minikube kubectl -- expose pod vault --type=NodePort
--port=8200
```

```
asagaydak@alinas-air lab1 % minikube kubectl -- expose pod vault --type=NodePort --port=8200
service/vault exposed
```

7. Прокинуть порт для доступа к контейнеру;

Для того, чтобы кластер мог получать запросы извне (например, с порта localhost:8200), мы должны перенаправить эти запросы на порт в рамках minikube cluster

```
minikube kubectl -- port-forward service/vault 8200:8200
```

```
[asagaydak@alinas-air lab1 % minikube kubectl -- port-forward service/vault 8200:8200
Forwarding from 127.0.0.1:8200 -> 8200
Forwarding from [::1]:8200 -> 8200
```

8. Найти токен для доступа к vault

С помощью команды, указанной ниже, мы можем посмотреть логи пода и в них найти токен для входа в vault

```
minikube kubectl logs vault
```

```
Unseal Key: VNMTQQUnAhDF7m8dda6G4ppfSJfAhDD3Hd1VS6ftoJI=
Root Token: hvs.SXqN2bxW9PphFbP9UfJuRmo0
```

9. Войти, используя токен.

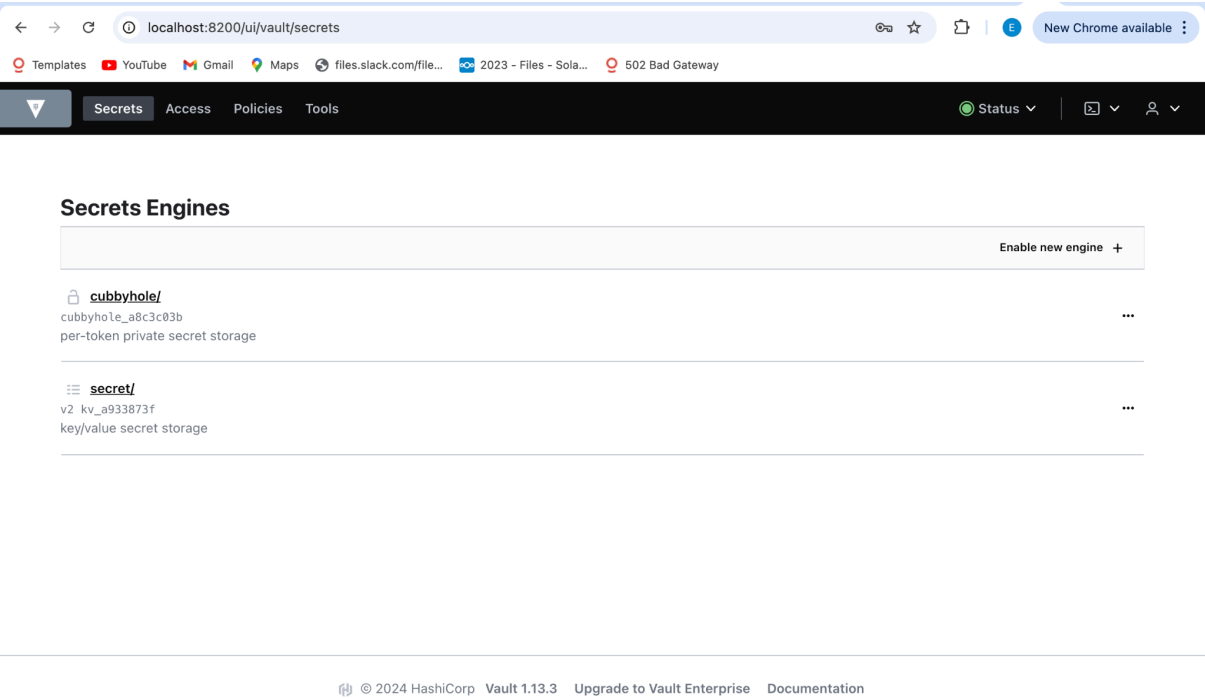


Схема организации контейнеров и сервисов

Схема организации кластера (например, применение манифеста должно пройти по такому пути - либо через ui (OpenLens), либо через sub ctl) для создания пода

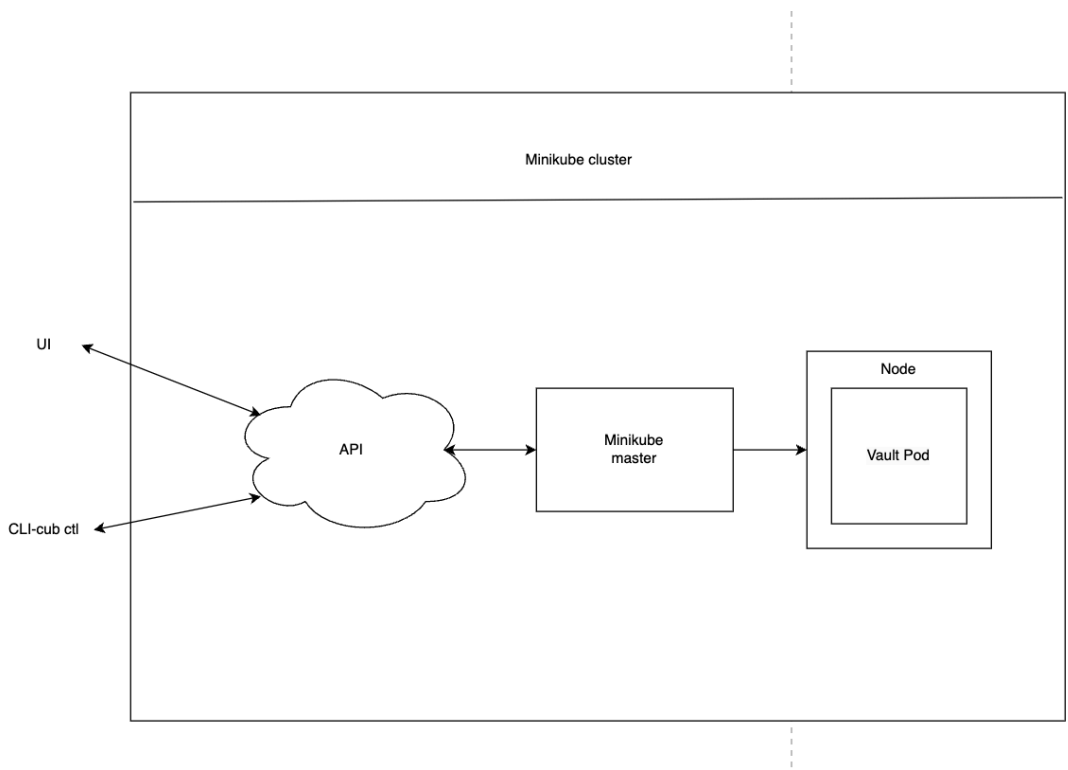


Схема прохождения запроса к поду

