УНИВЕРСИТЕТ ИТМО

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Лабораторная работа №1

Установка Docker и Minikube, мой первый манифест

Выполнил: Фомин Е.Г.

Группа № K4113c

Санкт-Петербург

2022

1) Скачаем образ Vault, проверим его наличие

```
toshiba@toshiba-SATELLITE-L735:~$ sudo docker image ls
[sudo] пароль для toshiba:
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
vault        latest    7193130a202b   13 days ago   221MB
```

2) Запустим Minicube

```
toshiba@toshiba-SATELLITE-L735:~$ minikube start
😄  minikube v1.27.1 на Debian bullseye/sid
✨  Используется драйвер virtualbox на основе существующего профиля
👍  Запускается control plane узел minikube в кластере minikube
🔄  Перезагружается существующий virtualbox VM для "minikube" ...
🐳  Подготавливается Kubernetes v1.25.2 на Docker 20.10.18 ...
    ▪ Используется образ gcr.io/k8s-minikube/storage-provisioner:v5
🔎  Компоненты Kubernetes проверяются ...
🌟  Включенные дополнения: default-storageclass, storage-provisioner
🏄  Готово! kubectl настроен для использования кластера "minikube" и "default" прос
транства имён по умолчанию
```

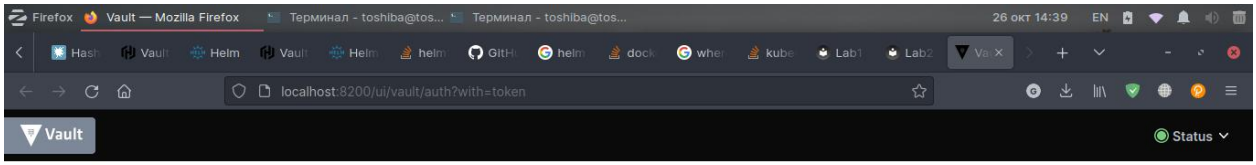
3) Создадим yaml файл для пода

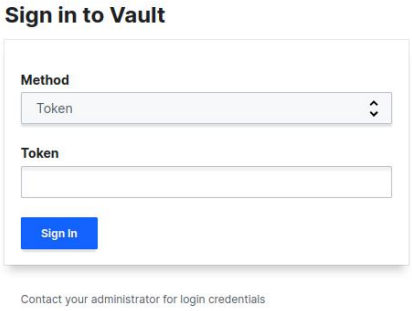
```
apiVersion: v1
kind: Pod
metadata:
   name: vault
   labels:
      run: vault
spec:
   containers:
   - name: vault
     image: vault:latest
```

4) Создадим сам под на основе yaml файла. После убедимся, что он работает, запустим сервер и прокинем порт

```
toshiba@toshiba-SATELLITE-L735:~/Загрузки$ kubectl apply -f vault.yaml
pod/vault created
toshiba@toshiba-SATELLITE-L735:~/Загрузки$ kubectl get pods -o wide
NAME    READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE
 READINESS GATES
vault   1/1     Running   0          15s   172.17.0.3   minikube   <none>
 <none>
toshiba@toshiba-SATELLITE-L735:~/Загрузки$ minikube kubectl -- expose pod vault --t
ype=NodePort --port=8200
service/vault exposed
toshiba@toshiba-SATELLITE-L735:~/Загрузки$ minikube kubectl -- port-forward service
/vault 8200:8200
Forwarding from 127.0.0.1:8200 -> 8200
Forwarding from [::1]:8200 -> 8200
Handling connection for 8200
```

5) Удостоверимся, что все прошло успешно. Зайдём на http://localhost:8200

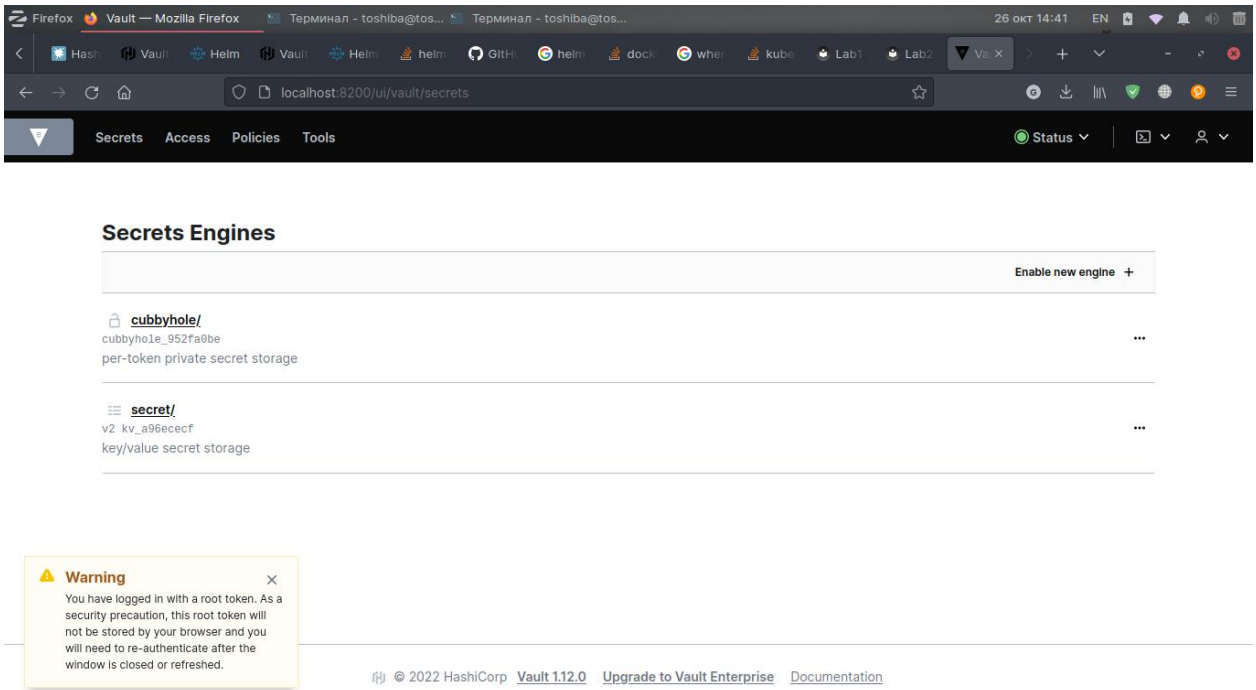




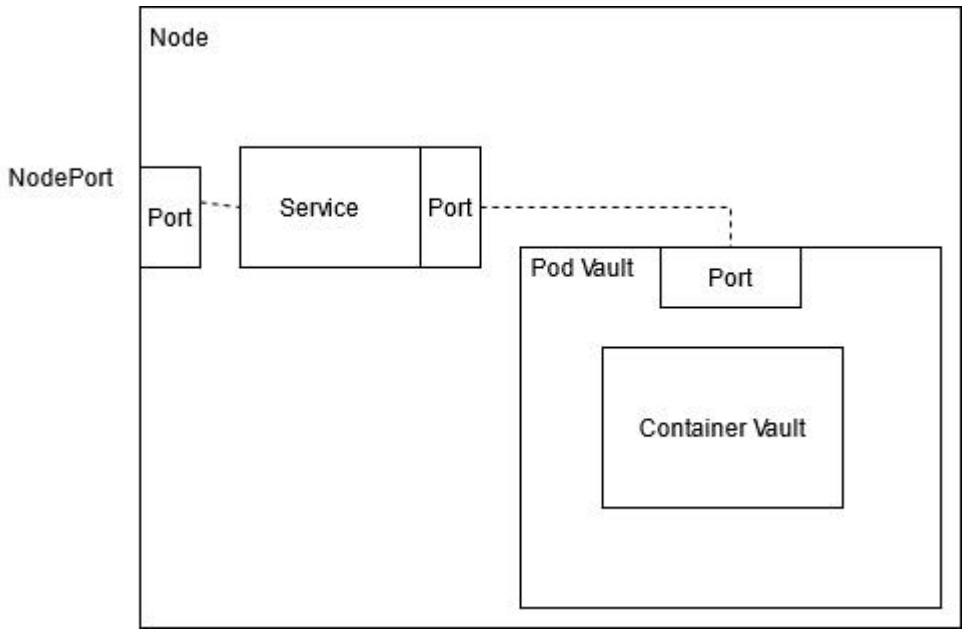
6) Выведем лог, чтобы найти токен для входа, и используем его

```
toshiba@toshiba-SATELLITE-L735:~$ minikube kubectl logs vault
Couldn't start vault with IPC_LOCK. Disabling IPC_LOCK, please use --cap-add IPC_LOCK
==> Vault server configuration:
             Api Address: http://0.0.0.0:8200
                     Cgo: disabled
         Cluster Address: https://0.0.0.0:8201
              Go Version: go1.19.1
              Listener 1: tcp (addr: "0.0.0.0:8200", cluster address: "0.0.0.0:8201",
max_request_duration: "1m30s", max_request_size: "33554432", tls: "disabled")
               Log Level: info
                   Mlock: supported: true, enabled: false
           Recovery Mode: false
                 Storage: inmem
                 Version: Vault v1.12.0, built 2022-10-10T18:14:33Z
             Version Sha: 558abfa75702b5dab4c98e86b802fb9aef43b0eb
==> Vault server started! Log data will stream in below:
2022-10-26T11:37:01.851Z [INFO]  proxy environment: http_proxy="" https_proxy="" no_proxy=""
2022-10-26T11:37:01.852Z [WARN]  no `api_addr` value specified in config or in VAULT_API_ADDR;
falling back to detection if possible, but this value should be manually set
2022-10-26T11:37:01.853Z [INFO]  core: Initializing version history cache for core
2022-10-26T11:37:01.854Z [INFO]  core: security barrier not initialized
2022-10-26T11:37:01.854Z [INFO]  core: security barrier initialized: stored=1 shares=1
threshold=1
2022-10-26T11:37:01.855Z [INFO]  core: post-unseal setup starting
2022-10-26T11:37:01.898Z [INFO]  core: loaded wrapping token key
2022-10-26T11:37:01.898Z [INFO]  core: Recorded vault version: vault version=1.12.0 upgrade
time="2022-10-26 11:37:01.898692842 +0000 UTC" build date=2022-10-10T18:14:33Z
2022-10-26T11:37:01.898Z [INFO]  core: successfully setup plugin catalog: plugin-directory=""
2022-10-26T11:37:01.898Z [INFO]  core: no mounts; adding default mount table
2022-10-26T11:37:01.919Z [INFO]  core: successfully mounted backend: type=cubbyhole
path=cubbyhole/
2022-10-26T11:37:01.919Z [INFO]  core: successfully mounted backend: type=system path=sys/
2022-10-26T11:37:01.920Z [INFO]  core: successfully mounted backend: type=identity
path=identity/
2022-10-26T11:37:01.923Z [INFO]  core: successfully enabled credential backend: type=token
path=token/ namespace="ID: root. Path: "
2022-10-26T11:37:01.924Z [INFO]  core: restoring leases
2022-10-26T11:37:01.924Z [INFO]  rollback: starting rollback manager
2022-10-26T11:37:01.926Z [INFO]  expiration: lease restore complete
2022-10-26T11:37:01.926Z [INFO]  identity: entities restored
```

```
2022-10-26T11:37:01.926Z [INFO]  identity: groups restored
2022-10-26T11:37:02.588Z [INFO]  core: post-unseal setup complete
2022-10-26T11:37:02.589Z [INFO]  core: root token generated
2022-10-26T11:37:02.589Z [INFO]  core: pre-seal teardown starting
2022-10-26T11:37:02.589Z [INFO]  rollback: stopping rollback manager
2022-10-26T11:37:02.589Z [INFO]  core: pre-seal teardown complete
2022-10-26T11:37:02.590Z [INFO]  core.cluster-listener.tcp: starting listener:
listener_address=0.0.0.0:8201
2022-10-26T11:37:02.590Z [INFO]  core.cluster-listener: serving cluster requests:
cluster_listen_address=[::]:8201
2022-10-26T11:37:02.590Z [INFO]  core: post-unseal setup starting
2022-10-26T11:37:02.590Z [INFO]  core: loaded wrapping token key
2022-10-26T11:37:02.590Z [INFO]  core: successfully setup plugin catalog: plugin-directory=""
2022-10-26T11:37:02.591Z [INFO]  core: successfully mounted backend: type=system path=sys/
2022-10-26T11:37:02.592Z [INFO]  core: successfully mounted backend: type=identity
path=identity/
2022-10-26T11:37:02.592Z [INFO]  core: successfully mounted backend: type=cubbyhole
path=cubbyhole/
2022-10-26T11:37:02.594Z [INFO]  core: successfully enabled credential backend: type=token
path=token/ namespace="ID: root. Path: "
2022-10-26T11:37:02.595Z [INFO]  rollback: starting rollback manager
2022-10-26T11:37:02.596Z [INFO]  core: restoring leases
2022-10-26T11:37:02.598Z [INFO]  identity: entities restored
2022-10-26T11:37:02.598Z [INFO]  identity: groups restored
2022-10-26T11:37:02.598Z [INFO]  core: post-unseal setup complete
2022-10-26T11:37:02.598Z [INFO]  core: vault is unsealed
2022-10-26T11:37:02.599Z [INFO]  expiration: lease restore complete
2022-10-26T11:37:02.615Z [INFO]  core: successful mount: namespace="" path=secret/ type=kv
2022-10-26T11:37:02.622Z [INFO]  secrets.kv.kv_a96ececf: collecting keys to upgrade
2022-10-26T11:37:02.622Z [INFO]  secrets.kv.kv_a96ececf: done collecting keys: num_keys=1
2022-10-26T11:37:02.622Z [INFO]  secrets.kv.kv_a96ececf: upgrading keys finished
WARNING! dev mode is enabled! In this mode, Vault runs entirely in-memory
and starts unsealed with a single unseal key. The root token is already
authenticated to the CLI, so you can immediately begin using Vault.
You may need to set the following environment variables:
    $ export VAULT_ADDR='http://0.0.0.0:8200'
The unseal key and root token are displayed below in case you want to
seal/unseal the Vault or re-authenticate.
Unseal Key: OExWSfH+YevrNPAp1aD/ktbm0kK9xJ81RW9orm/CU1s=
Root Token: hvs.XDxJM72uAZ0uJEhOBHE01mX3
Development mode should NOT be used in production installations!
```



Успешно



Kubectl - используется для доступа и управления кластерами Kubernetes.
Pod - минимальная рабочая единица Kubernetes, логическая коллекция одного/нескольких контейнеров. У каждого Pod свой IP.
Pod может быть разбит на реплики, собранные в ReplicaSet, и контролируемые Deployment'ом
Сервисы - нужны для логической группировки Подов и контроля доступа к ним. Сервисы могут предоставлять отдельные Поды, ReplicaSet'ы, Deployment etc.
Node - рабочие узлы, на которые выполняются приложения

Команда expose создаёт сервер типа NodePort. NodePort открывает определенный порт (в данном случае - 8200), и любой трафик, перенаправляемый на этот порт, перенаправляется на сервис. Таким образом, мы можем получить доступ к Поду извне.

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

Чтобы соединить порт нашего компьютера и контейнера через сервис, воспользуемся командой `port-forward`

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