# Установка Vault в кластер K8s

##### Предварительные условия

1. развернут и настроен кластер Kubernetes (K8s) с поддержкой PV
2. на рабочей станции Администратора установлен kubectl и helm (v3)
3. и  есть  ли возможность выпустить сертификат с поддержкой нескольких SAN .

Ориентировочное затраченное время до 8 часов.

Проверьте доступные узлы:

# kubectl get nodes

NAME STATUS ROLES AGE VERSION

master Ready control-plane 2m54s v1.24.3+k0s

node1 Ready <none> 112s v1.24.3+k0s

node2 Ready <none> 113s v1.24.3+k0s

node3 Ready <none> 108s v1.24.3+k0s

## Шаг 1 – Создайте постоянное хранилище для Vault

Vault требует, чтобы был создан класс хранилища с настроенным постоянным томом. Начните с создания пространства имён и установите его по умолчанию:

kubectl create namespace vault

kubectl config set-context --current --namespace vault

### Пример создания класса хранения вручную (только для справки)

Если в вашем кластере уже настроен класс хранения по умолчанию, вы **можете пропустить этот шаг**.

Создайте класс хранилища;

vim storageClass.yml

Добавьте в файл приведенные ниже строки:

kind: StorageClass

apiVersion: storage.k8s.io/v1

metadata:

name: my-local-storage

provisioner: kubernetes.io/no-provisioner

volumeBindingMode: WaitForFirstConsumer

Создайте класс хранилища и установите его по умолчанию.

kubectl create -f storageClass.yml

kubectl patch storageclass my-local-storage -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'

Затем создайте PV, используя класс хранения:

vim vault-pv.yml

Добавьте строки ниже:

apiVersion: v1

kind: PersistentVolume

metadata:

name: my-local-pv

spec:

capacity:

storage: 10Gi

accessModes:

- ReadWriteOnce

persistentVolumeReclaimPolicy: Retain

storageClassName: my-local-storage

local:

path: /mnt/disk/vol1

nodeAffinity:

required:

nodeSelectorTerms:

- matchExpressions:

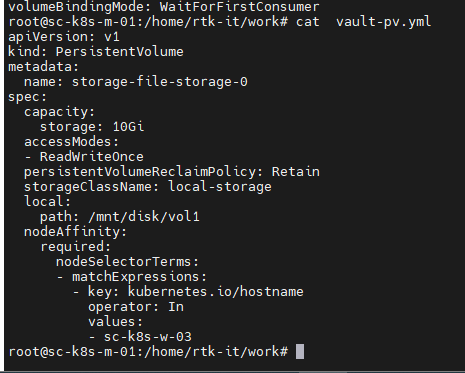
- key: kubernetes.io/hostname

operator: In

values:

- node1

Пример как это выглядит на стенде



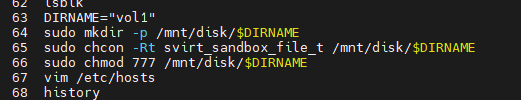
Если для узла задано соответствие **node1**, создайте путь к хранилищу на node1, как показано на рисунке. Данная  нода  должна  быть Worker Node

DIRNAME="vol1"

sudo mkdir -p /mnt/disk/$DIRNAME

sudo chcon -Rt svirt\_sandbox\_file\_t /mnt/disk/$DIRNAME

sudo chmod 777 /mnt/disk/$DIRNAME



Применить манифест:

kubectl create -f vault-pv.yml

## Шаг 2 – Запустите HashiCorp Vault в Kubernetes

В этом руководстве представлен самый простой способ развёртывания Vault в Kubernetes с помощью **Helm**. Сначала убедитесь, что Helm установлен в вашей системе.

curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3

chmod 700 get\_helm.sh

sudo ./get\_helm.sh

или

curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null

sudo apt-get install apt-transport-https --yes

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list

sudo apt-get update

sudo apt-get install helm

После установки проверьте, как показано на рисунке:

$ helm version

version.BuildInfo{Version:"v3.9.2", GitCommit:"1addefbfe665c350f4daf868a9adc5600cc064fd", GitTreeState:"clean", GoVersion:"go1.17.12"}

Git- клонирование в официальную таблицу управления хранилищем, поддерживаемую Hashicorp:

git clone https://github.com/hashicorp/vault-helm.git

cd vault-helm

В каталоге вы можете настроить **values.yaml** по своему усмотрению, а затем установить Vault, как показано ниже:

helm install vault . --namespace vault --values values.yaml или helm install vault . --namespace vault --values vault.yaml

Проверьте, запущены ли модули:

# kubectl get pods -n vault

NAME READY STATUS RESTARTS AGE

vault-0 0/1 Running 0 75s

vault-agent-injector-57f567889-js274 1/1 Running 0 80s

У вас также будет ПВХ, созданный с использованием класса хранения по умолчанию.

# kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

data-vault-0 Bound my-local-pv 10Gi RWO my-local-storage 9s

Инициализируйте кластер хранилища с помощью команды:

kubectl exec -ti --namespace vault vault-0 vault operator init

Команда должна возвращать **начальный токен** и **открытые ключи**, как показано на рисунке;

kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.

Unseal Key 1: vjwrDznfPk/7kHWY8L4OQL4PwXSuYFo3z45lt5SHolxj

Unseal Key 2: mnBo0TJGqDI1Qld18gM4kg6b58GYjLzKMAWaSX9uVwEg

Unseal Key 3: 6QWSei7R7re4sFlyz7os1TNpdxoJzpFOCvmhk09xIMWD

Unseal Key 4: iGZm2RiEQK3//RtUosUftb5dFU1R1YlqZmLQJJk7+I1I

Unseal Key 5: cnC9fyyxb4cBgKAKUbjTXT2R+y0CmyP/Ve7AlNvKZbut

Initial Root Token: hvs.QqliIhOLvhZq0nTCbKLAZXor

Vault initialized with 5 key shares and a key threshold of 3. Please securely

distribute the key shares printed above. When the Vault is re-sealed,

restarted, or stopped, you must supply at least 3 of these keys to unseal it

before it can start servicing requests.

Vault does not store the generated root key. Without at least 3 keys to

reconstruct the root key, Vault will remain permanently sealed!

It is possible to generate new unseal keys, provided you have a quorum of

existing unseal keys shares. See "vault operator rekey" for more information.

**Удалить VAULT если что то пошло не так**

helm uninstall vault --namespace vault

## Шаг 3 – Настройка службы Kubernetes HashiCorp Vault

После развёртывания HashiCorp Vault также будет развёрнут сервис ClusterIP, работающий на портах **8200** и **8201**.

# kubectl get svc

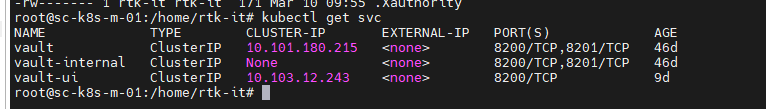
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

vault ClusterIP 10.102.36.175 <none> 8200/TCP,8201/TCP 21m

vault-agent-injector-svc ClusterIP 10.103.189.23 <none> 443/TCP 21m

vault-internal ClusterIP None <none> 8200/T

или



## Шаг 4 – Настройка службы Kubernetes HashiCorp Vault Secret

## Для создания секрета нам также потребуется ранее наши два сертификата. Сертификаты нужны что бы vault работал по https

## Создаем секрет

## kubectl create secret tls vault-tls-ingress -n vault --key t-rtk-it.ru.key --cert t-rtk-it.ru.crt

## Редактируем наш фаил инсталяции values.yaml

## Нас ршыеинтересуют вот эти строки

## 

## Поcле чего запускаем команду

## helm upgrade vault . --namespace vault --values vault.yaml

## И проверяем доступность и правильность выполнения команды и входа по HTTPS

## 

## Примечание:

## в папке с инструкциями лежит исходный код установки work.tar.gz

## Можно скопировать на ВМ и установить ППО VAULT