**Deploying a HA mode Vault with raft and use ingress load balancer**

**Deploying vault server:**

**Step-1:**

Install helm and add hashicorp repo .

Install helm:

apt-get update

apt-get updatecurl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 > get\_helm.sh

chmod 700 get\_helm.sh

./get\_helm.sh

Add repo:

helm repo add hashicorp https://helm.releases.hashicorp.com

"hashicorp" has been added to your repositories

**Step-2:**

To enable node port for ingress we need to the override the default configuration. This can be done by creating a **override-values.yaml** file as follows:

**override-values.yaml**

server:

ui:

enabled: true

ha:

enabled: true

replicas: 3

raft:

enabled: true

service:

type: NodePort

port: 8200

targetPort: 8200

protocol: TCP

**we will provide this file as argument during installation.**

**Install the hashicorp vault using the following command:**

helm install vault hashicorp/vault -f override-values.yml

**output:**

WARNING: Kubernetes configuration file is group-readable. This is insecure. Location: /root/.kube/config

WARNING: Kubernetes configuration file is world-readable. This is insecure. Location: /root/.kube/config

NAME: vault

LAST DEPLOYED: Mon Oct 12 04:40:53 2020

NAMESPACE: default

STATUS: deployed

REVISION: 1

TEST SUITE: None

NOTES:

Thank you for installing HashiCorp Vault!

Now that you have deployed Vault, you should look over the docs on using

Vault with Kubernetes available here:

https://www.vaultproject.io/docs/

Your release is named vault. To learn more about the release, try:

$ helm status vault

$ helm get vault

See the pods the are deployed using the following command,

kubectl get pods

**output:**

NAME READY STATUS RESTARTS AGE

vault-0 0/1 Running 0 8m12s

vault-1 0/1 Running 0 8m12s

vault-2 0/1 Running 0 8m12s

vault-agent-injector-bdbf7b844-6q7rs 1/1 Running 0 8m12s

to see the services created and the node ports:

kubectl get svc

**output:**

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

kubernetes ClusterIP 172.20.0.1 <none> 443/TCP 20d

vault NodePort 172.20.195.254 <none> 8200:30126/TCP,8201:31427/TCP 6m27s

vault-active NodePort 172.20.12.236 <none> 8200:32192/TCP,8201:31915/TCP 6m27s

vault-agent-injector-svc ClusterIP 172.20.79.220 <none> 443/TCP 6m27s

vault-internal ClusterIP None <none> 8200/TCP,8201/TCP 6m27s

vault-standby NodePort 172.20.17.203 <none> 8200:32041/TCP,8201:30895/TCP 6m27s

**Step-3: Initialize and unseal all the vaults join then to the cluster as follows**

**For Vault-0 :**

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

**output:**

Unseal Key 1: NpNQY6wy5AzUPSKiz2ygdb7YkAOnNS7i6Mo86FIaWhC/

Unseal Key 2: UIrps2q4ab6mnh282LkusBd7Ht1CAd980Pq+EehyUlCr

Unseal Key 3: /DFU5Rc4uyVH9C/HcLYy3HzTife45dtHJvolAqzbR3c8

Unseal Key 4: Idut7+kJiXEMgVgTMLkVGixJ0v8lUv/C7OBrxyfZavti

Unseal Key 5: HeB2SHWozRWD4URoUQqcJ/mvAnA8lMioprFeXkz6z+sj

Initial Root Token: s.9NuiX2lkeyVB8YhfC7ct3GDS

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 master key. Without at least 3 key to

reconstruct the master 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.

**Unseal vault-0:**

kubectl exec -ti vault-0 -- vault operator unseal

give this command and give the unseal key until it reaches it key threshold and until status for sealed becomes false as follows.

Key Value

--- -----

Seal Type shamir

Initialized true

Sealed false

Total Shares 5

Threshold 3

Version 1.5.2

Cluster Name vault-cluster-72ffbb12

Cluster ID bd393a8d-a285-4ba6-a926-baf11a5d9bd2

HA Enabled true

HA Cluster n/a

HA Mode standby

Active Node Address <none>

Raft Committed Index 24

Raft Applied Index 24

**For Vault-1 :**

First join the vault to the cluster:

kubectl exec -ti vault-1 -- vault operator raft join <http://vault-0.vault-internal:8200>

**output:**

Key Value

--- -----

Joined true

unseal vault as follows by using the same unseal keys of vault-0 until it reaches key threshold:

kubectl exec -ti vault-1 -- vault operator unseal

**For Vault-2 :**

First join the vault to the cluster:

kubectl exec -ti vault-2 -- vault operator raft join <http://vault-0.vault-internal:8200>

**output:**

Key Value

--- -----

Joined true

unseal vault as follows by using the same unseal keys of vault-0 until it reaches key threshold:

kubectl exec -ti vault-2 -- vault operator unseal

**Ingress deployment:**

Create a file vault-ingress.yaml :

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

name: "vault"

namespace: "default"

annotations:

kubernetes.io/ingress.class: alb

alb.ingress.kubernetes.io/scheme: internet-facing

alb.ingress.kubernetes.io/subnets: subnet-0fb67b0aecf8222a4,subnet-069743f33a9835f0b

labels:

app: vault-app

spec:

rules:

- http:

paths:

- path: /\*

backend:

serviceName: vault

servicePort: 8200

for vault path give only “ /\* ” vault will not support any other subpath.

Apply the yaml file to create ingress,

kubectl apply -f vault-ingress.yaml

**output:**

ingress.extensions/vault created

see your ingress using following command:

kubectl get ingress

**output:**

NAME HOSTS ADDRESS PORTS AGE

vault \* 1c05353a-default-vault-c083-1304294467.ap-southeast-1.elb.amazonaws.com 80 7s

To change the health check path sfrom AWS CONSOLE :

**goto loadbalancers 🡺 click on the name of the loadbalancer that is created for vault🡺goto listeners ,click on target 🡺 there goto health checks 🡺click on edit health check🡺change the path to “/ui/”**