**CONFIGURING VAULT IN HA MODE ON EKS CLUSTER USING CONSUL AS STORAGE BACKEND**

**Step -1 : Installation steps for 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

Now, run helm version to see if it is installed properly.

**helm version**

**output:** version.BuildInfo{Version:"v3.3.4", GitCommit:"a61ce5633af99708171414353ed49547cf05013d", GitTreeState:"clean", GoVersion:"go1.14.9"}

Run helm help to see the commands for helm

**helm help**

**Step-2 : HELM CHART**

The [Vault Helm chart](https://github.com/hashicorp/vault-helm) is the recommended way to install and configure Vault on Kubernetes. In addition to running Vault itself, the Helm chart is the primary method for installing and configuring Vault to integrate with other services such as Consul for High Availability (HA) deployments.

To use the Helm chart, add the Hashicorp helm repository and check that you have access to the chart.

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

"hashicorp" has been added to your repositories

**$** helm search repo hashicorp/vault

NAME CHART VERSION APP VERSION DESCRIPTION

hashicorp/vault 0.7.0 1.5.2 Official HashiCorp Vault Chart

you can see the existing vault version using the following command :

helm search repo hashicorp/vault –l

NAME CHART VERSION APP VERSION DESCRIPTION

hashicorp/vault 0.7.0 1.5.2 Official HashiCorp Vault Chart

hashicorp/vault 0.6.0 1.4.2 Official HashiCorp Vault Chart

hashicorp/vault 0.5.0 Install and configure Vault on Kubernetes.

hashicorp/vault 0.4.0 Install and configure Vault on Kubernetes.

**Step-up your consul backend using helm chart :**

helm install consul hashicorp/consul --set global.name=consul

**Step-up your vault server in HA mode :**

**Install vault :**

helm install vault hashicorp/vault \

--set "server.ha.enabled=true"

**Now check if your pods are running using the following command:**

kubectl get pods -l app.kubernetes.io/name=vault

NAME READY STATUS RESTARTS AGE

vault-0 0/1 Running 0 2d4h

vault-1 0/1 Running 0 2d4h

vault-2 0/1 Running 0 2d4h

**Unseal your vault servers :**

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

this command will give you the unseal keys and also the root token

**Output :**

Unseal Key 1: SjKAWyvemHccdms4autLszbood+r4m3IBaob67OKsrvf  
Unseal Key 2: OKJZ9fug/Hqdwz3zSc2wJj9RPonH9ljSzbBznicrqBG6  
Unseal Key 3: ouLRe0KwdclxJstKFyJHzqZE/jvAR50waCgH/Ch0ZUkM  
Unseal Key 4: lJXaHqPzWe8cyhGviq5xuSUQVfpIgCxw/KWF7Fwzs1kJ  
Unseal Key 5: e5lNWh8ZbvelrtpEGExISYaVLkthT5LkmNVj+EBrKmrg

Initial Root Token: s.uoAXo7RSsbmLBuUIYoe22zSZ

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 each vault using :**

kubectl exec -it vault-0 -- vault operator unseal SjKAWyvemHccdms4autLszbood+r4m3IBaob67OKsrvf

kubectl exec -it vault-0 -- vault operator unseal OKJZ9fug/Hqdwz3zSc2wJj9RPonH9ljSzbBznicrqBG6

kubectl exec -it vault-0 -- vault operator unseal ouLRe0KwdclxJstKFyJHzqZE/jvAR50waCgH/Ch0ZUkM

change the vault name and do the same for every vault

**Config files in each vault :**

Get inside each vault using the command : kubectl exec -it vault-0 bin/sh

**Go to path : cd tmp/ storageconfig.hcl**

**Storageconfig.hcl for vault-0 :**

disable\_mlock = true ui = true listener "tcp" { tls\_disable = 1 address = "[::]:8200" cluster\_address = "[::]:8201" }

storage "consul" { path = "vault" address = "127.0.0.1:8500" } service\_registration "kubernetes" {} # Example configuration for using auto-unseal, using Google Cloud KMS. The # GKMS keys must already exist, and the cluster must have a service account # that is authorized to access GCP KMS. #seal "gcpckms" { # project = "vault-helm-dev-246514" # region = "global" # key\_ring = "vault-helm-unseal-kr" # crypto\_key = "vault-helm-unseal-key" #}

**config file on vault-1 :**

disable\_mlock = true ui = true listener "tcp" { tls\_disable = 1 address = "[::]:8200" cluster\_address = "[::]:8201" }

storage "consul" { path = "vault" address = "10.44.0.223:8500" } service\_registration "kubernetes" {} # Example configuration for using auto-unseal, using Google Cloud KMS. The # GKMS keys must already exist, and the cluster must have a service account # that is authorized to access GCP KMS. #seal "gcpckms" { # project = "vault-helm-dev-246514" # region = "global" # key\_ring = "vault-helm-unseal-kr" # crypto\_key = "vault-helm-unseal-key" #}

**config file of vault-2 :**

disable\_mlock = true ui = true listener "tcp" { tls\_disable = 1 address = "[::]:8200" cluster\_address = "[::]:8201" }

storage "consul" { path = "vault" address = "10.44.0.233:8500" } service\_registration "kubernetes" {} # Example configuration for using auto-unseal, using Google Cloud KMS. The # GKMS keys must already exist, and the cluster must have a service account # that is authorized to access GCP KMS. #seal "gcpckms" { # project = "vault-helm-dev-246514" # region = "global" # key\_ring = "vault-helm-unseal-kr" # crypto\_key = "vault-helm-unseal-key" #}

**Try creating secrets in one vault to verify :**

Enable kv secrets engine :

vault secrets enable -version=1 kv

add secrets using following command :

vault kv put kv/my-secret my-value=s3cr3t

you will be able to see these secrets in any vault server in the cluster

vault kv get kv/my-secret

Key Value

--- -----

my-value s3cr3t

**LINKS FOLLOWED :**

**For vault installation :**

[**https://www.vaultproject.io/docs/platform/k8s/helm/run**](https://www.vaultproject.io/docs/platform/k8s/helm/run)

**For consul installation :**

[**https://github.com/hashicorp/consul-helm**](https://github.com/hashicorp/consul-helm)