**HashiCorp Vault Prod Setup in EKS**

**✅ STEP 1: Create Namespace for Vault**

kubectl create namespace vault

**🧠 Why?**

To separate Vault’s workloads from the rest of your cluster for organization and RBAC control.

**✅ STEP 2: Add Helm Repo and Install Vault with Raft HA**

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

helm repo update

**Create vault-values.yaml for Production:**

server:

ha:

enabled: true

replicas: 3

raft:

enabled: true

config: |

ui = true

listener "tcp" {

address = "0.0.0.0:8200"

cluster\_address = "0.0.0.0:8201"

tls\_disable = 1

}

storage "raft" {

path = "/vault/data"

retry\_join {

leader\_api\_addr = "http://vault-0.vault-internal:8200"

}

retry\_join {

leader\_api\_addr = "http://vault-1.vault-internal:8200"

}

retry\_join {

leader\_api\_addr = "http://vault-2.vault-internal:8200"

}

}

service\_registration "kubernetes" {}

dataStorage:

enabled: true

size: 10Gi

storageClass: "ebs-sc" # or "ebs-sc" if you've defined your own

extraEnvironmentVars:

VAULT\_LOG\_LEVEL: "debug"

injector:

enabled: true

ui:

enabled: true

**Install Vault with this config:**

helm install vault hashicorp/vault -n vault -f vault-values.yaml

**✅ STEP 3: Expose Vault Using LoadBalancer**

Create a service vault-service.yaml:

apiVersion: v1

kind: Service

metadata:

name: vault

namespace: vault

spec:

type: LoadBalancer

ports:

- port: 8200

targetPort: 8200

selector:

app.kubernetes.io/name: vault

Apply it:

kubectl apply -f vault-service.yaml

**✅ STEP 4: Initialize Vault (Run Once)**

kubectl exec -n vault -it vault-0 -- vault operator init

☝️ Copy and save:

* **5 unseal keys**
* **1 initial root token**

**✅ STEP 5: Unseal Vault on All Pods**

Use any 3 keys on each Vault pod:

kubectl exec -n vault -it vault-0 -- vault operator unseal <key>

kubectl exec -n vault -it vault-1 -- vault operator unseal <key>

kubectl exec -n vault -it vault-2 -- vault operator unseal <key>

**✅ STEP 6: Login to Vault**

kubectl exec -n vault -it vault-0 -- vault login <root-token>

**✅ 1. Enable Kubernetes Auth on Vault**

kubectl exec -n vault -it vault-0 -- vault auth enable kubernetes

**✅ 2. Create the Vault Policy**

**Create a policy file locally:**

cat <<EOF > noteapp-policy.hcl

path "secret/data/noteapp" {

capabilities = ["read"]

}

path "secret/metadata/noteapp" {

capabilities = ["list"]

}

EOF

**Then upload and apply it:**

kubectl cp noteapp-policy.hcl vault/vault-0:/tmp/noteapp-policy.hcl

kubectl exec -n vault -it vault-0 -- vault policy write noteapp /tmp/noteapp-policy.hcl

**✅ 3. Create a Service Account in Your App Namespace**

kubectl create namespace webapps

kubectl create serviceaccount noteapp-sa -n webapps

**✅ 4. Create a Vault Role for the Service Account**

kubectl exec -n vault -it vault-0 -- vault write auth/kubernetes/role/noteapp \

bound\_service\_account\_names=noteapp-sa \

bound\_service\_account\_namespaces=webapps \

policies=noteapp \

ttl=24h

**✅ 5. Configure Kubernetes Auth Backend**

**Extract values from cluster:**

SERVICE\_ACCOUNT\_NAME=noteapp-sa

NAMESPACE=webapps

SECRET\_NAME=$(kubectl get sa $SERVICE\_ACCOUNT\_NAME -n $NAMESPACE -o jsonpath="{.secrets[0].name}")

TOKEN\_REVIEW\_JWT=$(kubectl get secret $SECRET\_NAME -n $NAMESPACE -o jsonpath="{.data.token}" | base64 --decode)

KUBE\_CA\_CERT=$(kubectl get secret $SECRET\_NAME -n $NAMESPACE -o jsonpath="{.data['ca.crt']}" | base64 --decode)

KUBE\_HOST=$(kubectl config view --raw -o=jsonpath='{.clusters[0].cluster.server}')

**Now apply config:**

kubectl exec -n vault -it vault-0 -- vault write auth/kubernetes/config \

token\_reviewer\_jwt="$TOKEN\_REVIEW\_JWT" \

kubernetes\_host="$KUBE\_HOST" \

kubernetes\_ca\_cert="$KUBE\_CA\_CERT"

# Enable KV V2 Engine

kubectl exec -n vault -it vault-0 -- vault secrets enable -path=secret -version=2 kv

**✅ 6. Store the MongoDB Connection String in Vault**

**Make sure you match casing with your app (your file reads from /vault/secrets/mongodb\_\_connectionstring):**

kubectl exec -n vault -it vault-0 -- vault kv put secret/noteapp mongodb\_\_connectionstring="mongodb://mongo-0.mongo,mongo-1.mongo,mongo-2.mongo:27017/NoteDb?replicaSet=rs0"