**SECRETS USING AWS SECRETS ENGINE**

**Step – 1 :**

Install helm – the helm version which we installed in our vm is **v3.3.4**

**Commands to install :**

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

chmod 700 get\_helm.sh

helm version

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

**Step-2 :**

To install hashicorp vault

**Commands :**

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

There are different modes to install vault :

**Stand-alone mode :** This installs a single Vault server with a file storage backend.

helm install vault hashicorp/vault

**Dev mode :** This installs a single Vault server with a memory storage backend.

This is ideal for learning and demonstration environments but NOT recommended for a production environment.

helm install vault hashicorp/vault \

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

**HA mode :** This installs three Vault servers with an existing Consul storage backend. It is suggested that Consul is installed via the [Consul Helm chart](https://github.com/hashicorp/consul-helm).

helm install vault hashicorp/vault \

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

**External mode :** This installs no vault server and relies on network addressable vault server to exist.

helm install vault hashicorp/vault \

--set "injector.externalVaultAddr=http://external-vault:8200"

Here we tried both dev mode and standalone modes.

**Step – 3 :** kubectl get pods

vault-0 1/1 Running 0 18h

vault-agent-injector-bdbf7b844-r46lj 1/1 Running 0 18h

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

vault-0 1/1 Running 0 18h

get inside your vault to see te secrets and status of your vault

kubectl exec -ti vault-0 /bin/sh

if you now run your vault status - vault status

Key Value

--- -----

Seal Type shamir

Initialized true

Sealed false

Total Shares 1

Threshold 1

Version 1.5.2

Cluster Name vault-cluster-47c242df

Cluster ID 4d70f23b-202f-456c-ccba-b8e4bc7766c4

HA Enabled false

Currently it is running in **dev** mode so **sealed** is **false .** you will be able see the secrets without unsealing it. If you run the same thing in **stand-alone** mode then **sealed** will be **true**. You will be able to see secrets. you need to unseal before performing any action.

**Step – 4 :**

Enabling AWS secret engine ,

vault secrets enable -path=aws aws

vault write aws/config/root \

access\_key=ACCESS\_KEY\_ID \

secret\_key=SECRET\_ACCESS\_KEY

vault write aws/roles/my-role \

credential\_type=iam\_user \

policy\_document=-<<EOF

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "ec2:\*",

"Resource": "\*"

}

]

}

EOF

vault read aws/creds/my-role

output :

Key Value

--- -----

lease\_id aws/creds/my-role/e75QUrah9HuVm3c5KUBJaKg9

lease\_duration 768h

lease\_renewable true

access\_key AKIA4OJTSXOYK5G5ODEH

secret\_key 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

security\_token <nil>

**Step – 5 :**

**To change the lease time -**

vault write aws/config/lease lease=30m lease\_max=1h

vault read aws/creds/my-role

**output:**

Key Value

--- -----

lease\_id aws/creds/my-role/Kgz43H9tBr1hrUX7tRRcQXGz

lease\_duration 30m

lease\_renewable true

access\_key AKIA4OJTSXOYOZXAH3Q4

secret\_key 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

security\_token <nil>

**Step – 6 :**

vault lease revoke -prefix aws/creds/my-role

this command will delete all leases in that prefix .

To delete a particular lease

vault lease revoke aws/creds/my-role/eHKMYIA7HbgRUC0uLhIsqwdC

Links :

<https://www.vaultproject.io/docs/secrets/aws>

<https://www.cloudops.com/blog/how-to-deploy-a-development-vault-server-to-kubernetes-using-helm/>

<https://www.hashicorp.com/blog/injecting-vault-secrets-into-kubernetes-pods-via-a-sidecar>

<https://www.vaultproject.io/docs/platform/k8s/helm/run>

**SECRETS USING DATABASE SECRETS ENGINE**

**Step-1 :**

Enable database secrets engine

vault secrets enable database

**Step-2 :**

Database secrets engine supports many databases through a plugin interface. For PostgreSQL the following plugin should be used - postgresql-database-plugin

vault write database/config/postgresql \

plugin\_name=postgresql-database-plugin \

connection\_url="postgresql://{{username}}:{{password}}@localhost:5432/postgres?sslmode=disable" \

allowed\_roles=readonly \

username="root" \

password="rootpassword"

**Step-3 :**

Create a role

tee readonly.sql <<EOF

CREATE ROLE "{{name}}" WITH LOGIN PASSWORD '{{password}}' VALID UNTIL '{{expiration}}' INHERIT;

GRANT ro TO "{{name}}";

EOF

The SQL contains the templatized fields {{name}}, {{password}}, and {{expiration}}. These values are provided by Vault when the credentials are created. This creates a new role and then grants that role the permissions defined in the Postgres role named ro. This Postgres role was created when postgres was started.

Create the role named readonly that creates credentials with the readonly.sql.

vault write database/roles/readonly \

db\_name=postgresql \

creation\_statements=@readonly.sql \

default\_ttl=1h \

max\_ttl=24h

**Step-4: Request postgres credentials**

vault read database/creds/readonly

Key Value

--- -----

lease\_id database/creds/readonly/fyF5xDomnKeCHNZNQgStwBKD

lease\_duration 1h

lease\_renewable true

password A1a-ckirtymYaXACpIHn

username v-token-readonly-6iRIcGv8tLpu816oblPY-1556567086

Now, connect to your postgres sql and see if a new user is created .

docker exec -it postgres psql

SELECT usename, valuntil FROM pg\_user;

--------------------------------------------------+------------------------

root |

v-token-readonly-47vOtpF7pZq79Xajx7yq-1556567237 | 2020-09-14 17:36:48+00

ro |

(3 rows)

**Links :**

<https://www.vaultproject.io/docs/secrets/databases/mysql-maria>

<https://learn.hashicorp.com/tutorials/vault/database-secrets>

**HA MODE ON EKS CLUSTER :**

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

955 helm search repo hashicorp/consul

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

957 kubectl get pods

958 helm install vault hashicorp/vault --set "server.ha.enabled=true"

959 kubectl get pods

**{**

**"accessKey": "AKIATSRJXG25ROJYWA57",**

**"secretKey": "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",**

**"leaseId": "aws/creds/ec2-role/32TGbuwhmttvl5MILmgYQ1m2"**

**}**