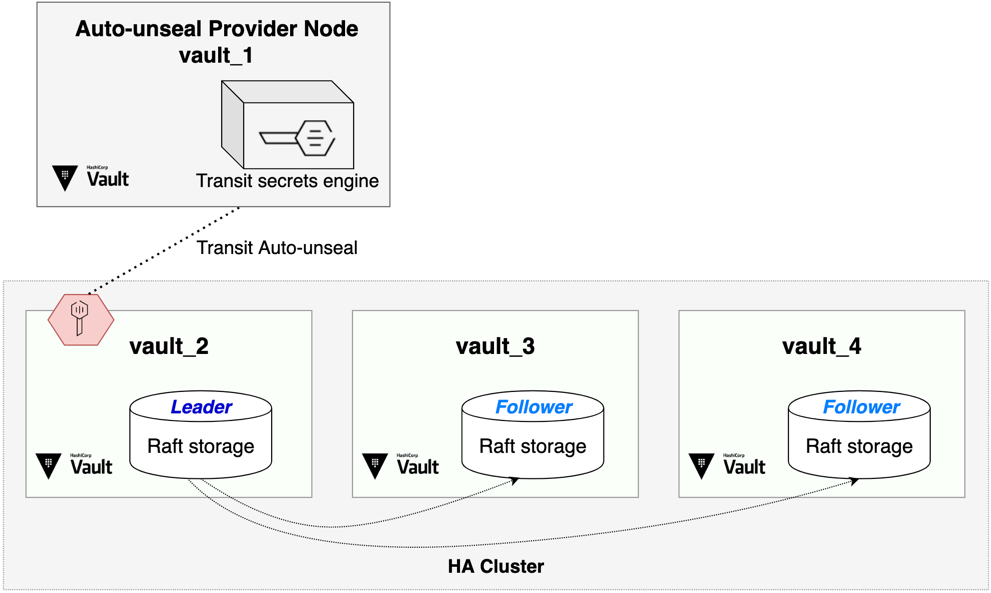
**VAULT HA MODE CONFIGURATION WITH RAFT BACKEND**



**vault\_1** is initialized and unsealed. The root token creates a transit key that enables the other Vaults auto-unseal. This Vault does not join the cluster.

**vault\_2** is initialized and unsealed. This Vault starts as the cluster leader. An example K/V-V2 secret is created.

**vault\_3** is only started. You will join it to the cluster.

**vault\_4** is only started. You will join it to the cluster.

**Prerequisites :**

* Install terraform

**Vault transit** : **auto-unseal a Vault with Transit secrets engine**

It is the encryption service provider, and its transit key protects the **Vault 2** server's master key, which is a part of HA cluster.

**PROCEDURE :**

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

cd vault-guides/operations/raft-storage/aws

**$** export AWS\_ACCESS\_KEY\_ID = "<YOUR\_AWS\_ACCESS\_KEY\_ID>"

**$** export AWS\_SECRET\_ACCESS\_KEY = "<YOUR\_AWS\_SECRET\_ACCESS\_KEY>"

cp terraform.tfvars.example terraform.tfvars

Edit the tfvars file : give the region, availability zone ,key-pair in that region.

Initialize the terraform : terraform init

Now run the terraform : terraform apply

**OUTPUT :**

vault\_1 (54.145.131.93) | internal: (10.0.101.21)  
    - Initialized and unsealed.  
    - The root token creates a transit key that enables the other Vaults to auto-unseal.  
    - Does not join the High-Availability (HA) cluster.

  vault\_2 (18.205.103.30) | internal: (10.0.101.22)  
    - Initialized and unsealed.  
    - The root token and recovery key is stored in /tmp/key.json.  
    - K/V-V2 secret engine enabled and secret stored.  
    - Leader of HA cluster

    $ ssh -l ubuntu 18.205.103.30 -i vault-eks.pem

    # Root token:  
    $ ssh -l ubuntu 18.205.103.30 -i vault-eks.pem "cat ~/root\_token"  
    # Recovery key:  
    $ ssh -l ubuntu 18.205.103.30 -i vault-eks.pem "cat ~/recovery\_key"

  vault\_3 (34.238.120.59) | internal: (10.0.101.23)  
    - Started  
    - You will join it to cluster started by vault\_2

    $ ssh -l ubuntu 34.238.120.59 -i vault-eks.pem

  vault\_4 (3.91.145.125) | internal: (10.0.101.24)  
    - Started  
    - You will join it to cluster started by vault\_2

    $ ssh -l ubuntu 3.91.145.125 -i vault-eks.pem

root@ip-172-31-19-186:~/vault-guides/operations/raft-storage/aws# vault status

After terraform apply first go inside the vault-2 and then export token using following command :

export VAULT\_TOKEN=$(cat ~/root\_token)

Currently, **vault\_2** is initialized, unsealed, and has HA enabled. It is the only node in a cluster. The remaining nodes, **vault\_3** and **vault\_4**, have not joined its cluster.

ssh -l ubuntu 18.205.103.30 -i vault-eks.pem

sudo cat /etc/vault.d/vault.hcl

storage "raft" {

path = "/vault/vault\_2"

node\_id = "vault\_2"

}

listener "tcp" {

address = "0.0.0.0:8200"

cluster\_address = "0.0.0.0:8201"

tls\_disable = true

}

seal "transit" {

address = "http://10.0.101.21:8200"

token = "root"

disable\_renewal = "false"

// Key configuration

key\_name = "unseal\_key"

mount\_path = "transit/"

}

api\_addr = "http://18.205.103.30:8200"

cluster\_addr = "http://10.0.101.22:8201"

disable\_mlock = true

ui=true

Configure the vault CLI to use the root token for requests.

export VAULT\_TOKEN=$(cat ~/root\_token)

vault operator raft list-peers

using above command you can view the raft configuration.

**Add nodes to the cluster :**

**Vault-3 :**

ssh -l ubuntu 34.238.120.59 -i vault-eks.pem

**join vault-3 to vault-2 :**

**configuration file :**

storage "raft" {  
  path    = "/vault/vault\_3"  
  node\_id = "vault\_3"  
}

listener "tcp" {  
  address     = "0.0.0.0:8200"  
  cluster\_address     = "0.0.0.0:8201"  
  tls\_disable = true  
}

seal "transit" {  
  address            = "http://10.0.101.21:8200"  
  token              = "root"  
  disable\_renewal    = "false"

  // Key configuration  
  key\_name           = "unseal\_key"  
  mount\_path         = "transit/"  
}

api\_addr = "http://34.238.120.59:8200"  
cluster\_addr = "http://10.0.101.23:8201"  
disable\_mlock = true  
ui=true

vault operator raft join http: 18.205.103.30 //:8200

Key Value

--- -----

Joined true

Configure the vault CLI to use **vault\_2** root token for requests which is stored in the ~/root\_token file on the **vault\_2** host.

export VAULT\_TOKEN="s.qr20f2k5rtQ08UjmdaxkdByV"

Examine the raft peer set using ,

vault operator raft list-peers

at this point you will be able to see vault-2 and vault-3 in the list.

vault kv get kv/apikey

using the above command you will be able to see the secrets in vault-2 from vault-3

**Join vault-4 :**

ssh -l ubuntu 3.91.145.125 -i vault-eks.pem

sudo vi /etc/vault.d/vault.hcl

sudo systemctl stop vault

As we know vault-2 and vault-3 ips we can add them to retry-join .Then start the vault.

storage "raft" {  
  path    = "/vault/vault\_4"  
  node\_id = "vault\_4"  
retry\_join {  
    leader\_api\_addr = "http://18.205.103.30:8200"  
  }  
  retry\_join {  
    leader\_api\_addr = "http://34.238.120.59:8200"  
  }  
}

listener "tcp" {  
  address     = "0.0.0.0:8200"  
  cluster\_address     = "0.0.0.0:8201"  
  tls\_disable = true  
}

seal "transit" {  
  address            = "http://10.0.101.21:8200"  
  token              = "root"  
  disable\_renewal    = "false"

  // Key configuration  
  key\_name           = "unseal\_key"  
  mount\_path         = "transit/"  
}

api\_addr = "http://3.91.145.125:8200"  
cluster\_addr = "http://10.0.101.24:8201"  
disable\_mlock = true  
ui=true

sudo systemctl start vault

**configure your vault :**

export VAULT\_TOKEN="s.qr20f2k5rtQ08UjmdaxkdByV"

**List you vault peers :**

vault operator raft list-peers  
Node       Address             State       Voter  
----       -------             -----       -----  
vault\_2    10.0.101.22:8201    leader      true  
vault\_3    10.0.101.23:8201    follower    true  
vault\_4    10.0.101.24:8201    follower    true

**Link:**

[**https://learn.hashicorp.com/tutorials/vault/autounseal-transit**](https://learn.hashicorp.com/tutorials/vault/autounseal-transit)

[**https://learn.hashicorp.com/tutorials/vault/raft-storage-aws**](https://learn.hashicorp.com/tutorials/vault/raft-storage-aws)

**our vault set in mindtree aws account** :

vault\_1 (18.206.57.48) | internal: (10.0.101.21)

- Initialized and unsealed.

- The root token creates a transit key that enables the other Vaults to auto-unseal.

- Does not join the High-Availability (HA) cluster.

vault\_2 (52.90.88.15) | internal: (10.0.101.22)

- Initialized and unsealed.

- The root token and recovery key is stored in /tmp/key.json.

- K/V-V2 secret engine enabled and secret stored.

- Leader of HA cluster

$ ssh -l ubuntu 52.90.88.15 -i vault-eks.pem

# Root token:

$ ssh -l ubuntu 52.90.88.15 -i vault-eks.pem "cat ~/root\_token"

# Recovery key:

$ ssh -l ubuntu 52.90.88.15 -i vault-eks.pem "cat ~/recovery\_key"

vault\_3 (3.93.80.239) | internal: (10.0.101.23)

- Started

- You will join it to cluster started by vault\_2

$ ssh -l ubuntu 3.93.80.239 -i vault-eks.pem

vault\_4 (54.85.182.3) | internal: (10.0.101.24)

- Started

- You will join it to cluster started by vault\_2

$ ssh -l ubuntu 54.85.182.3 -i vault-eks.pem

**Root token for the vault created in aws :**

export VAULT\_TOKEN="s.IZeqoi10Pez65z1hEqHvbiWb"