HashiCorp Vault Cluster Unseal

These steps are for a new vault cluster that has been configured and started up, but is in an unsealed state.

For an existing cluster with unsealed nodes, see this page: Handling Vault Node Restart

Leader Initialization

The following will initialize a new vault cluster and return a set of unseal keys and a root token.

> NOTE: The ca.crt file is privileged, You will need to run these commands as the root user. Run the following to switch to the root user:

```
sudo -i
```

Go to the first node, and do these (as root):

```
# From an admin shell that can reach the VLAN:
export VAULT_ADDR="https://vault0204.ogsofttech.lan:8200"
export VAULT_CACERT="/opt/vault/tls/ca.crt"   # path on your admin box

# Initialize the cluster (choose your own shares/threshold)
vault operator init -key-shares=5 -key-threshold=3
```

> NOTE: Use the fully-qualified hostname above, as it appears in the node's cert.

Once executed, the vault node will reply with 5 unseal keys and an initial root token.

> Distribute each of these unseal keys to trusted admins, to store in offline password storage.

> NOTE: Three (3) unseal keys are required to unseal the vault.

>

> Use the initial root token to setup policies and auth.
> Then, retire it.

Unseal the Leader

With the unseal keys from the initialized node (received above), we need to unseal its vault.

> NOTE: We do this, while still as root, and on the same host that we got the keys from.

Now, unseal each node, by calling this command once each, for three of the five unseal keys:

> NOTE: It will prompt you for the unseal key, each time you run it.

```
vault operator unseal
```

Initial Root Login

With the vault unsealed, we need to perform an initial login as root:

```
# Log in with the root token for initial setup tasks
vault login <root_token>
```

Once logged in, you can check the vault status with this:

```
vault status
```

If successful, you should see `Initialized: true`, `Sealed: false`, `HA Enabled: true`, and this node as leader.

The first node is online, and the cluster is up... sort of.

Each cluster member has auto-discovered a leader and established a RAFT quorum.

But, the other nodes are still not unsealed (since we did not configured auto-unseal).

Unseal Other Nodes

Similar to what you did, to unseal the first node, we will do the same to each member, below.

Switch to root on each node with:

```
sudo -i
```

Set exports for each node:

> NOTE: Make sure that the vault_addr variable is pointing to the local node being unsealed, here.

```
export VAULT_ADDR="https://vault0205.ogsofttech.lan:8200";
export VAULT_CACERT="/opt/vault/tls/ca.crt"
```

> NOTE: Use the fully-qualified hostname above, as it appears in the node's cert.

Now, unseal each node, by calling this command once each, for three of the five unseal keys:

> NOTE: It will prompt you for the unseal key, each time you run it.

```
vault operator unseal
```

Check Status

Once you have initialized the leader node, and unsealed all nodes, we need to confirm that the cluster is good.

Run this:

> NOTE: The vault_addr should be pointing to the leader, here.

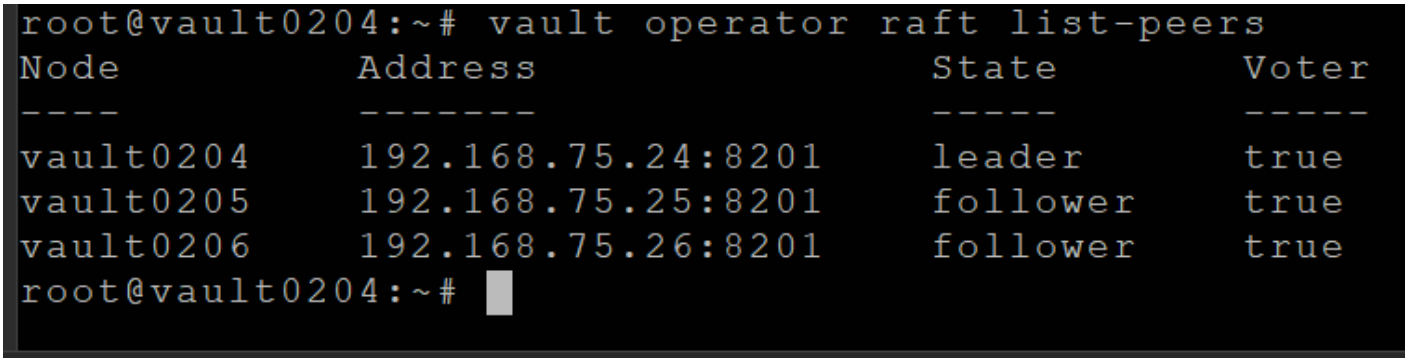
```
# Set this if you are coming back to this page, and the environment value is not set...
export VAULT_ADDR="https://vault0204.ogsofttech.lan:8200";
# Run this to check status...
vault status
```

Confirm RAFT peers with this:

```
vault operator raft list-peers
```

> NOTE: The above may only work on the current leader, because https.
> We need to work through why this is, and solve it, so it can be run on any node.

When run, you will see something like this:



If healthy, you will see one node as leader, and the others as voting followers.

> NOTE: Make sure each node you configured, is present.

There is a health check that can be performed, by calling this:

```
curl -s -H "X-Vault-Token: $VAULT_TOKEN" "$VAULT_ADDR/v1/sys/ha-status" | jq .
```

> NOTE: Be sure that the VAULT_TOKEN and VAULT_ADDR environment variables are set.
> Or, you can hardcode them with a minimal privilege user account.

When run, you will get a JSON list of nodes and their status.

Revision #20
Created 4 September 2025 00:49:31 by glwhite
Updated 4 September 2025 05:05:28 by glwhite