1. **First of all, let's start with installing vault on all 3 nodes.**

Vault1:
Prerequisites(necessary packages, directories, user configs):

```
root@vault1:~# apt install -y unzip curl firewalld
root@vault1:~# systemctl enable --now firewalld
```

```
root@vault1:~# useradd --system --home /etc/vault.d --shell /usr/sbin/nologin vault
root@vault1:~# mkdir -p /opt/vault/bin /etc/vault.d /var/lib/vault /var/log/vault
root@vault1:~# chown -R vault:vault /etc/vault.d /var/lib/vault /var/log/vault /opt/vault
root@vault1:~# chmod 750 /etc/vault.d /var/lib/vault
```

Installation:

```
root@vault1:~# cd /tmp/
root@vault1:/tmp# curl -LO https://releases.hashicorp.com/vault/1.20.2/vault_1.20.2_linux_amd64.zip
root@vault1:/tmp# unzip vault_1.20.2_linux_amd64.zip
Archive:  vault_1.20.2_linux_amd64.zip
  inflating: vault
  inflating: LICENSE.txt
root@vault1:/tmp# install -o root -g root -m 0755 vault /usr/local/bin/
root@vault1:/tmp# vault --version
Vault v1.20.2 (824d12909d5b596ddd3f34d9c8f169b4f9701a0c), built 2025-08-05T19:05:39Z
```

Vault2:
Prerequisites(necessary packages, directories, user configs):

```
root@vault2:~# apt install -y unzip curl firewalld
root@vault2:~# systemctl enable --now firewalld
```

```
root@vault2:~# useradd --system --home /etc/vault.d --shell /usr/sbin/nologin vault
root@vault2:~# mkdir -p /opt/vault/bin /etc/vault.d /var/lib/vault /var/log/vault
root@vault2:~# chown -R vault:vault /etc/vault.d /var/lib/vault /var/log/vault /opt/vault
root@vault2:~# chmod 750 /etc/vault.d /var/lib/vault
```

Installation:

```
root@vault2:~# cd /tmp/
root@vault2:/tmp# curl -LO https://releases.hashicorp.com/vault/1.20.2/vault_1.20.2_linux_amd64.zip
root@vault2:/tmp#  unzip vault_1.20.2_linux_amd64.zip
Archive:  vault_1.20.2_linux_amd64.zip
  inflating: vault
  inflating: LICENSE.txt
root@vault2:/tmp# install -o root -g root -m 0755 vault /usr/local/bin/
root@vault2:/tmp# vault --version
Vault v1.20.2 (824d12909d5b596ddd3f34d9c8f169b4f9701a0c), built 2025-08-05T19:05:39Z
```

Vault3:
Prerequisites(necessary packages, directories, user configs):

```
root@vault3:~# apt install -y unzip curl firewalld
root@vault3:~# systemctl enable --now firewalld
```

```
root@vault3:~# useradd --system --home /etc/vault.d --shell /usr/sbin/nologin vault
root@vault3:~# mkdir -p /opt/vault/bin /etc/vault.d /var/lib/vault /var/log/vault
root@vault3:~# chown -R vault:vault /etc/vault.d /var/lib/vault /var/log/vault /opt/vault
root@vault3:~# chmod 750 /etc/vault.d /var/lib/vault
```

Installation:

```
root@vault3:~# cd /tmp/
root@vault3:/tmp# curl -LO https://releases.hashicorp.com/vault/1.20.2/vault_1.20.2_linux_amd64.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  165M  100  165M    0     0  11.6M      0  0:00:14  0:00:14 --:--:-- 11.5M
root@vault3:/tmp# unzip vault_1.20.2_linux_amd64.zip
Archive:  vault_1.20.2_linux_amd64.zip
  inflating: vault
  inflating: LICENSE.txt
root@vault3:/tmp# install -o root -g root -m 0755 vault /usr/local/bin/
root@vault3:/tmp# vault --version
Vault v1.20.2 (824d12909d5b596ddd3f34d9c8f169b4f9701a0c), built 2025-08-05T19:05:39Z
```

2. Next step is, TLS setup.

Create TLS directory for each node:

Vault1:

```
root@vault1:~# mkdir -p /etc/vault.d/tls
root@vault1:~# cd /etc/vault.d/tls
```

Vault2:

```
root@vault2:~# mkdir -p /etc/vault.d/tls
root@vault2:~# cd /etc/vault.d/tls
```

Vault3:

```
root@vault3:~# mkdir -p /etc/vault.d/tls
root@vault3:~# cd /etc/vault.d/tls
```

Generate a self-signed CA on vault1:

```
root@vault1:/etc/vault.d/tls# openssl genrsa -out ca.key 4096
```

```
root@vault1:/etc/vault.d/tls# openssl req -x509 -new -nodes -key ca.key -subj "/CN=Vault-CA" -days 3650 -out ca.crt
```

Create a node certificate for each node.

Vault1 certificate:

```
root@vault1:/etc/vault.d/tls# vi csr.conf
```

```
[req]
default_bits = 2048
distinguished_name = req_distinguished_name
prompt = no

[req_distinguished_name]
CN = vault1

[ v3_ext ]
subjectAltName = @alt_names

[alt_names]
IP.1 = 192.168.100.201
DNS.1 = vault1
IP.2 = 192.168.100.210
DNS.2 = vault.local
```

```
root@vault1:/etc/vault.d/tls# openssl genrsa -out vault.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................+++++
.........................+++++
e is 65537 (0x010001)
root@vault1:/etc/vault.d/tls# openssl req -new -key vault.key -out vault.csr -config csr.conf
root@vault1:/etc/vault.d/tls# openssl x509 -req -in vault.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
>   -out vault.crt -days 825 -extensions v3_ext -extfile csr.conf
Signature ok
subject=CN = vault1
Getting CA Private Key
root@vault1:/etc/vault.d/tls# sudo chown -R vault:vault /etc/vault.d/tls
root@vault1:/etc/vault.d/tls# sudo chmod 640 /etc/vault.d/tls/vault.key
```

Copy ca.crt and ca.key to other machines to create and sign their certificates. NOT SAFE FOR REAL ENVIRONMENT BUT OK FOR LAB TEST!!!

```
root@vault1:/etc/vault.d/tls# scp ca.crt ca.key root@192.168.100.202:/etc/vault.d/tls
The authenticity of host '192.168.100.202 (192.168.100.202)' can't be established.
ECDSA key fingerprint is SHA256:jKHwqGI5dBTk+E3HcEDDGXZDfP4K6PzFzoSNDvoktxQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.202' (ECDSA) to the list of known hosts.
root@192.168.100.202's password:
ca.crt                                                                    100% 1805     5.1MB/s   00:00
ca.key                                                                    100% 3243    10.5MB/s   00:00
root@vault1:/etc/vault.d/tls# scp ca.crt ca.key root@192.168.100.203:/etc/vault.d/tls
The authenticity of host '192.168.100.203 (192.168.100.203)' can't be established.
ECDSA key fingerprint is SHA256:6Mc0/lPY1/v+eomDObC+s8z9ZO0Ehl1iszPVP9Gunis.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.203' (ECDSA) to the list of known hosts.
root@192.168.100.203's password:
ca.crt                                                                    100% 1805     4.1MB/s   00:00
ca.key                                                                    100% 3243     7.9MB/s   00:00
```

Vaul2:

```
root@vault2:/etc/vault.d/tls# vi csr.conf
```

```
[req]
default_bits = 2048
distinguished_name = req_distinguished_name
prompt = no

[req_distinguished_name]
CN = vault2

[ v3_ext ]
subjectAltName = @alt_names

[alt_names]
IP.1 = 192.168.100.202
DNS.1 = vault2
IP.2 = 192.168.100.210
DNS.2 = vault.local
```

```
root@vault2:/etc/vault.d/tls# vi csr.conf
root@vault2:/etc/vault.d/tls#
root@vault2:/etc/vault.d/tls# openssl genrsa -out vault.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................+++++
.....................................................................................................+++++
e is 65537 (0x010001)
root@vault2:/etc/vault.d/tls# openssl req -new -key vault.key -out vault.csr -config csr.conf
root@vault2:/etc/vault.d/tls# openssl x509 -req -in vault.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
>   -out vault.crt -days 825 -extensions v3_ext -extfile csr.conf
Signature ok
subject=CN = vault2
Getting CA Private Key
root@vault2:/etc/vault.d/tls# sudo chown -R vault:vault /etc/vault.d/tls
root@vault2:/etc/vault.d/tls# sudo chmod 640 /etc/vault.d/tls/vault.key
```

Vault3:

```
root@vault3:/etc/vault.d/tls# vi csr.conf
```

```
[req]
default_bits = 2048
distinguished_name = req_distinguished_name
prompt = no

[req_distinguished_name]
CN = vault3

[ v3_ext ]
subjectAltName = @alt_names

[alt_names]
IP.1 = 192.168.100.203
DNS.1 = vault3
IP.2 = 192.168.100.210
DNS.2 = vault.local
```

```
root@vault3:/etc/vault.d/tls# openssl genrsa -out vault.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................+++++
.....................................................................+++++
e is 65537 (0x010001)
root@vault3:/etc/vault.d/tls# openssl req -new -key vault.key -out vault.csr -config csr.conf
root@vault3:/etc/vault.d/tls# openssl x509 -req -in vault.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
>   -out vault.crt -days 825 -extensions v3_ext -extfile csr.conf
Signature ok
subject=CN = vault3
Getting CA Private Key
root@vault3:/etc/vault.d/tls# sudo chown -R vault:vault /etc/vault.d/tls
root@vault3:/etc/vault.d/tls# sudo chmod 640 /etc/vault.d/tls/vault.key
```

3. Let's configure vault.hcl files on each node.

Vault1:

```
root@vault1:/etc/vault.d# vi vault.hcl
```

```
ui = true
disable_mlock = true

listener "tcp" {
  address          = "0.0.0.0:8200"
  cluster_address = "0.0.0.0:8201"
  tls_disable      = 0
  tls_cert_file    = "/etc/vault.d/tls/vault.crt"
  tls_key_file     = "/etc/vault.d/tls/vault.key"
  tls_client_ca_file = "/etc/vault.d/tls/ca.crt"
}

storage "raft" {
  path    = "/var/lib/vault"
  node_id = "vault1"

  retry_join {
    leader_api_addr     = "https://192.168.100.201:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
  retry_join {
    leader_api_addr     = "https://192.168.100.202:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
  retry_join {
    leader_api_addr     = "https://192.168.100.203:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
}

api_addr     = "https://192.168.100.201:8200"
cluster_addr = "https://192.168.100.201:8201"
```

Vault2:

```
root@vault2:/etc/vault.d# vi vault.hcl
```

```
ui = true
disable_mlock = true

listener "tcp" {
  address          = "0.0.0.0:8200"
  cluster_address = "0.0.0.0:8201"
  tls_disable      = 0
  tls_cert_file    = "/etc/vault.d/tls/vault.crt"
  tls_key_file     = "/etc/vault.d/tls/vault.key"
  tls_client_ca_file = "/etc/vault.d/tls/ca.crt"
}

storage "raft" {
  path    = "/var/lib/vault"
  node_id = "vault2"

  retry_join {
    leader_api_addr     = "https://192.168.100.201:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
  retry_join {
    leader_api_addr     = "https://192.168.100.202:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
  retry_join {
    leader_api_addr     = "https://192.168.100.203:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
}

api_addr     = "https://192.168.100.202:8200"
cluster_addr = "https://192.168.100.202:8201"
```

Vault3:

```
root@vault3:/etc/vault.d# vi vault.hcl
```

```
ui = true
disable_mlock = true

listener "tcp" {
  address          = "0.0.0.0:8200"
  cluster_address = "0.0.0.0:8201"
  tls_disable      = 0
  tls_cert_file    = "/etc/vault.d/tls/vault.crt"
  tls_key_file     = "/etc/vault.d/tls/vault.key"
  tls_client_ca_file = "/etc/vault.d/tls/ca.crt"
}

storage "raft" {
  path    = "/var/lib/vault"
  node_id = "vault3"

  retry_join {
    leader_api_addr     = "https://192.168.100.201:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
  retry_join {
    leader_api_addr     = "https://192.168.100.202:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
  retry_join {
    leader_api_addr     = "https://192.168.100.203:8200"
    leader_ca_cert_file = "/etc/vault.d/tls/ca.crt"
  }
}

api_addr     = "https://192.168.100.203:8200"
cluster_addr = "https://192.168.100.203:8201"
```

4. **Create systemd service for each vault node and enable.**

Vault1:

```
root@vault1:/etc/vault.d# vi /etc/systemd/system/vault.service
root@vault1:/etc/vault.d# systemctl daemon-reload
root@vault1:/etc/vault.d# systemctl enable vault
Created symlink /etc/systemd/system/multi-user.target.wants/vault.service → /etc/systemd/system/vault.service.
```

Vault2:

```
root@vault2:/etc/vault.d# vi /etc/systemd/system/vault.service
root@vault2:/etc/vault.d# systemctl daemon-reload
root@vault2:/etc/vault.d# systemctl enable vault
Created symlink /etc/systemd/system/multi-user.target.wants/vault.service → /etc/systemd/system/vault.service.
```

Vault3:

```
root@vault3:/etc/vault.d# vi /etc/systemd/system/vault.service
root@vault3:/etc/vault.d# systemctl daemon-reload
root@vault3:/etc/vault.d# systemctl enable vault
Created symlink /etc/systemd/system/multi-user.target.wants/vault.service → /etc/systemd/system/vault.service.
```

File content is the same for all vault nodes:

```
[Unit]
Description=HashiCorp Vault
Documentation=https://www.vaultproject.io/
After=network-online.target
Wants=network-online.target

[Service]
User=vault
Group=vault
ExecStart=/usr/local/bin/vault server -config=/etc/vault.d/vault.hcl
ExecReload=/bin/kill --signal HUP $MAINPID
Restart=on-failure
LimitMEMLOCK=infinity
CapabilityBoundingSet=CAP_IPC_LOCK
AmbientCapabilities=CAP_IPC_LOCK
NoNewPrivileges=true
ProtectSystem=full
ProtectHome=read-only
PrivateTmp=yes
ProtectControlGroups=yes
ProtectKernelModules=yes
ProtectKernelTunables=yes

[Install]
WantedBy=multi-user.target
```

5. **Open firewall ports on each node.**

Vault1:

```
root@vault1:/etc/vault.d# firewall-cmd --permanent --add-port=8200/tcp
success
root@vault1:/etc/vault.d# firewall-cmd --permanent --add-port=8201/tcp
success
root@vault1:/etc/vault.d# firewall-cmd --reload
success
```

Vault2:

```
root@vault2:/etc/vault.d# firewall-cmd --permanent --add-port=8200/tcp
success
root@vault2:/etc/vault.d# firewall-cmd --permanent --add-port=8201/tcp
success
root@vault2:/etc/vault.d# firewall-cmd --reload
success
```

Vault3:

```
root@vault3:/etc/vault.d# firewall-cmd --permanent --add-port=8200/tcp
success
root@vault3:/etc/vault.d# firewall-cmd --permanent --add-port=8201/tcp
success
root@vault3:/etc/vault.d# firewall-cmd --reload
success
```

6. Start vault on each node now.

Vault1:

```
root@vault1:/etc/vault.d# systemctl start vault
```

Vault2:

```
root@vault2:/etc/vault.d# systemctl start vault
```

Vault3:

```
root@vault3:/etc/vault.d# systemctl start vault
```

7. Time to generate the cluster.

For this, firstly, export environment variables for vault1.

```
root@vault1:/etc/vault.d# export VAULT_ADDR="https://192.168.100.201:8200"
root@vault1:/etc/vault.d# export VAULT_CACERT="/etc/vault.d/tls/ca.crt"
```

Initialize vault1 and save the keys into vault_init.txt file:

```
root@vault1:/etc/vault.d# vault operator init -key-shares=5 -key-threshold=3 | tee ~/vault_init.txt
Unseal Key 1: sGddKf8gkLY6K1YF49vvDMRB8ybmLFKk6I/lhPOcBpHJ
Unseal Key 2: GIcQhlu3ejlGnpFhKGq3QlbAdxWYq2gH8gL90zAJ5Juf
Unseal Key 3: RbW4TAAe/6gpgZkHglbI6++slXa9IprqctyisOWoV5xP
Unseal Key 4: ZKzTo+yp3zLsdgEu+17EX3z9meOzLhuFjeB7tv5OGyae
Unseal Key 5: th3zOEFUsrx/nklKCjfCXDrboAzhxlg8Z6YS03PIVhhT

Initial Root Token: hvs.UlBgLieF7U1ZvlUu4GPAseQD

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 root key. Without at least 3 keys to
reconstruct the root 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 vault1 using any 3 out of 5 keys.

```
root@vault1:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  true
Total Shares            5
Threshold               3
Unseal Progress         1/3
Unseal Nonce            c97ffa4b-272b-5049-51d2-a622c3c69f02
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Removed From Cluster    false
HA Enabled              true
root@vault1:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  true
Total Shares            5
Threshold               3
Unseal Progress         2/3
Unseal Nonce            c97ffa4b-272b-5049-51d2-a622c3c69f02
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Removed From Cluster    false
HA Enabled              true
root@vault1:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  false
Total Shares            5
Threshold               3
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Cluster Name            vault-cluster-b7a86e21
Cluster ID              db28bfa4-b34c-4d99-9128-6de5725194aa
Removed From Cluster    false
HA Enabled              true
HA Cluster              n/a
HA Mode                 standby
Active Node Address     <none>
Raft Committed Index    32
Raft Applied Index      32
```

Set environment variables for vault2 and vault3, then unseal them as well.

Vault2:

```
root@vault2:/etc/vault.d# export VAULT_ADDR="https://192.168.100.202:8200"
root@vault2:/etc/vault.d# export VAULT_CACERT="/etc/vault.d/tls/ca.crt"
```

```
root@vault2:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  true
Total Shares            5
Threshold               3
Unseal Progress         1/3
Unseal Nonce            8a6aec41-3019-13f2-9586-51133825c1a0
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Removed From Cluster    false
HA Enabled              true
root@vault2:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  true
Total Shares            5
Threshold               3
Unseal Progress         2/3
Unseal Nonce            8a6aec41-3019-13f2-9586-51133825c1a0
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Removed From Cluster    false
HA Enabled              true
root@vault2:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  false
Total Shares            5
Threshold               3
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Cluster Name            vault-cluster-b7a86e21
Cluster ID              db28bfa4-b34c-4d99-9128-6de5725194aa
Removed From Cluster    false
HA Enabled              true
HA Cluster              https://192.168.100.201:8201
HA Mode                 standby
Active Node Address     https://192.168.100.201:8200
Raft Committed Index    40
Raft Applied Index      40
```

Vault3:

```
root@vault3:/etc/vault.d# export VAULT_ADDR="https://192.168.100.203:8200"
root@vault3:/etc/vault.d# export VAULT_CACERT="/etc/vault.d/tls/ca.crt"
```

```
root@vault3:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  true
Total Shares            5
Threshold               3
Unseal Progress         1/3
Unseal Nonce            9a4351a6-f9b8-2b0f-8c49-e182c2768ced
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Removed From Cluster    false
HA Enabled              true
root@vault3:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  true
Total Shares            5
Threshold               3
Unseal Progress         2/3
Unseal Nonce            9a4351a6-f9b8-2b0f-8c49-e182c2768ced
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Removed From Cluster    false
HA Enabled              true
root@vault3:/etc/vault.d# vault operator unseal
Unseal Key (will be hidden):
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  false
Total Shares            5
Threshold               3
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Cluster Name            vault-cluster-b7a86e21
Cluster ID              db28bfa4-b34c-4d99-9128-6de5725194aa
Removed From Cluster    false
HA Enabled              true
HA Cluster              https://192.168.100.201:8201
HA Mode                 standby
Active Node Address     https://192.168.100.201:8200
Raft Committed Index    48
Raft Applied Index      48
```

Verify cluster:

On CLI:

```
root@vault1:/etc/vault.d# vault login
Token (will be hidden):
Success! You are now authenticated. The token information displayed below
is already stored in the token helper. You do NOT need to run "vault login"
again. Future Vault requests will automatically use this token.

Key                  Value
---                  -----
token                hvs.UlBgLieF7U1ZvlUu4GPAseQD
token_accessor       HIOb9lPDrn9GUEDtFsciQ943
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
root@vault1:/etc/vault.d# vault status
Key                     Value
---                     -----
Seal Type               shamir
Initialized             true
Sealed                  false
Total Shares            5
Threshold               3
Version                 1.20.2
Build Date              2025-08-05T19:05:39Z
Storage Type            raft
Cluster Name            vault-cluster-b7a86e21
Cluster ID              db28bfa4-b34c-4d99-9128-6de5725194aa
Removed From Cluster    false
HA Enabled              true
HA Cluster              https://192.168.100.201:8201
HA Mode                 active
Active Since            2025-08-14T11:56:39.095599687Z
Raft Committed Index    48
Raft Applied Index      48
root@vault1:/etc/vault.d# vault operator raft list-peers
Node      Address                 State       Voter
----      -------                 -----       -----
vault1    192.168.100.201:8201    leader      true
vault2    192.168.100.202:8201    follower    true
vault3    192.168.100.203:8201    follower    true
```

On UI:

Raft Storage

Snapshots

Address	Leader	Voter	Actions
192.168.100.201:8201	Yes	Yes	...
192.168.100.202:8201	No	Yes	...
192.168.100.203:8201	No	Yes	...

8. **Configure HAProxy and Keepalived for both haproxy nodes.**

Haproxy1:

Install:

```
root@haproxy1:~# apt install -y haproxy keepalived
```

Configure /etc/haproxy/haproxy.cfg:

```
root@haproxy1:~# vi /etc/haproxy/haproxy.cfg
global
    log /dev/log local0
    maxconn 20000
    daemon

defaults
    mode tcp
    log global
    option tcplog
    timeout connect 5s
    timeout client  30s
    timeout server  30s

frontend vault_api
    bind :8200
    default_backend vault_nodes

backend vault_nodes
    balance source
    option tcp-check
    server vault1 192.168.100.201:8200 check
    server vault2 192.168.100.202:8200 check
    server vault3 192.168.100.203:8200 check
```

Test config and start/enable:

```
root@haproxy1:~# haproxy -c -f /etc/haproxy/haproxy.cfg
Configuration file is valid
root@haproxy1:~# systemctl enable --now haproxy
Synchronizing state of haproxy.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable haproxy
```

Next, create /etc/keepalived/keepalived.conf file:

```
root@haproxy1:~# vi /etc/keepalived/keepalived.conf
```

```
vrrp_instance VI_1 {
    state MASTER
    interface eth1
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass StrongPassword
    }
    virtual_ipaddress {
        192.168.100.210/24 dev eth1
    }
}
```

Haproxy2:

Install:

```
root@haproxy2:~# apt install -y haproxy keepalived
```

Configure /etc/haproxy/haproxy.cfg:

```
root@haproxy2:~# vi /etc/haproxy/haproxy.cfg
```

```
global
    log /dev/log local0
    maxconn 20000
    daemon

defaults
    mode tcp
    log global
    option tcplog
    timeout connect 5s
    timeout client  30s
    timeout server  30s

frontend vault_api
    bind :8200
    default_backend vault_nodes

backend vault_nodes
    balance source
    option tcp-check
    server vault1 192.168.100.201:8200 check
    server vault2 192.168.100.202:8200 check
    server vault3 192.168.100.203:8200 check
```

Test config and start/enable:

```
root@haproxy2:~# haproxy -c -f /etc/haproxy/haproxy.cfg
Configuration file is valid
root@haproxy2:~# systemctl enable --now haproxy
Synchronizing state of haproxy.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable haproxy
```

Next, create /etc/keepalived/keepalived.conf file:

```
root@haproxy2:~# vi /etc/keepalived/keepalived.conf
```

```
vrrp_instance VI_1 {
    state BACKUP
    interface eth1
    virtual_router_id 51
    priority 90
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass StrongPassword
    }
    virtual_ipaddress {
        192.168.100.210/24 dev eth1
    }
}
```

Enable and start keepalived for both nodes.

```
root@haproxy1:~# systemctl enable --now keepalived
Synchronizing state of keepalived.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable keepalived
```

```
root@haproxy2:~# systemctl enable --now keepalived
Synchronizing state of keepalived.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable keepalived
```

Prove keepalived & haproxy nodes work:

1) **Pinging VIP on vault1:**

```
root@vault1:/etc/vault.d# ping -c 4 192.168.100.210
PING 192.168.100.210 (192.168.100.210) 56(84) bytes of data.
64 bytes from 192.168.100.210: icmp_seq=1 ttl=64 time=0.182 ms
64 bytes from 192.168.100.210: icmp_seq=2 ttl=64 time=0.268 ms
64 bytes from 192.168.100.210: icmp_seq=3 ttl=64 time=0.272 ms
64 bytes from 192.168.100.210: icmp_seq=4 ttl=64 time=0.877 ms

--- 192.168.100.210 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3247ms
rtt min/avg/max/mdev = 0.182/0.399/0.877/0.277 ms
```

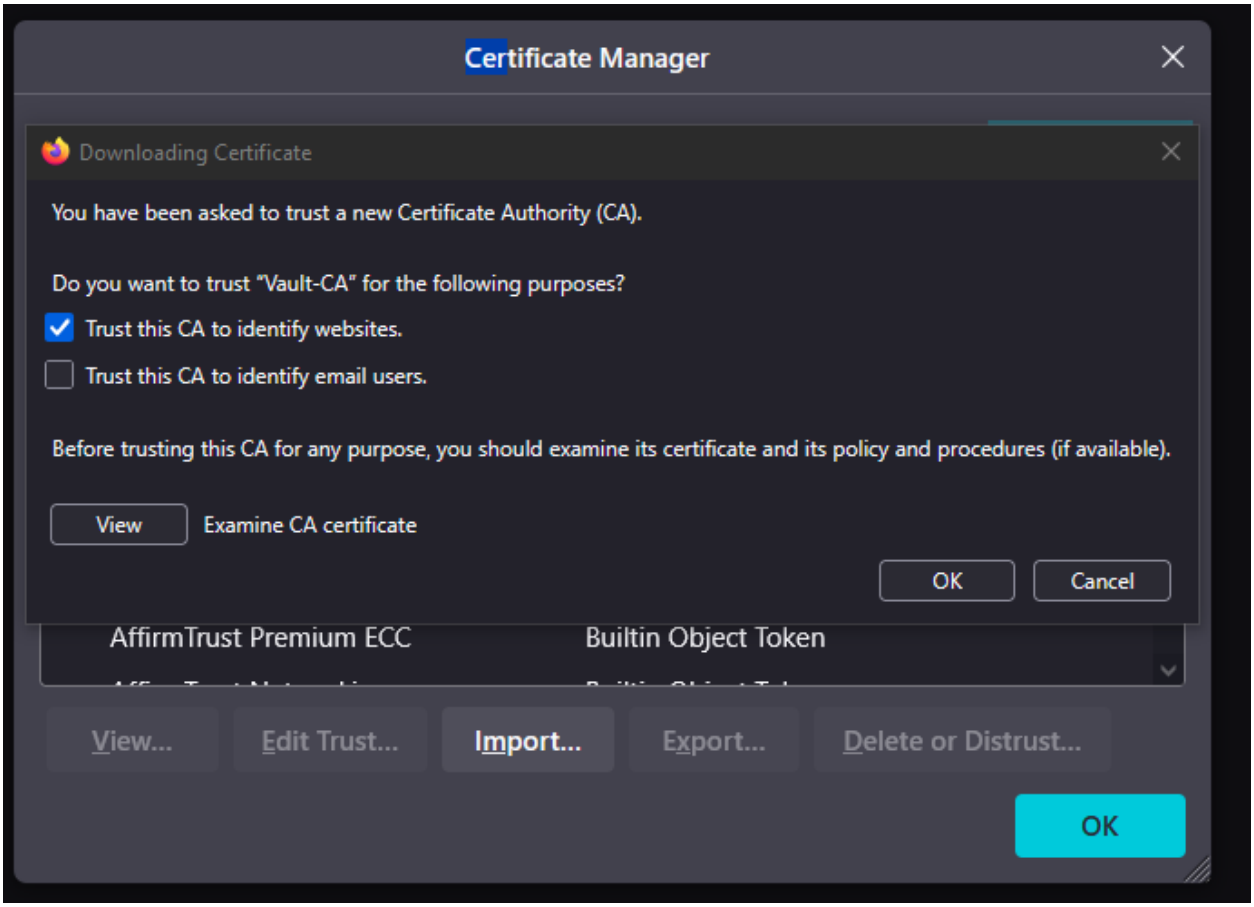
2) **"ip a" output on haproxy1:**

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b5:eb:a8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.204/24 brd 192.168.100.255 scope global dynamic eth1
       valid_lft 76972sec preferred_lft 76972sec
    inet 192.168.100.210/24 scope global secondary eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb5:eba8/64 scope link
       valid_lft forever preferred_lft forever
```

3) **Finally, to access vault via domain-name (vault.local), we have to transfer ca.cert from Linux to Windows and add an entry to "C:\Windows\System32\drivers\etc\hosts" file on Windows.**

Command: cscp root@192.168.100.201:/etc/vault.d/tls/ca.crt Desktop

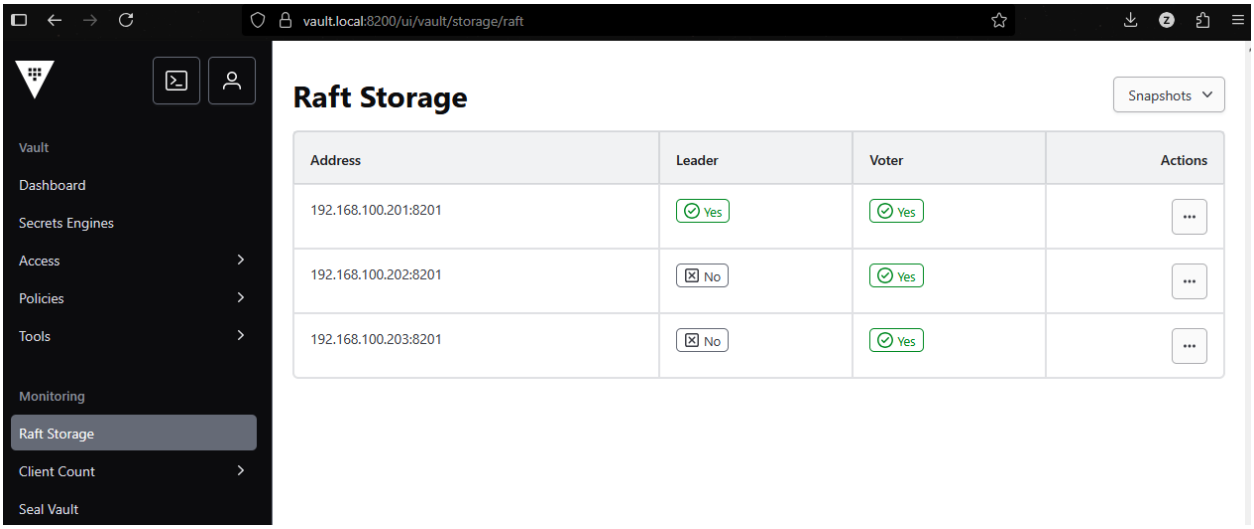
Verify CA on browser.



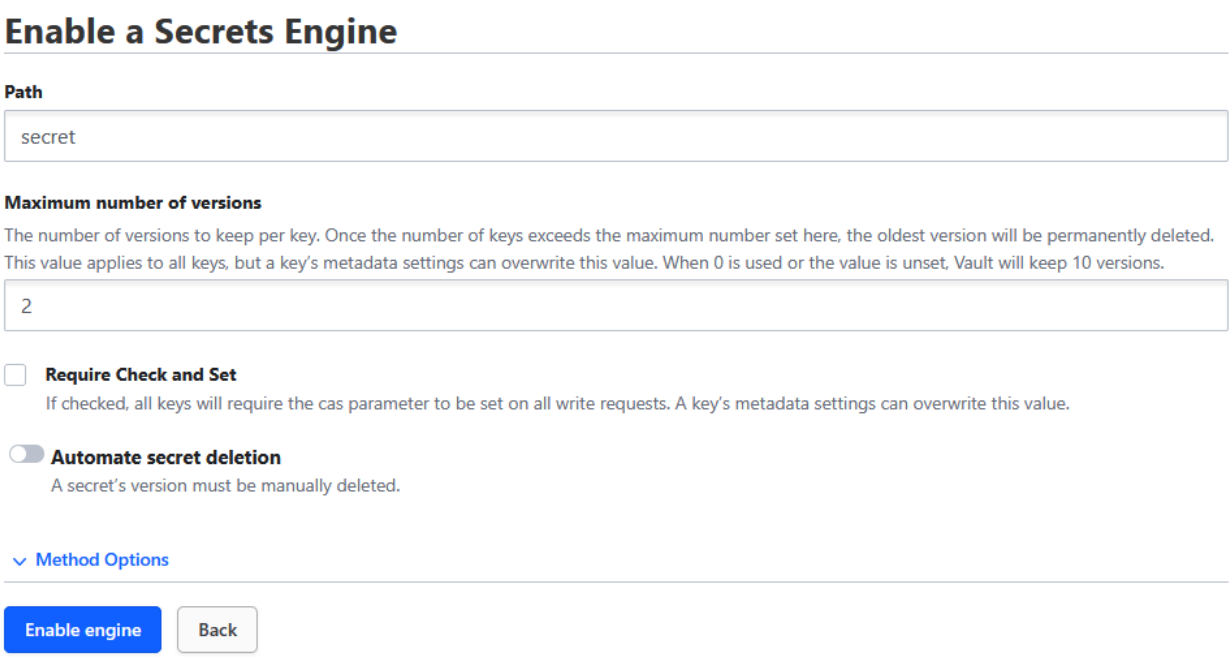
Add entry into Window's host file.

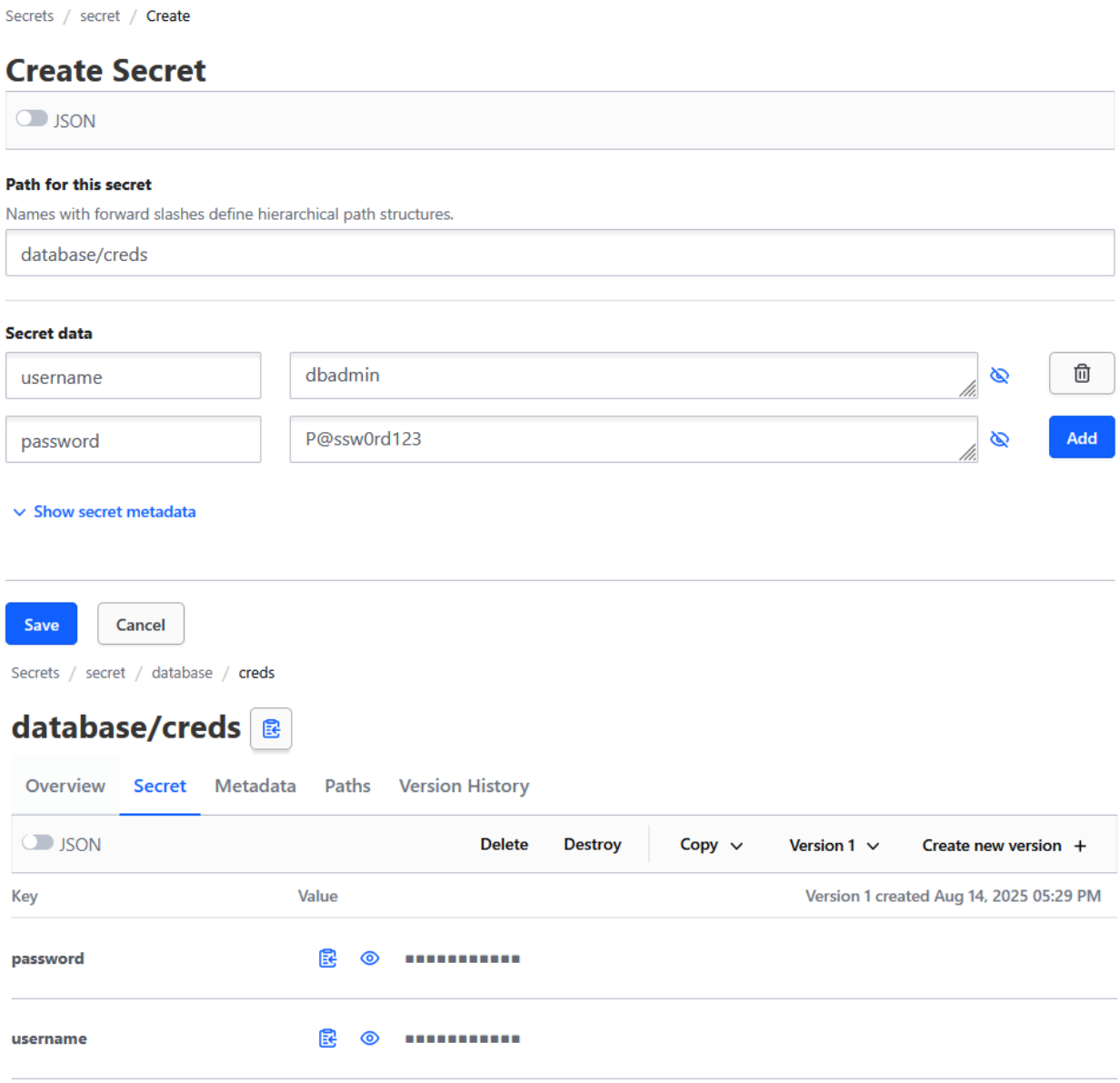
```
192.168.100.210   vault.local
```

Access via domain-name with https connection.



9. Enable and Configure KV Secret Engine





10. Create Group Policies for Admins and Developers.

Admin Policy:

ACL policies / Create

Create ACL Policy

Name

admin-policy.hcl

You can use Alt+Tab (Option+Tab on MacOS) in the code editor to skip to the next field.

Policy How to write a policy **Upload file**

```
# admin-policy.hcl
path "*" {
  capabilities = ["create", "read", "update", "delete", "list", "sudo"]
}
```

Create policy Cancel

ACL policies / admin-policy.hcl

admin-policy.hcl

Download policy Edit policy

Policy
(hcl format)

```
# admin-policy.hcl
path "*" {
  capabilities = ["create", "read", "update", "delete", "list", "sudo"]
}
```

Developer Policy:

ACL policies / Create

Create ACL Policy

Name

dev-policy.hcl

You can use Alt+Tab (Option+Tab on MacOS) in the code editor to skip to the next field.

Policy How to write a policy **Upload file**

```
# dev-policy.hcl
path "secret/data/dev/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}

path "secret/metadata/dev/*" {
  capabilities = ["list", "delete"]
}
```

Create policy Cancel

ACL policies / dev-policy.hcl

dev-policy.hcl

Download policy Edit policy

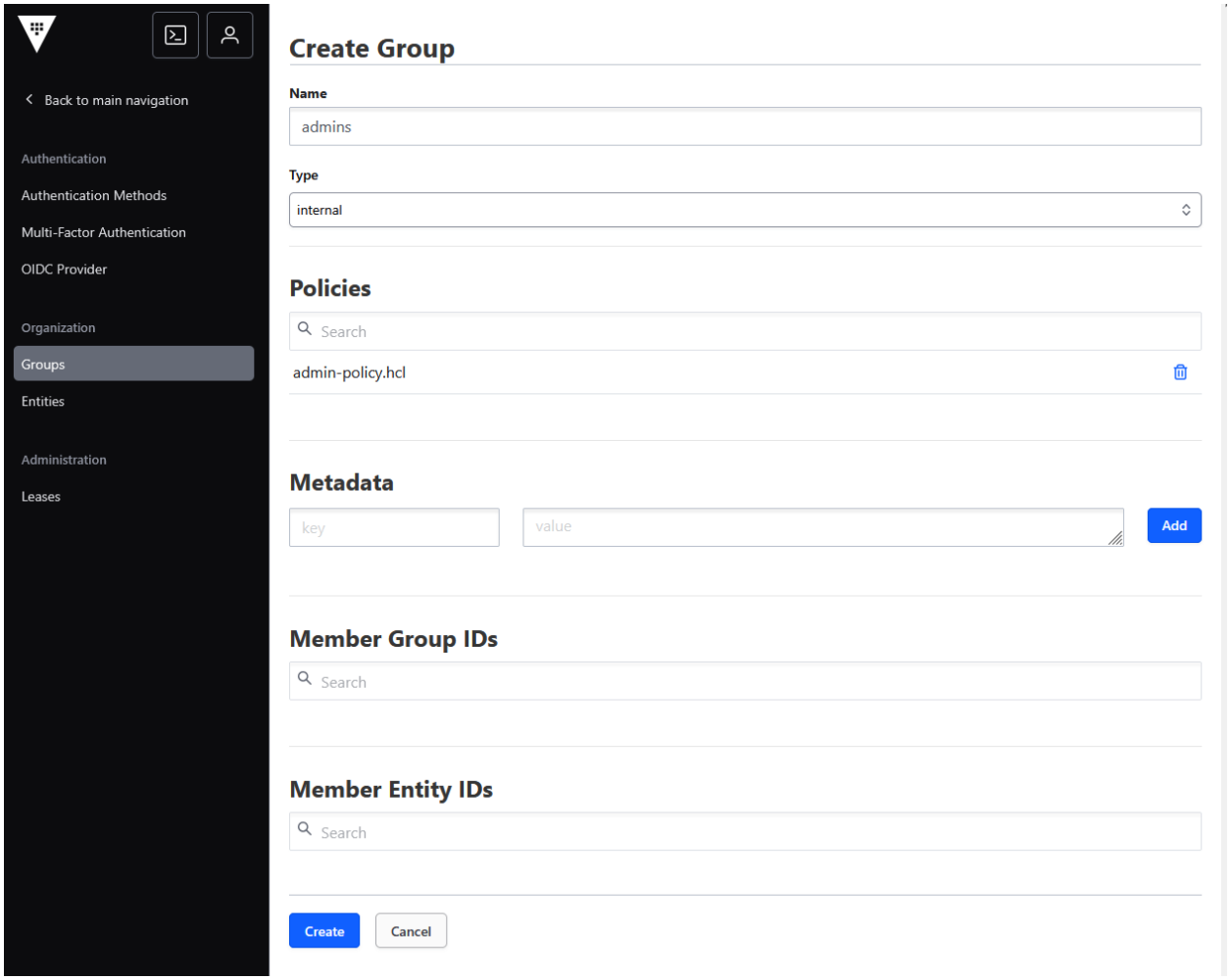
Policy
(hcl format)

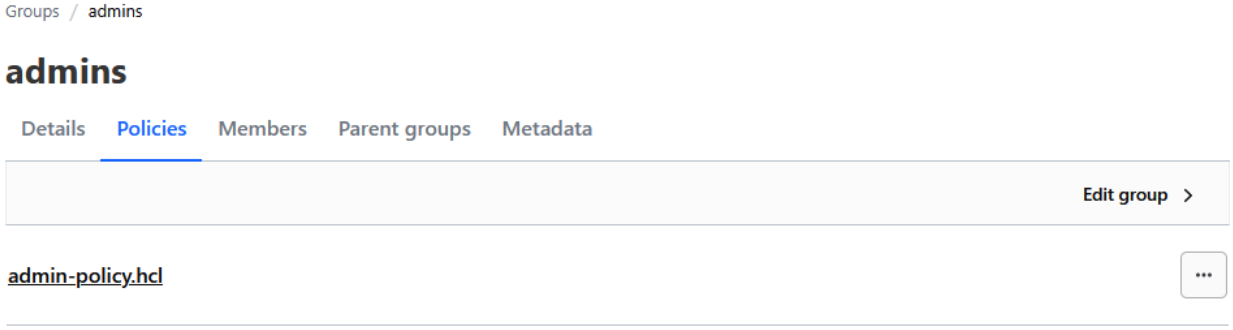
```
# dev-policy.hcl
path "secret/data/dev/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}

path "secret/metadata/dev/*" {
  capabilities = ["list", "delete"]
}
```

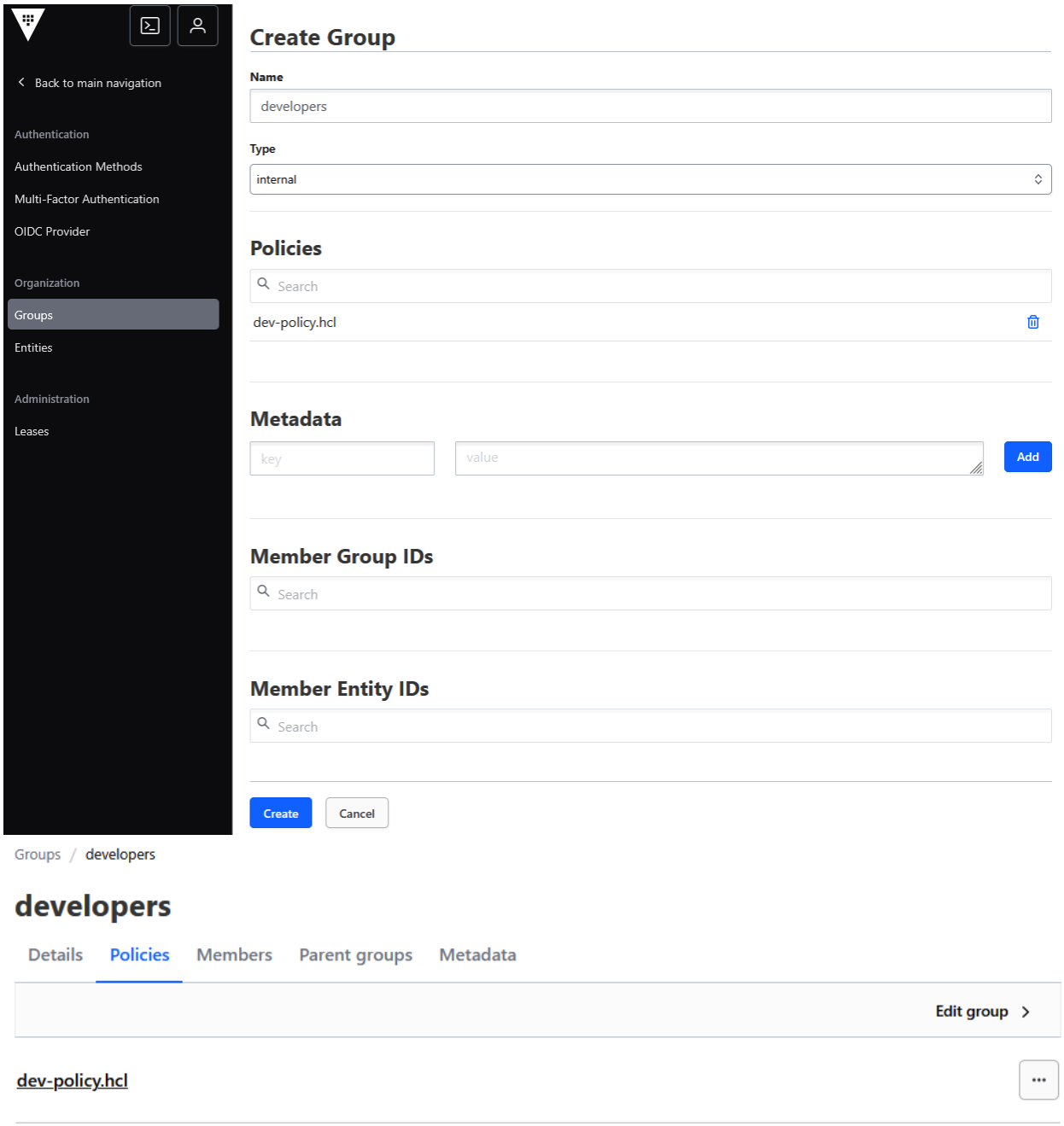
11. Create admins and developers groups and attach their policies.

Admins:


Back to main navigation
Authentication
Authentication Methods
Multi-Factor Authentication
OIDC Provider
Organization
Groups
Entities
Administration
Leases
Create Group
Name
admins
Type
internal
Policies
Search
admin-policy.hcl
Metadata
key
value
Add
Member Group IDs
Search
Member Entity IDs
Search
Create
Cancel


Groups / admins
admins
Details
Policies
Members
Parent groups
Metadata
Edit group
admin-policy.hcl

Developers:



12. Generate and Use Tokens for Authentication:

Create an Admin Token:

```
root@vault1:/etc/vault.d# vault token create -policy="admin-policy.hcl" -display-name="admin-token" \
>
Key                  Value
---                  -----
token                hvs.CAESIN8gO1AQfI8erQzhs738RrHnToss-p3PBc8gfEK7c9kEGh4KHGh2cy4wczdvN2JobHp6MDZFRkV5OEIxdDZMaVY
token_accessor       beHXOXZn6Ukh6frmXyKGDKDv
token_duration       768h
token_renewable      true
token_policies       ["admin-policy.hcl" "default"]
identity_policies    []
policies             ["admin-policy.hcl" "default"]
```

Create a Developer Token:

```
root@vault1:/etc/vault.d# vault token create -policy="dev-policy.hcl" -display-name="dev-token" \
>
Key                  Value
---                  -----
token                hvs.CAESIIm4mlza_5agc8BumiQIOtSEV8SGE0Ki8l1We5GiH9lvGh4KHGh2cy5FOW9kT1RQblNUZUp0OXdXWW0zM1R5STU
token_accessor       6Ba3vYrrS3xNj7FA9ErwLDEF
token_duration       768h
token_renewable      true
token_policies       ["default" "dev-policy.hcl"]
identity_policies    []
policies             ["default" "dev-policy.hcl"]
```

Test Admin Token:

```
root@vault1:/etc/vault.d# vault login
Token (will be hidden):
Success! You are now authenticated. The token information displayed below
is already stored in the token helper. You do NOT need to run "vault login"
again. Future Vault requests will automatically use this token.

Key                  Value
---                  -----
token                hvs.CAESICBxhpX1x6FFP9VCuwKcWB_upmaZQv_2QeVkpIrupyFcGh4KHGh2cy55SGNRSkRDT0o5YkZBdWI3NW5GNkpzenA
token_accessor       jC6N7DGqEzQL0h4R6x4ZUT07
token_duration       767h58m10s
token_renewable      true
token_policies       ["admin-policy.hcl" "default"]
identity_policies    []
policies             ["admin-policy.hcl" "default"]
root@vault1:/etc/vault.d# vault kv get -mount="secret" "database/creds"
======= Secret Path =======
secret/data/database/creds

======= Metadata =======
Key                Value
---                -----
created_time       2025-08-14T13:29:04.727481907Z
custom_metadata    <nil>
deletion_time      n/a
destroyed          false
version            1

====== Data ======
Key         Value
---         -----
password    P@ssw0rd123
username    dbadmin
```