Context

1. There are two Vault Clusters, vault.primary and vault.secondary
2. First objective is to enable DR replication and configure vault.primary as the Primary Cluster, and vault.secondary as the Secondary cluster
3. Second objective is to promote vault.secondary as the NEW Primary, and demote vault.primary to become the new Secondary without losing data.
4. Third objective is to promote vault.primary as the NEW Primary again and demote vault.secondary as the NEW Secondary again without losing data.
5. TLS is not enabled

References

1. Vault DR API
2. Monitoring Replication
3. Learn Guide on Disaster Recovery Setup

Step One - Setup Commands

- Setup Reusable commands on vault_primary so that everthing can be executed from one vault cluster

```
 /* In the Beginning the Secondary DR cluster will have its own Root roken and Unseal Ke
When it is enabled as a DR cluster
It will adopt the Primary's cluster own Root Token and Unseal key */

export VAULT_PRIMARY_ADDR=http://vault.primary:8200
export VAULT_SECONDARY_ADDR=http://vault.secondary:8200

export VAULT_SECONDARY_CLUSTER_ADDR=http://vault.secondary:8201
export VAULT_PRIMARY_CLUSTER_ADDR=http://vault.primary:8201

vault_primary () {
VAULT_ADDR=${VAULT_PRIMARY_ADDR} vault $@
}

vault_secondary () {
  VAULT_ADDR=${VAULT_SECONDARY_ADDR} vault $@
}

# Save Primary and DR Tokens

read -rs ROOT_TOKEN
 <enter the token>
 export VAULT_TOKEN=${ROOT_TOKEN}


read -rs DR_ROOT_TOKEN
 <enter the token>
```

Step Two - Enable vault.primary as the Primary Cluster and enable DR Replication

- Enable Replication on Primary Cluster

```
 export VAULT_TOKEN=${ROOT_TOKEN}
vault_primary login ${ROOT_TOKEN}
vault_primary write -f /sys/replication/dr/primary/enable

sleep 10
```

> Here is the command to Disable replication.
> Use with caution for it cause any secondary cluster that reconnects to wipe out its data.

> `vault_primary write -f /sys/replication/dr/primary/disable`

- Create an Unwrapped Token to Link a Secondary Cluster

```
PRIMARY_DR_TOKEN=$(vault_primary write -format=json /sys/replication/dr/primary/secondar

echo $PRIMARY_DR_TOKEN

vault_primary read sys/replication/dr/status
```

Note to revoke a token

> `vault_primary write -format=json /sys/replication/performance/primary/revoke-secondary id=first_secondary`

- Enable Secondary Cluster (vault_secondary) as a DR Cluser

```
export VAULT_TOKEN=${DR_ROOT_TOKEN}

vault_secondary login $DR_ROOT_TOKEN

vault_secondary write /sys/replication/dr/secondary/enable token=${PRIMARY_DR_TOKEN} pri

sleep 10

vault_secondary read sys/replication/dr/status
```

- Observe that the cluster ids are the same when you run replicatiin status on both clusters. Pay attention to mode, primary cluster address, and secondary list

```
vault_secondary read -format=json sys/replication/dr/status
```

> ```{ "request_id": "5e749ba7-b135-203a-e24f-55fc8ec9af2c", "lease_id": "", "lease_duration": 0, "renewable": false, "data": { "cluster_id": "e9a5861c-0c11-5927-463b-1d975a9ac8b3", "known_primary_cluster_addrs": [ "https://192.168.56.107:8201" ], "last_reindex_epoch": "0", "last_remote_wal": 0, "merkle_root": "bb438643500e8bd7d32bd027bba0209730c2129f", "mode": "secondary", "primary_cluster_addr": "http://vault.secondary:8201", "secondary_id": "first_secondary", "state": "stream-wals" }, "warnings": null }

```
vault_primary read -format=json sys/replication/dr/status
```

> ```{ "request_id": "514b4d43-78cf-cb98-9451-c07381729674", "lease_id": "", "lease_duration": 0, "renewable": false, "data": { "cluster_id": "e9a5861c-0c11-5927-463b-1d975a9ac8b3", "known_secondaries": [ "first_secondary" ], "last_reindex_epoch": "0", "last_wal": 101, "merkle_root": "8dc4ff8d6408ce65629c49c5c88b667264c41b13", "mode": "primary", "primary_cluster_addr": "http://vault.secondary:8201", "state": "running" }, "warnings": null }

Step Three - Demote Primary Cluster as Secondary Before Making DR CLuster Primary

- Always take care to never have two primary clusters running. You may lose data

```
### FIRST Demote primary vault instance. You CANNOT Have Two primary Instances at once

export VAULT_TOKEN=${ROOT_TOKEN}
vault_primary login ${ROOT_TOKEN}

curl --header "X-Vault-Token: ${VAULT_TOKEN}" --request POST ${VAULT_PRIMARY_ADDR}/v1/sy

vault_primary read -format=json sys/replication/dr/status
```

> Equivalent CLI Command
>
> > `vault_primary write -f /sys/replication/dr/primary/demote`

Step Four - Promote vault.secondary DR Cluster to Primary

- In order to accomplish this you need a DR Operation Token on the DR Cluster to perform any operations
- Remember a DR cluster cannot accept any external transactions normally

BEGIN Generate DR Token

1. Generate One Time Password (OTP) Needed to Generate DR token

```
ONE_TIME_PASSWORD=$(vault_secondary operator  generate-root -dr-token -generate-otp)
echo $ONE_TIME_PASSWORD
```

> Alternatively you can also
>
> > `vault operator generate-root -dr-token -init`

2. Start Generation of DR Operation Token Attempt

- Get NONCE to give to all you UNSEAL KEY holders

```
NONCE=$(curl --header "X-Vault-Token: ${VAULT_TOKEN}" --request PUT --data '{"otp":"'"$
echo ${NONCE}
```

> To cancel attempt at any time
>
> > `vault_secondary delete /sys/replication/dr/secondary/generate-operation-token/attempt`

3. Get Your ENCODED TOKEN that Will be Combined with OTP to Produce DR operation Token

- Provie UNSEAL SEAL Keys one at a time until you Get the ENCODED TOKEN at last attempt.
- The Encoded Token will Only be produced upon last UNSEAL Key entered

```
 # Repeat for each UNSEAL KEY
# If you have 3 UNSEAL KEYS as your UNSEAL threashold you can do this
# Alternatively create a for loop

read -rs UNSEAL_KEY
<enter the unseal key>

read -rs UNSEAL_KEY2
<enter the unseal key 2>

read -rs UNSEAL_KEY3
<enter the unseal key 3>


ENCODED_TOKEN=$(curl  --header "X-Vault-Token: ${VAULT_TOKEN}" --request PUT --data '{"k

#ENCODED_TOKEN=$(curl  --header "X-Vault-Token: ${VAULT_TOKEN}" --request PUT --data '{"
#ENCODED_TOKEN=$(curl  --header "X-Vault-Token: ${VAULT_TOKEN}" --request PUT --data '{"

echo ${ENCODED_TOKEN}
```

> Alternative CLI commands
>
> > `ENCODED_TOKEN=$(vault_secondary operator generate-root -format=json -dr-token -nonce=${NONCE} ${UNSEAL_KEY} | jq --raw-output '.encoded_token')`
> >
> > `ENCODED_TOKEN=$(vault_secondary operator generate-root -format=json -dr-token -nonce=${NONCE} ${UNSEAL_KEY2} | jq --raw-output '.encoded_token')`
> >
> > `ENCODED_TOKEN=$(vault_secondary operator generate-root -format=json -dr-token -nonce=${NONCE} ${UNSEAL_KEY3} | jq --raw-output '.encoded_token')`

4. Generate DR TOKEN FINALLY

```
DR_PROMOTE_TOKEN=$(vault_secondary operator generate-root -dr-token -otp="${ONE_TIME_PAS

echo ${DR_PROMOTE_TOKEN}
```

> NOTE: The DR_PROMOTE_TOKEN must begin with a 's.'. If it returns anything else, repeat steps to generate it again

END Generate DR Token

---

5. Promote vault.secondary DR Cluster to PRIMARY

```
curl --header "X-Vault-Token: ${VAULT_TOKEN}" --request POST --data '{"dr_operation_tok

# check status
vault_secondary read -format=json sys/replication/dr/status
```

> Alternative command
>
> ```
> #vault_secondary write -f /sys/replication/dr/secondary/promote
> dr_operation_token="${DR_PROMOTE_TOKEN}"
> primary_cluster_addr="${VAULT_SECONDARY_ADDR}"
> ```

Step Five: Switch commands for New Primary and New Secondary

- Now remember
    1. vault.secondary is now your NEW ACTIVE PRIMARY
    2. vaul.primary is now you NEW DR SECONDARY
- When you promoted DR Secondary and demoted Active Primary you broke he replcation link
- NOW RE-ESTABLISH RELATIONSHIP BETWEEN NEW PRIMARY AND NEW SECONDARY

```
# Reverse which is Primary and Which is Not

export VAULT_PRIMARY_ADDR=http://vault.secondary:8200

export VAULT_SECONDARY_ADDR=http://vault.primary:8200

export VAULT_SECONDARY_CLUSTER_ADDR=http://vault.primary:8201
export VAULT_PRIMARY_CLUSTER_ADDR=http://vault.secondary:8201

vault_primary () {
VAULT_ADDR=${VAULT_PRIMARY_ADDR} vault $@
}


vault_secondary () {
  VAULT_ADDR=${VAULT_SECONDARY_ADDR} vault $@
}

vault_primary login ${VAULT_TOKEN}
```

Step Six: Relink Primary with Secondary without Losing Data

- Get a New Secondary Unwrap Token to relink secondary

```
vault_primary write -format=json /sys/replication/dr/primary/revoke-secondary id=demote

PRIMARY_DR_TOKEN=$(vault_primary write -format=json /sys/replication/dr/primary/secondar

echo "PRIMARY_DR_TOKEN : $PRIMARY_DR_TOKEN"

vault_primary  read  -format=json sys/replication/dr/status
```

Step Seven: Update the DR Secondary Cluster to find New Primary

- You once again need to generate a new DR operation Token to update Secondary with new Primaruy Cluster address
- GOTO Abover GENERATE_DR_TOKEN Code block above, execute it and AND RETURN HERE

- Update New DR Secondary Cluster with the Right Token and Primary ADDR

```
curl --header "X-Vault-Token: ${VAULT_TOKEN}" --request POST --data '{"dr_operation_tok

vault_secondary  read  -format=json sys/replication/dr/status
```

> Alternative Code
>
> ```
> vault_secondary write -f /sys/replication/dr/secondary/update-primary
> dr_operation_token=${DR_PROMOTE_TOKEN} token=${PRIMARY_DR_TOKEN}
> primary_adi_addr=${VAULT_PRIMARY_ADDR}
> ```

Step Eight: Clean Up DR token Used

- IMPORTANT Delete DR TOKEN USED

```
curl --request POST --data '{"dr_operation_token":"'"${DR_PROMOTE_TOKEN}"'"}'
${VAULT_SECONDARY_ADDR}/v1/sys/replication/dr/secondary/operation-token/delete
```

Step Nine: Re-promote vault.primary to Primary and Demote vault.secondary to DR Secondary again

1. Perform Demote Steps again here atStep Three - Demote Primary Cluster as Secondary Before Making DR CLuster Primary
2. Peform Step Four - Promote vault.secondary DR Cluster to Primary
3. Reswitch commands

```
 # Return vault.primary to Primary and vault.secondary to DR

export VAULT_PRIMARY_ADDR=http://vault.primary:8200
export VAULT_SECONDARY_ADDR=http://vault.secondary:8200

export VAULT_SECONDARY_CLUSTER_ADDR=http://vault.secondary:8201
export VAULT_PRIMARY_CLUSTER_ADDR=http://vault.primary:8201

vault_primary () {
VAULT_ADDR=${VAULT_PRIMARY_ADDR} vault $@
}

vault_secondary () {
  VAULT_ADDR=${VAULT_SECONDARY_ADDR} vault $@
}
```

4. Do Steps, 6, 7, 8