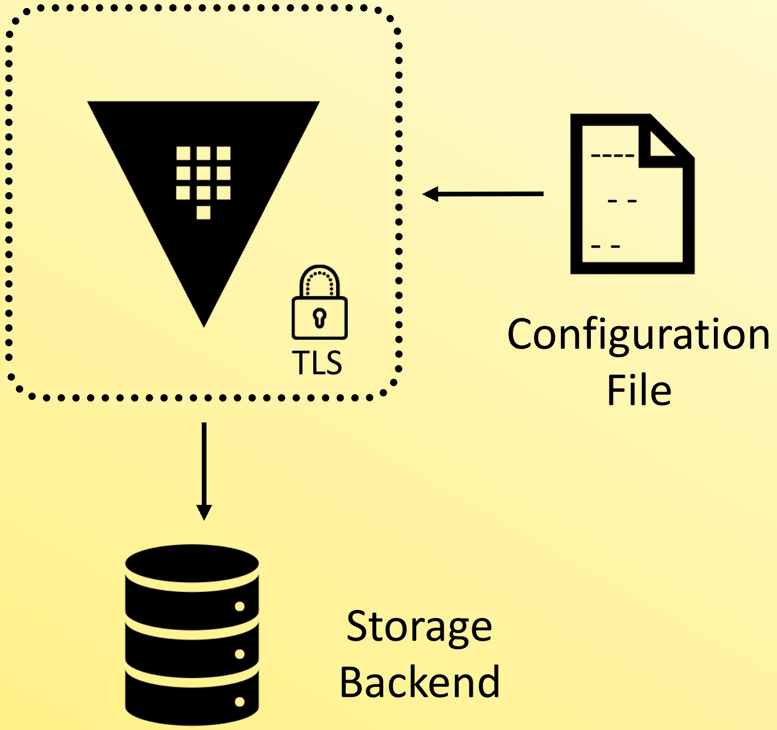
Vault

CERTIFIED
OPERATIONS
PROFESSIONAL

Configure a Highly Available [HA] Cluster

Single-Node Vault Server



Not a Recommended Architecture

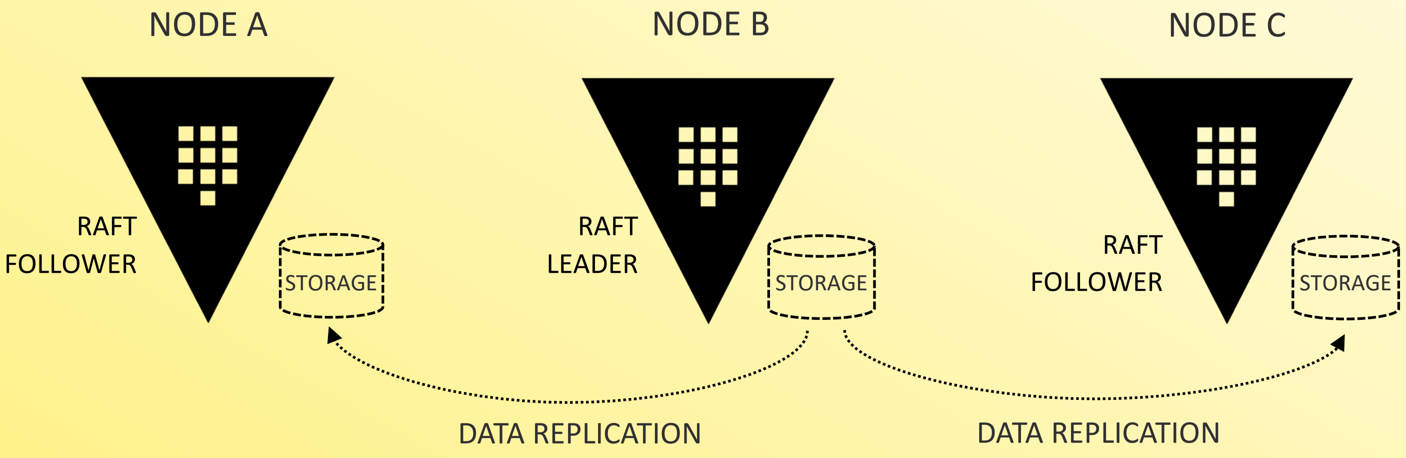
- No redundancy
- No scalability
- No failure tolerance

What Should a Cluster Look Like?

- Ideally, we want something that provides redundancy, failure tolerance, scalability, and a fully replicated architecture
- For Vault Enterprise, you are limited to either Integrated Storage or Consul storage backends
- HashiCorp (and consultants like me) are moving away from Consul as the primary storage backend and using Integrated Storage for everything
- The Vault Operations Professional exam will NOT feature Consul as a configuration or deployment option

Multi-Node Cluster using Integrated Storage

- Integrated Storage (aka Raft) allows Vault nodes to provide its own replicated storage across the Vault nodes within a cluster
- Define a local path to store replicated data
- All data is replicated among all nodes in the cluster

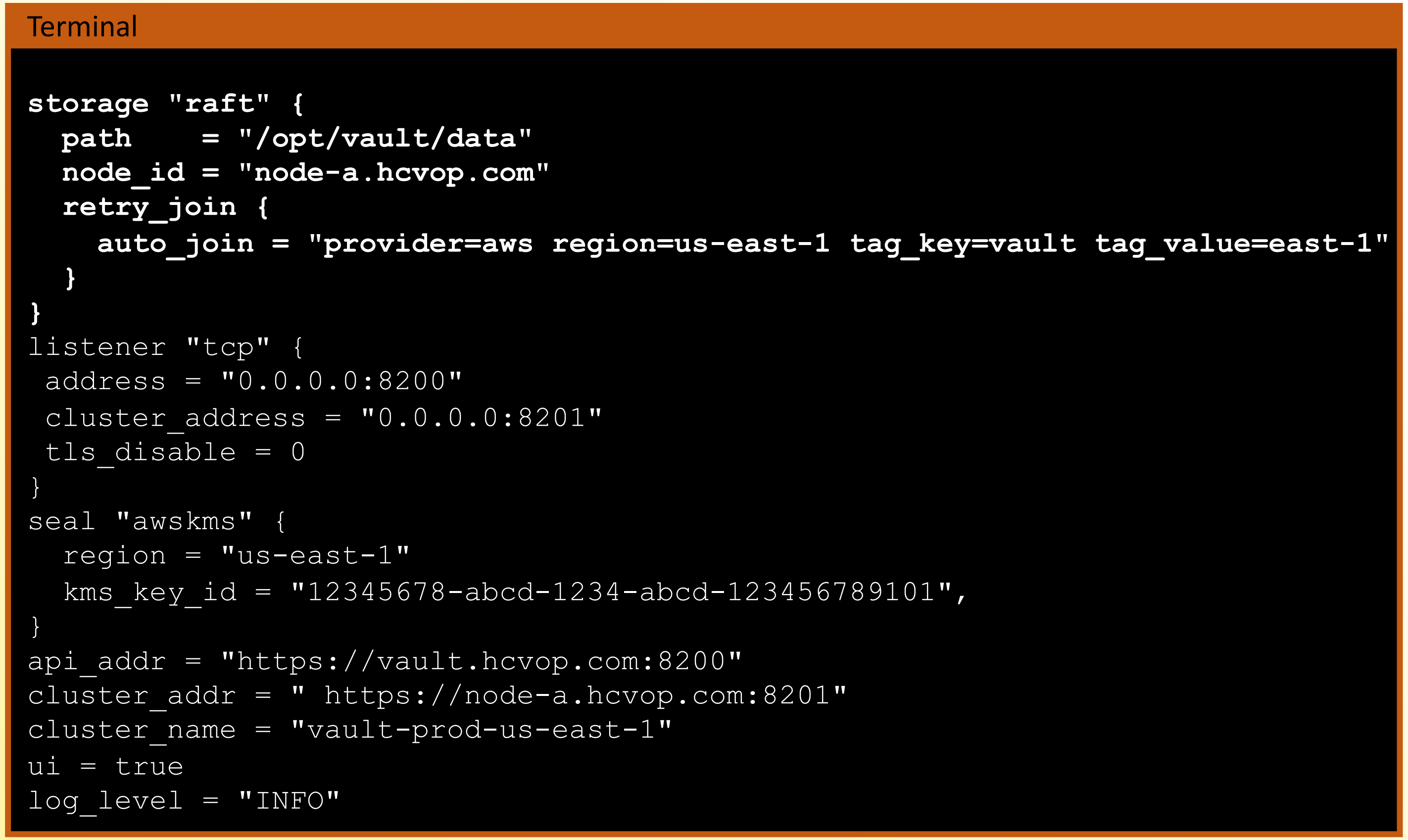


How Do I Configure Integrated Storage?

- Initial configuration of Integrated Storage is done in the Vault configuration file

- Multiple ways to join nodes to create a Vault cluster in the configuration file….or you do it manually

- Use `retry_join` stanza to automate the creation of the cluster from participating Vault nodes



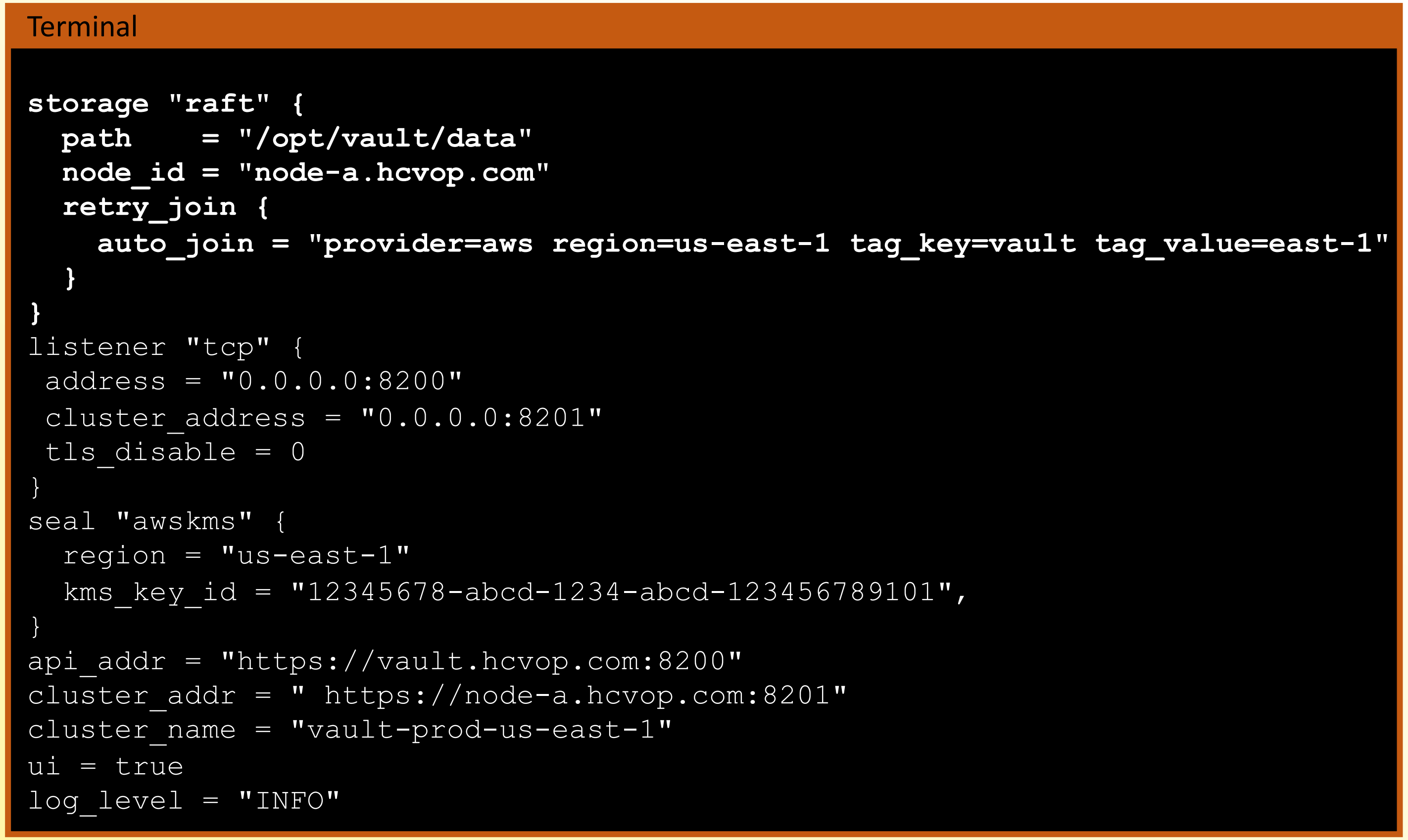
```
storage "raft" {
  path     = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    auto_join = "provider=aws region=us-east-1 tag_key=vault tag_value=east-1"
  }
}
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
}
seal "awskms" {
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101",
}
api_addr = "https://vault.hcvop.com:8200"
cluster_addr = " https://node-a.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

How Do I Configure Integrated Storage?

- **`path`** = the filesystem path where all the Vault data will be stored

- **`node_id`** = the identifier for the node in the cluster – cannot be duplicated within a cluster

- **`retry_join`** [optional] – automatically join the listed nodes to create a cluster

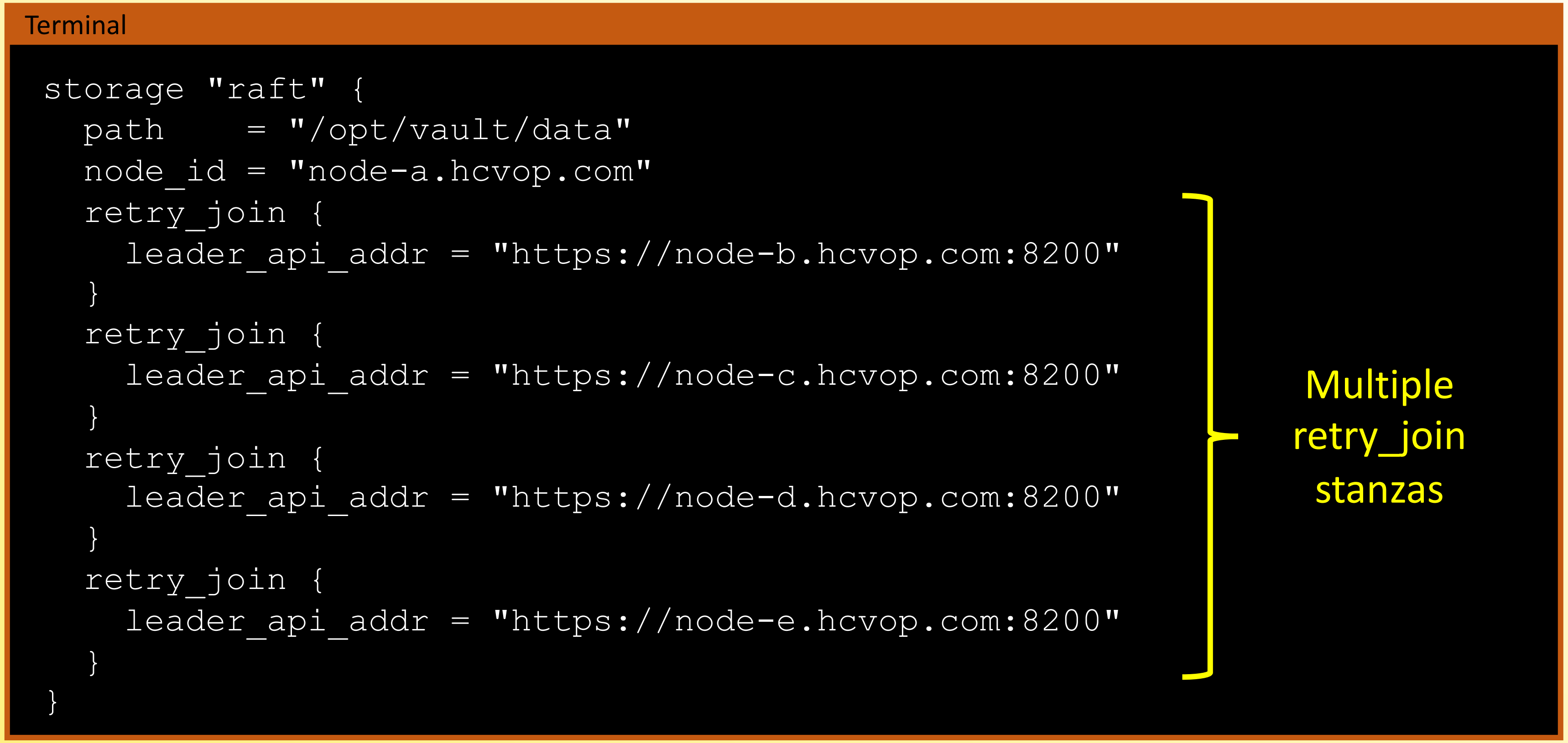


Terminal

```
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    auto_join = "provider=aws region=us-east-1 tag_key=vault tag_value=east-1"
  }
}
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
}
seal "awskms" {
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101",
}
api_addr = "https://vault.hcvop.com:8200"
cluster_addr = " https://node-a.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

Configure Integrated Storage in the Vault Configuration File

Each `retry_join` stanza can include DNS names or IP addresses and the port



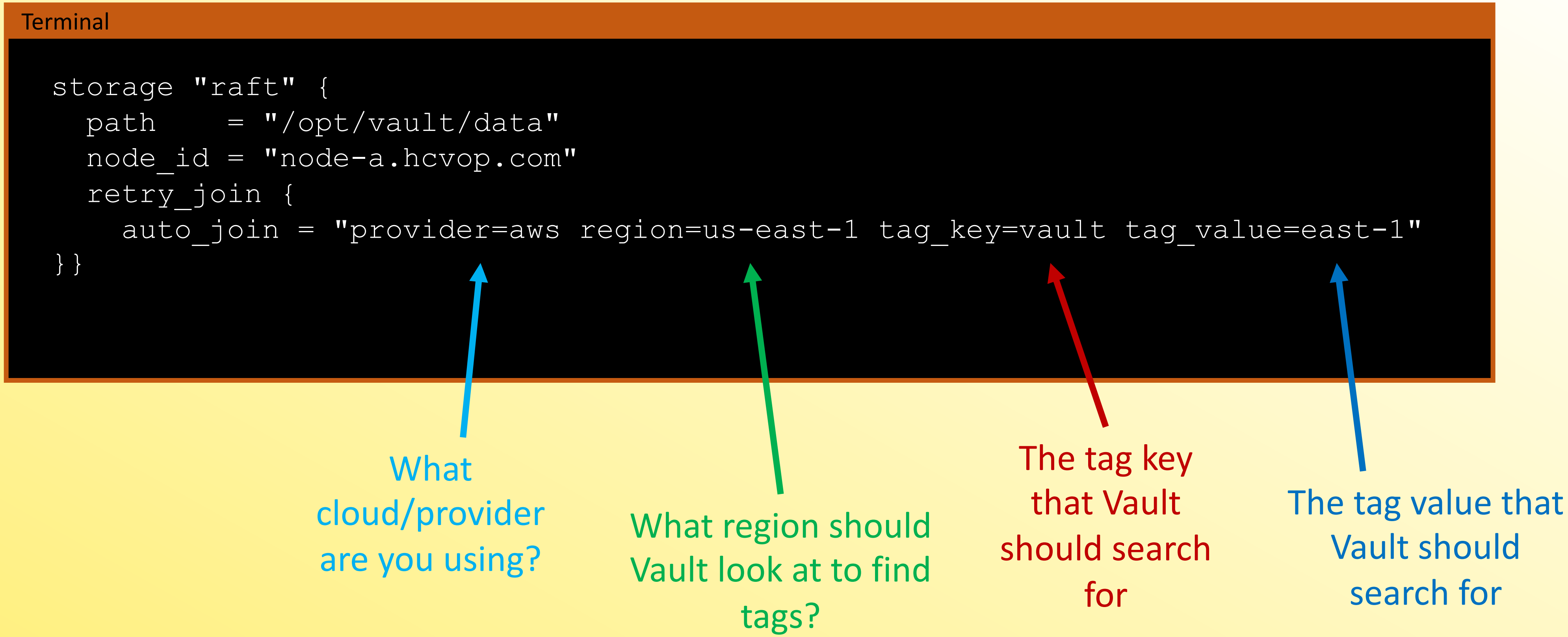
Terminal

```
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    leader_api_addr = "https://node-b.hcvop.com:8200"
  }
  retry_join {
    leader_api_addr = "https://node-c.hcvop.com:8200"
  }
  retry_join {
    leader_api_addr = "https://node-d.hcvop.com:8200"
  }
  retry_join {
    leader_api_addr = "https://node-e.hcvop.com:8200"
  }
}
```

Multiple
retry_join
stanzas

Configure Integrated Storage in the Vault Configuration File

Using `auto_join` to discover other Vault nodes using tags



```
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    auto_join = "provider=aws region=us-east-1 tag_key=vault tag_value=east-1"
}}
```

What cloud/provider are you using?

What region should Vault look at to find tags?

The tag key that Vault should search for

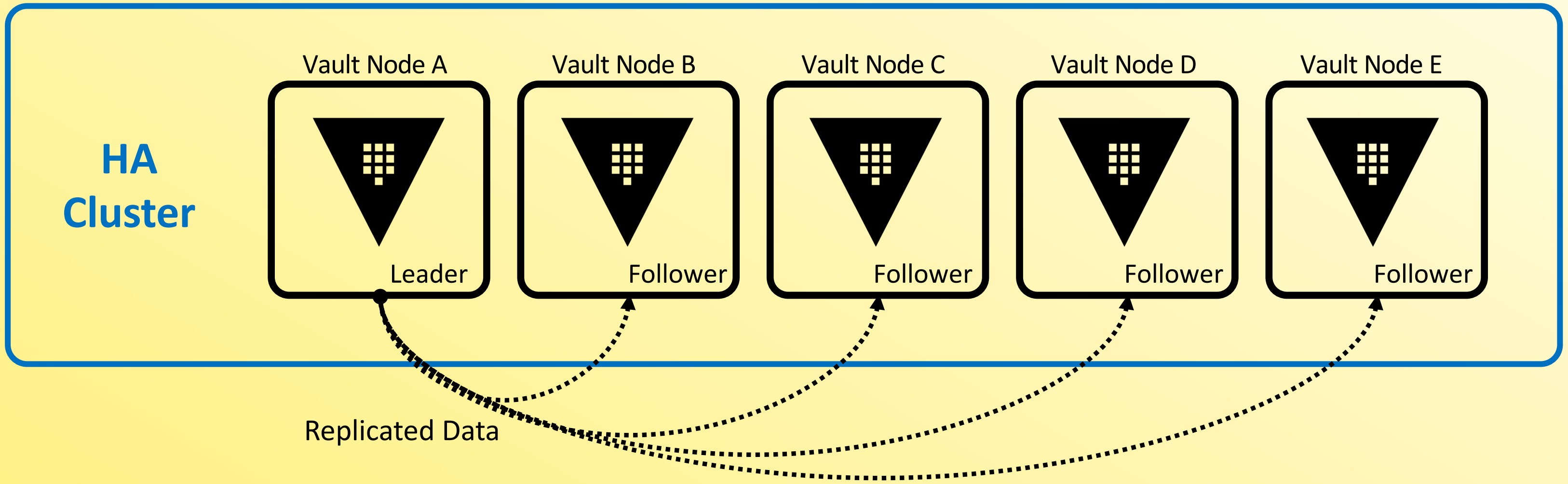
The tag value that Vault should search for

Join Nodes to Form a Cluster

Manually join standby nodes to the cluster using the CLI:

Terminal

```
$ vault operator raft join https://active_node.example.com:8200
```

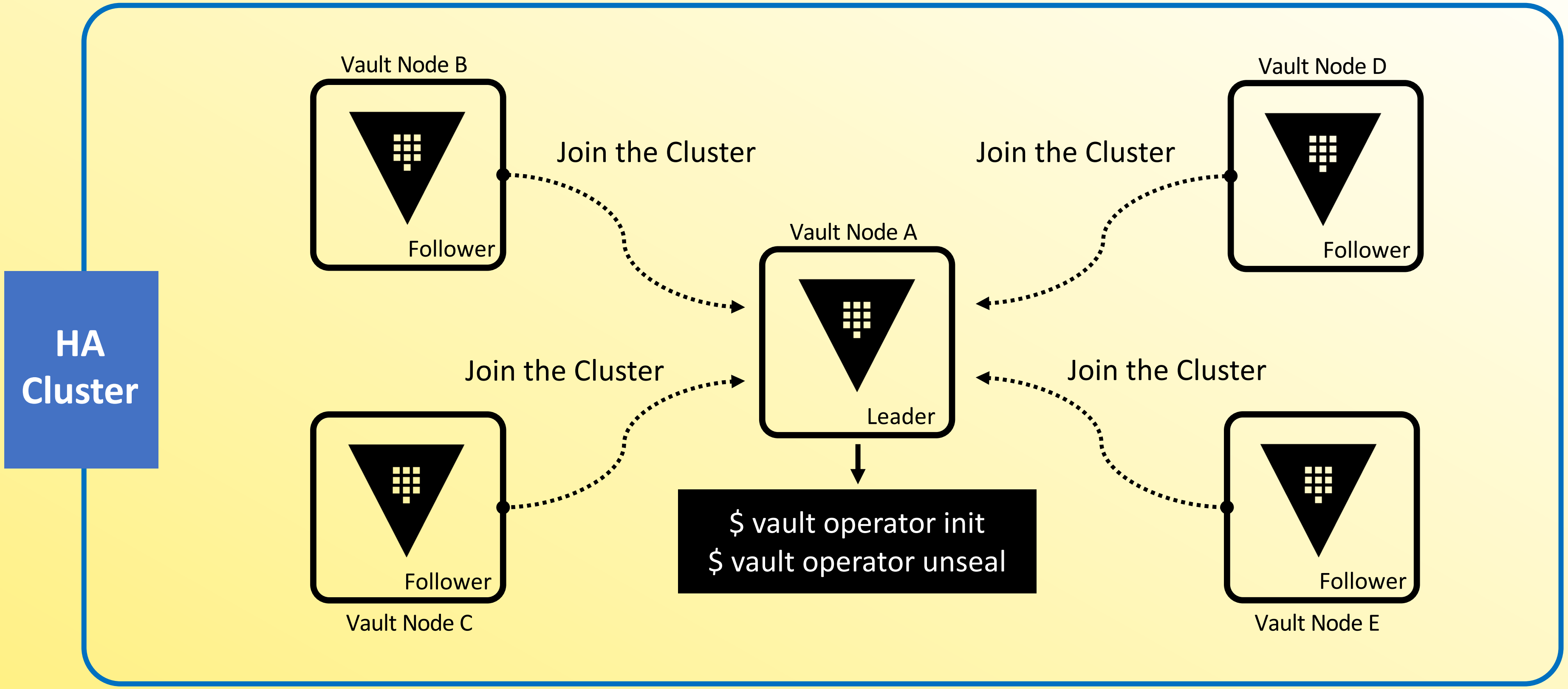


Managing Integrated Storage via CLI

Use the `vault operator raft` command

`list-peers`	Returns the raft cluster member information
`join`	Joins a node to the cluster
`remove-peer`	Removes a node from the cluster
`snapshot`	Restores and saves snapshots from the cluster

Manual Cluster Configuration Workflow

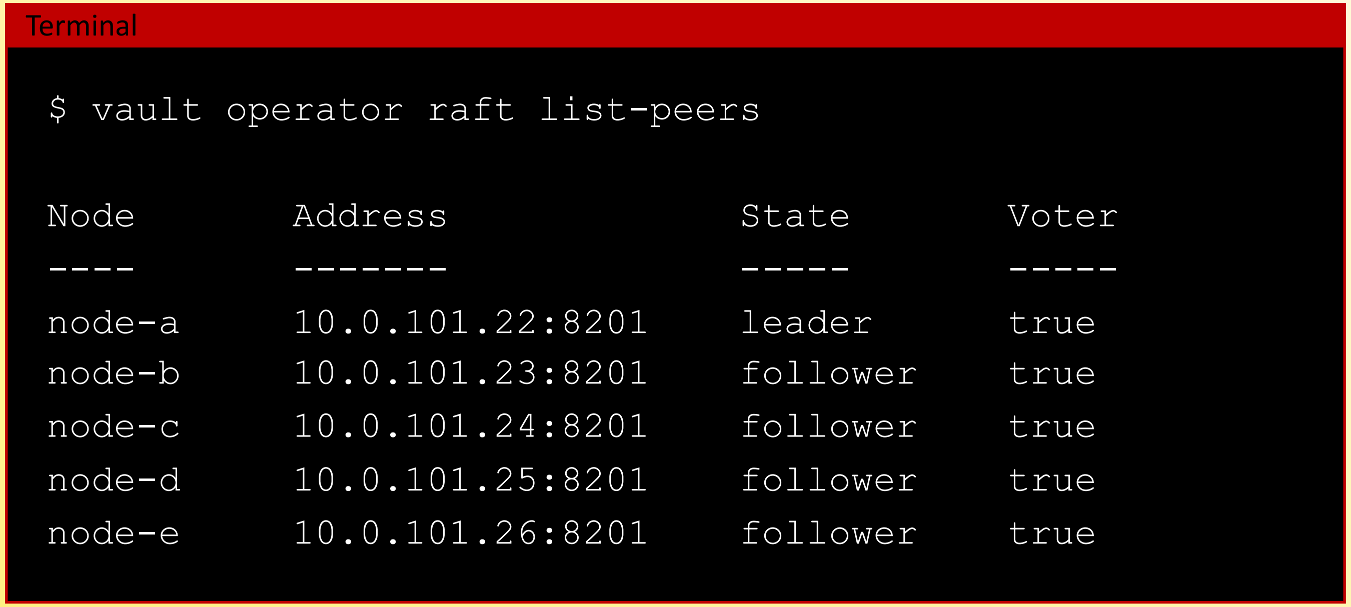


```
$ vault operator init
$ vault operator unseal
```

Viewing Cluster Information

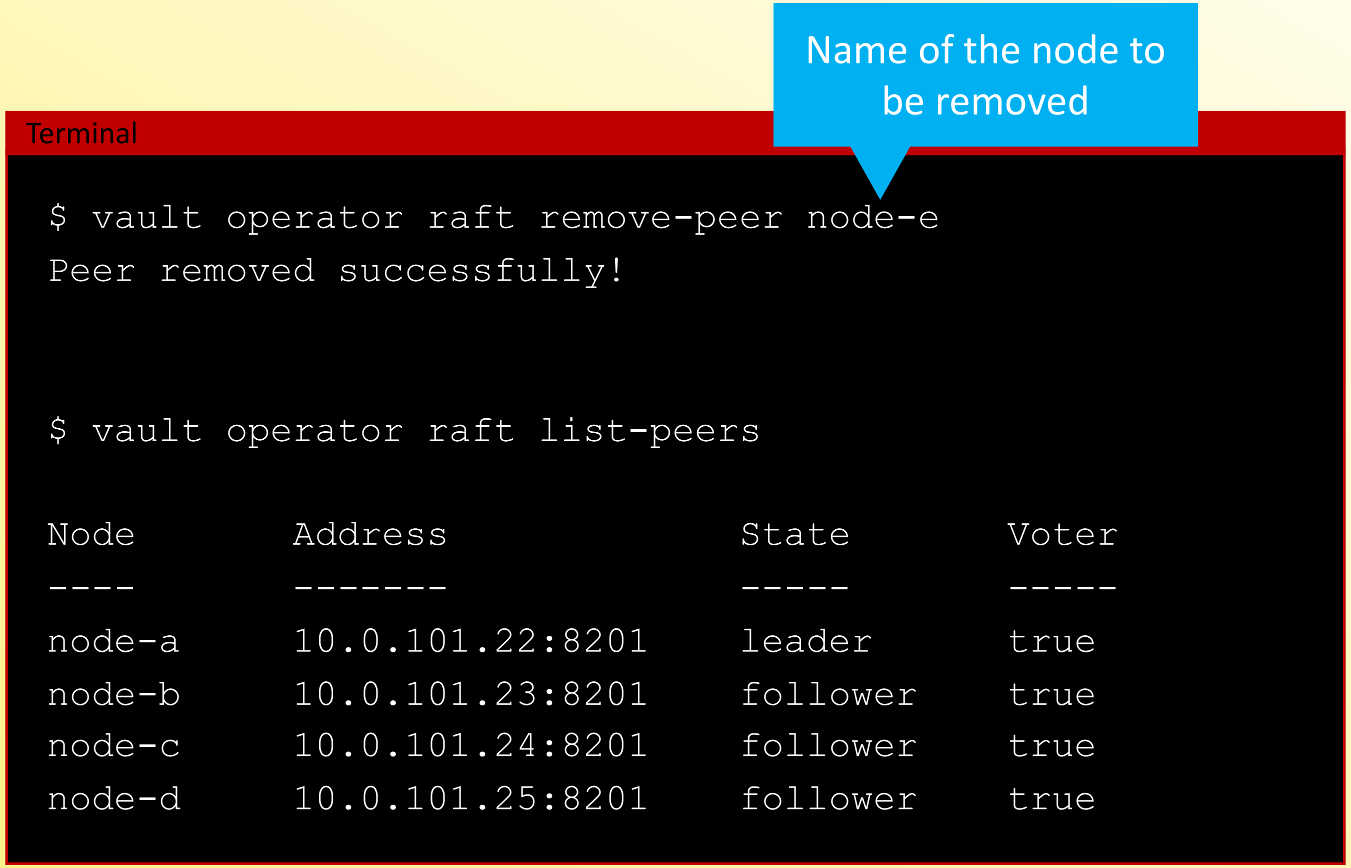
List the cluster members - determine which node is the leader

- Note: You must be authenticated (client token) to run this command



```
$ vault operator raft list-peers

Node       Address              State       Voter
----       -------              -----       -----
node-a     10.0.101.22:8201     leader      true
node-b     10.0.101.23:8201     follower    true
node-c     10.0.101.24:8201     follower    true
node-d     10.0.101.25:8201     follower    true
node-e     10.0.101.26:8201     follower    true
```

Remove a Node from the Cluster



Name of the node to be removed

```
$ vault operator raft remove-peer node-e
Peer removed successfully!


$ vault operator raft list-peers

Node       Address              State       Voter
----       -------              -----       -----
node-a     10.0.101.22:8201     leader      true
node-b     10.0.101.23:8201     follower    true
node-c     10.0.101.24:8201     follower    true
node-d     10.0.101.25:8201     follower    true
```

Vault

CERTIFIED
OPERATIONS
PROFESSIONAL

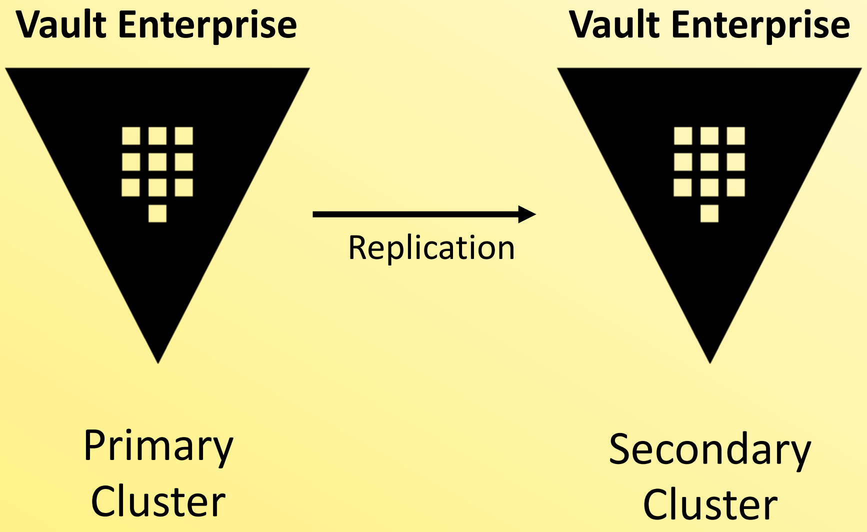
Enable and Configure Disaster Recovery (DR) Replication

What is Vault Replication?

Organizations usually have infrastructure that **spans multiple datacenters**

- Vault needs to be highly-available for application access
- Needs to scale as organizations continue to add use cases and apps
- Common set of policies that are enforced **globally**
- Consistent set of secrets and configurations available to applications that need them regardless of data center

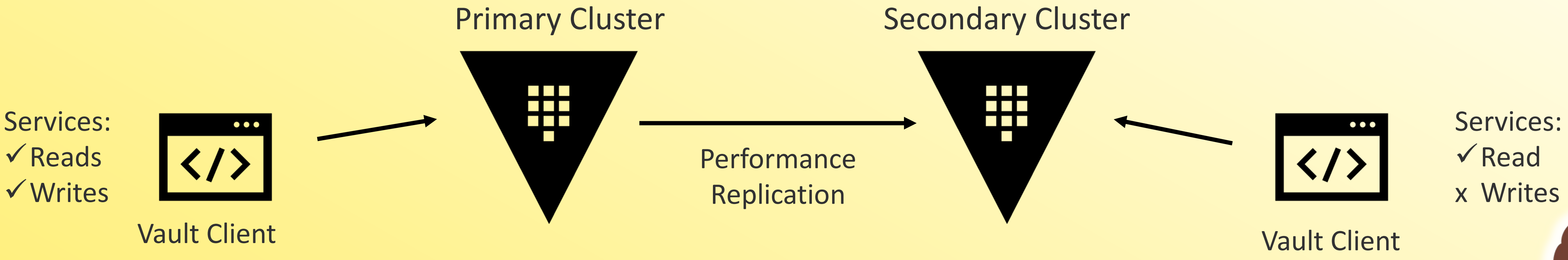
What is Vault Replication?



- **Only available in Vault Enterprise**
- Replication operates on a leader-follower model (**primaries** and **secondaries**)
- The primary cluster acts as the system of record and replicates most Vault data asynchronously
- All communication between primaries and secondaries is **end-to-end encrypted** with mutually-authenticated TLS sessions

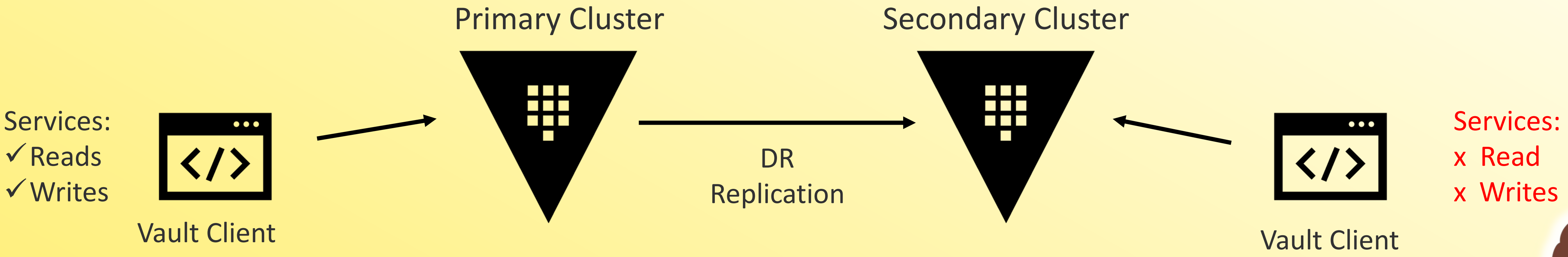
Performance Replication

- Replicates the underlying configuration, policies, and other data
- Ability to service reads from client requests
- Clients will authenticate to the performance replicated cluster separately
- Does not replicate tokens or leases to performance secondaries

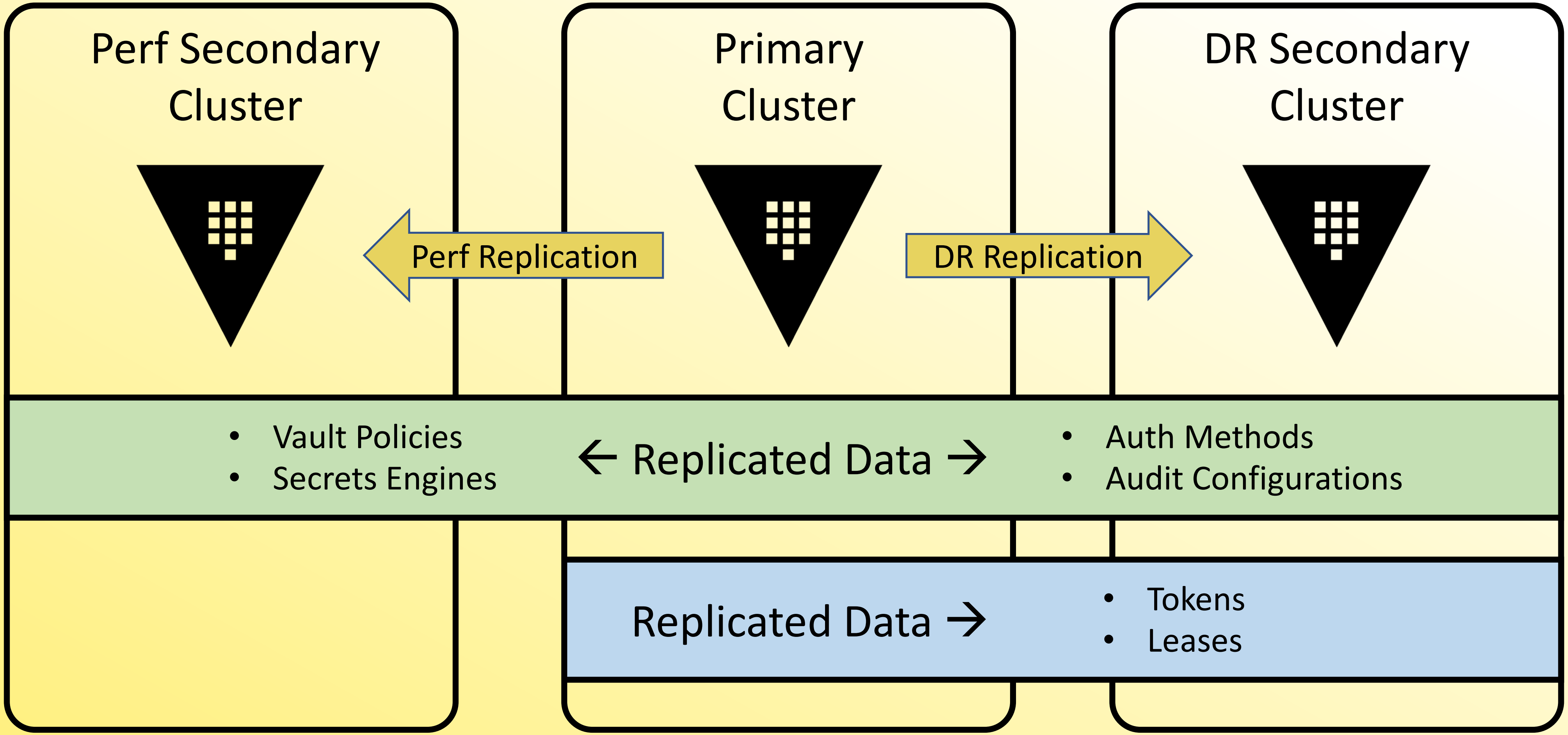


Disaster Recovery Replication

- Replicates the underlying configuration, policies, and all other data
- Cannot service reads from client requests
- Clients should authenticate with the primary cluster only (or a perf cluster)
- Will replicate tokens and leases created on the primary cluster



Comparison



- Vault Policies
- Secrets Engines

← Replicated Data →

- Auth Methods
- Audit Configurations

Replicated Data →

- Tokens
- Leases

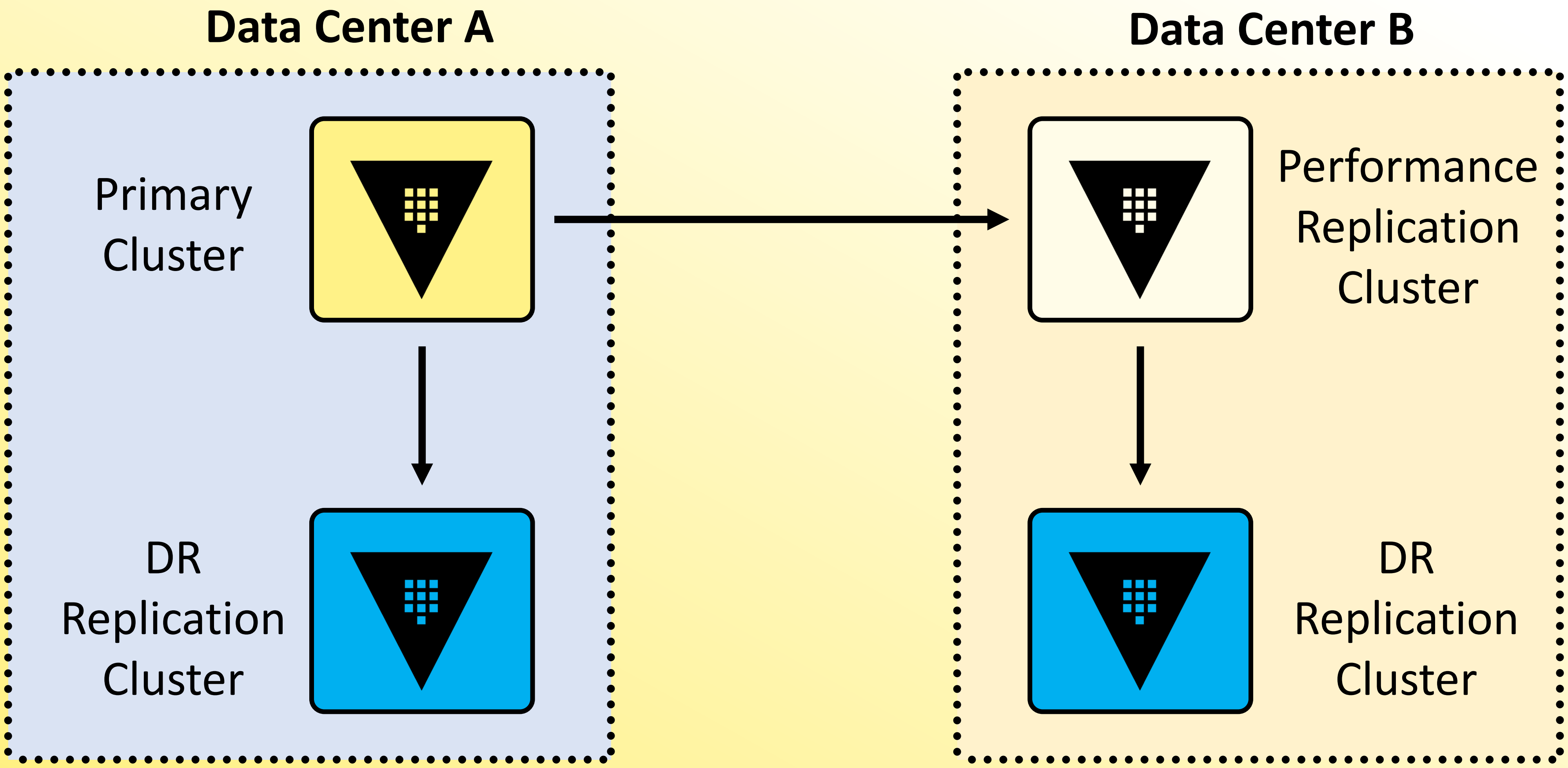
Disaster Recovery Replication

- Provides a warm-standby cluster where EVERYTHING is replicated to the DR secondary cluster(s)

- DR clusters DO NOT respond to clients unless they are promoted to a primary cluster

- Even as an admin or using a root token, most paths on a secondary cluster are disabled, meaning you can't do much of anything on a DR cluster

Replication Architecture



Data Center A

Primary Cluster

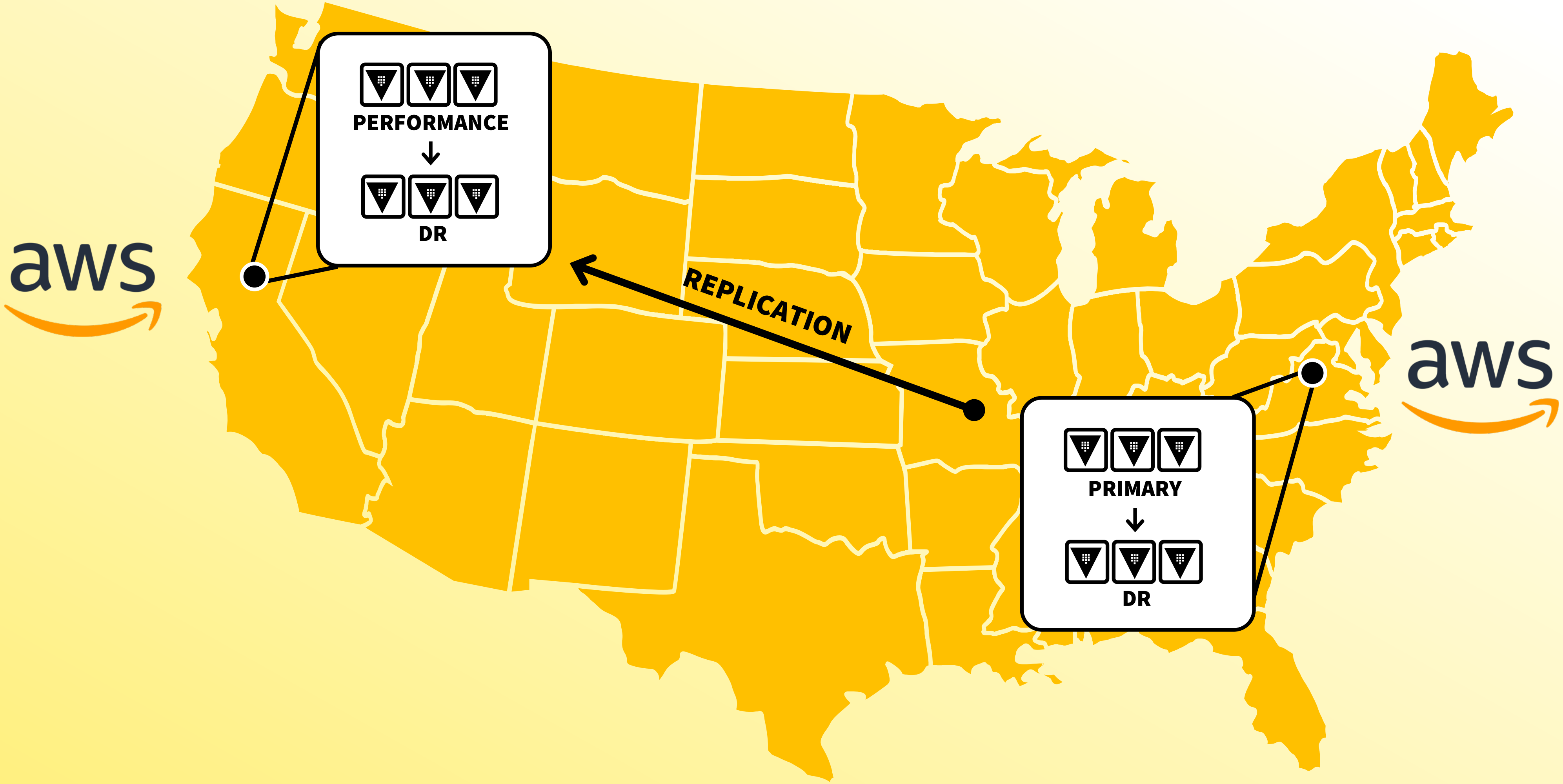
DR Replication Cluster

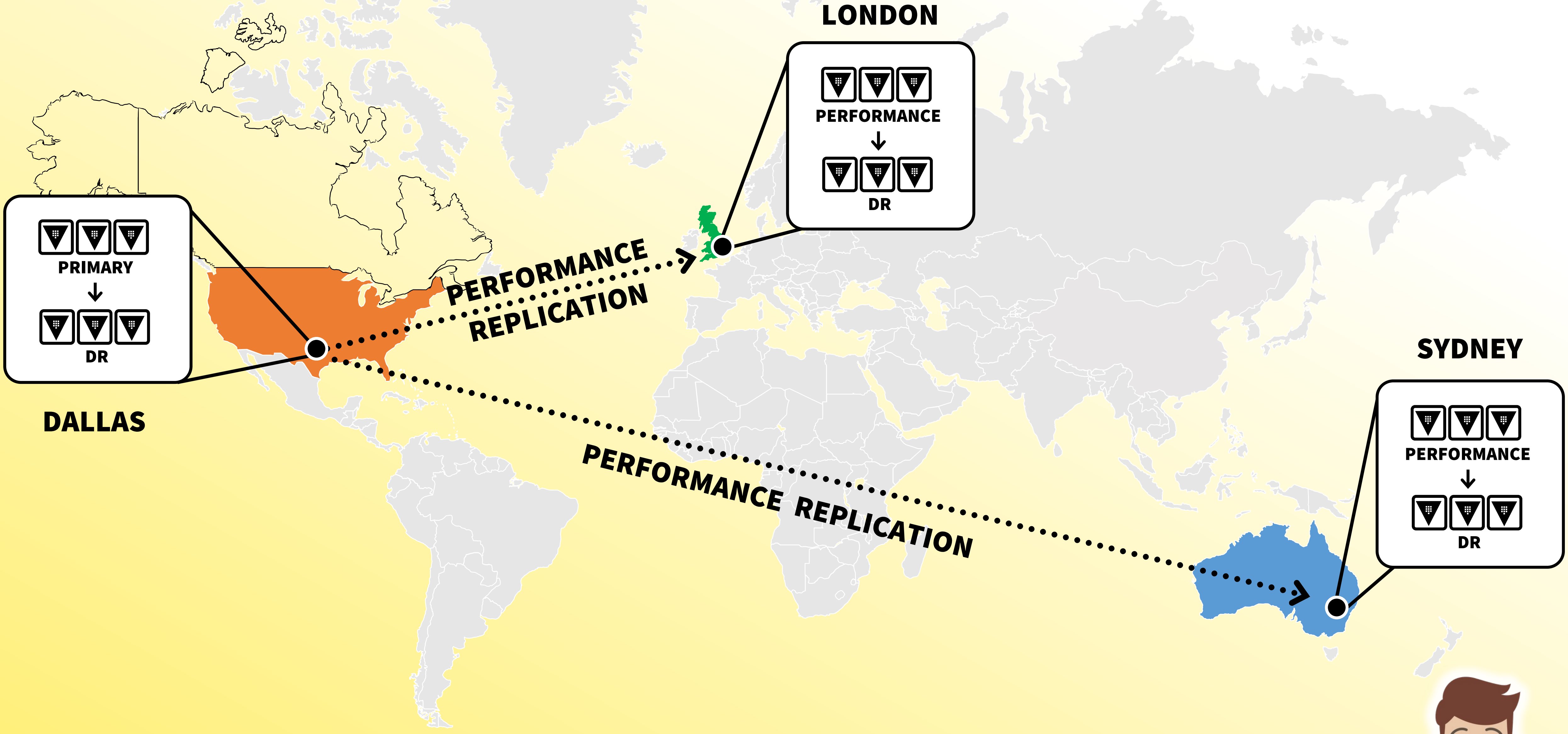
Data Center B

Performance Replication Cluster

DR Replication Cluster

Replication Architecture




LONDON

PERFORMANCE

↓

DR

DALLAS

PRIMARY

↓

DR

PERFORMANCE REPLICATION

PERFORMANCE REPLICATION

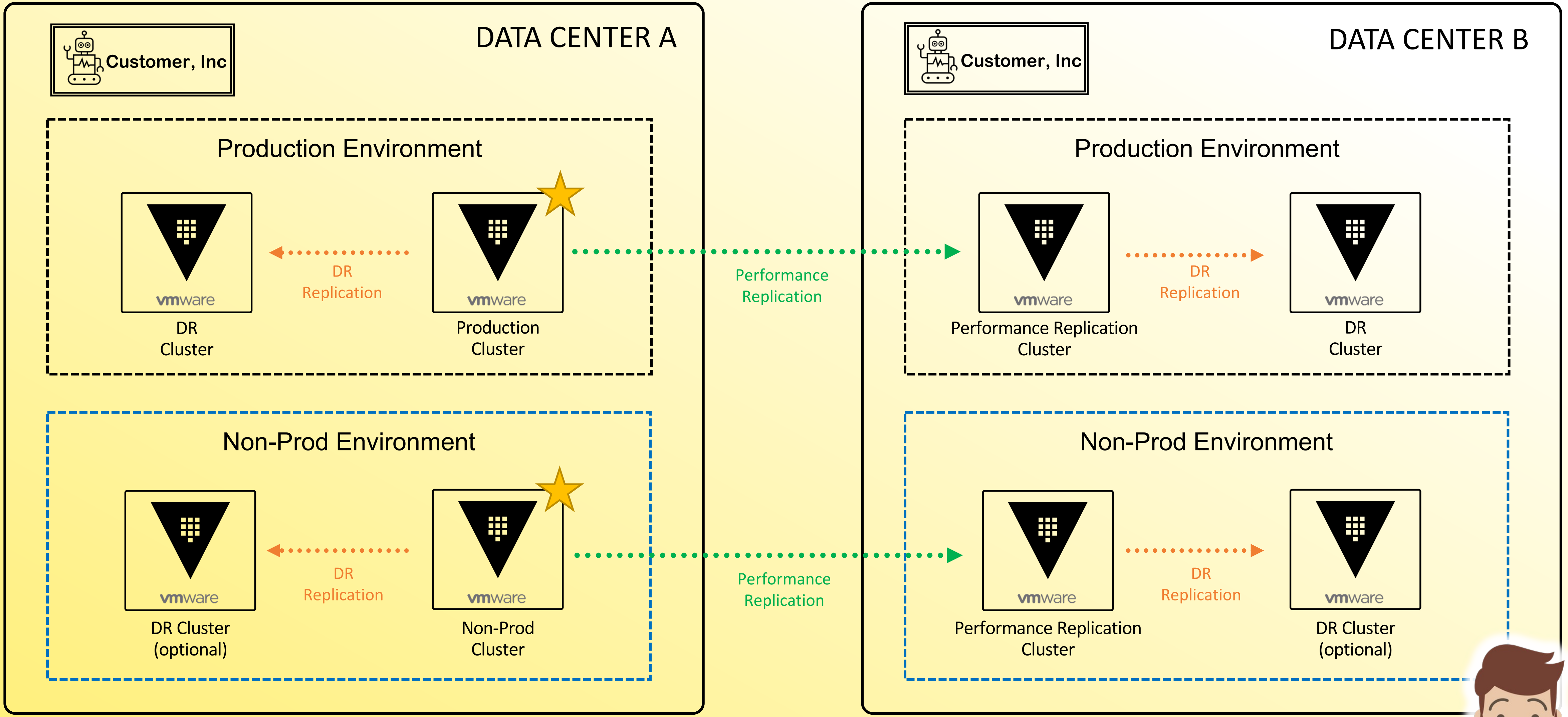
SYDNEY

PERFORMANCE

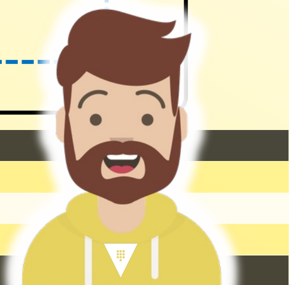
↓

DR

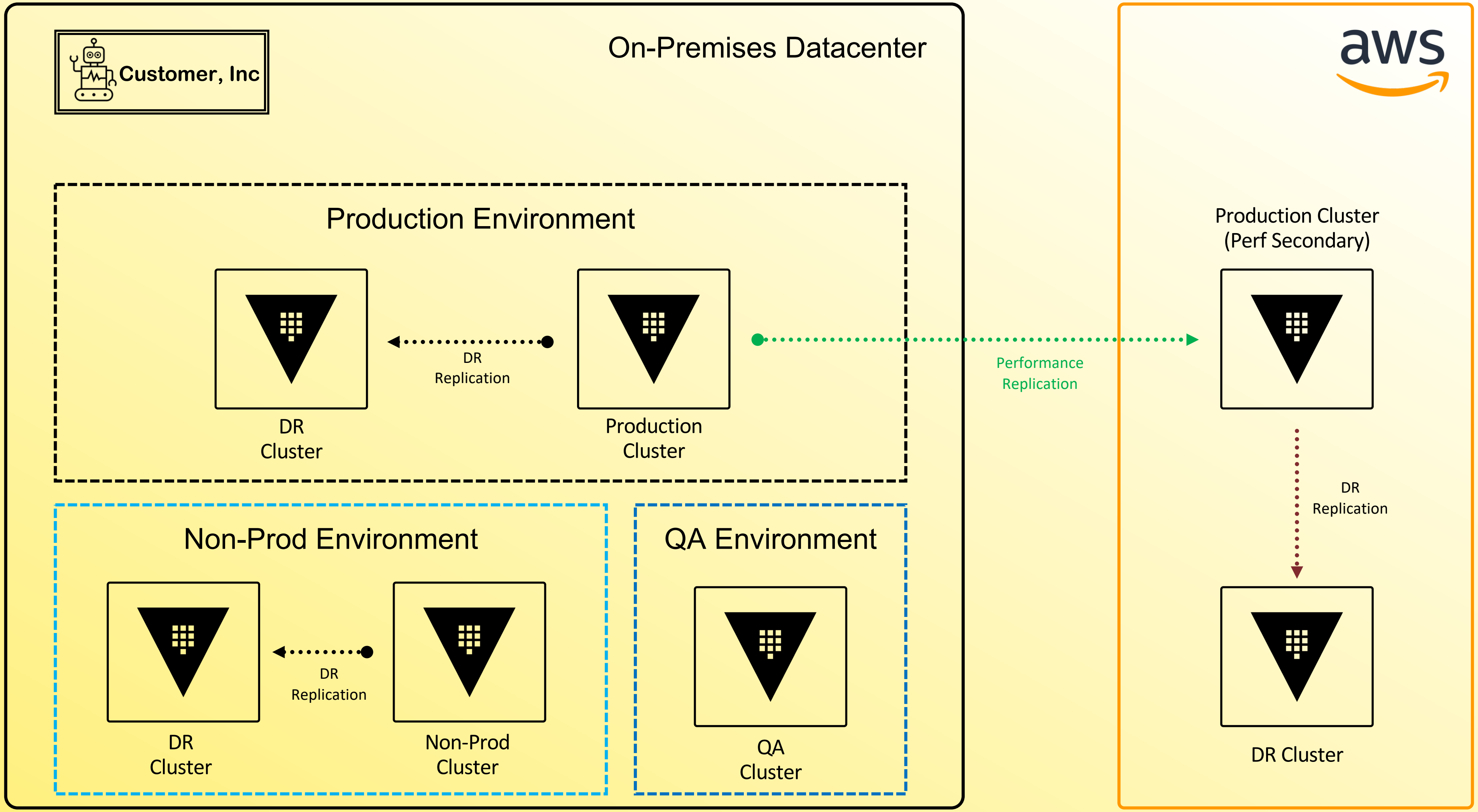
Real-World Customer Example



★ Primary Cluster in Replica Set

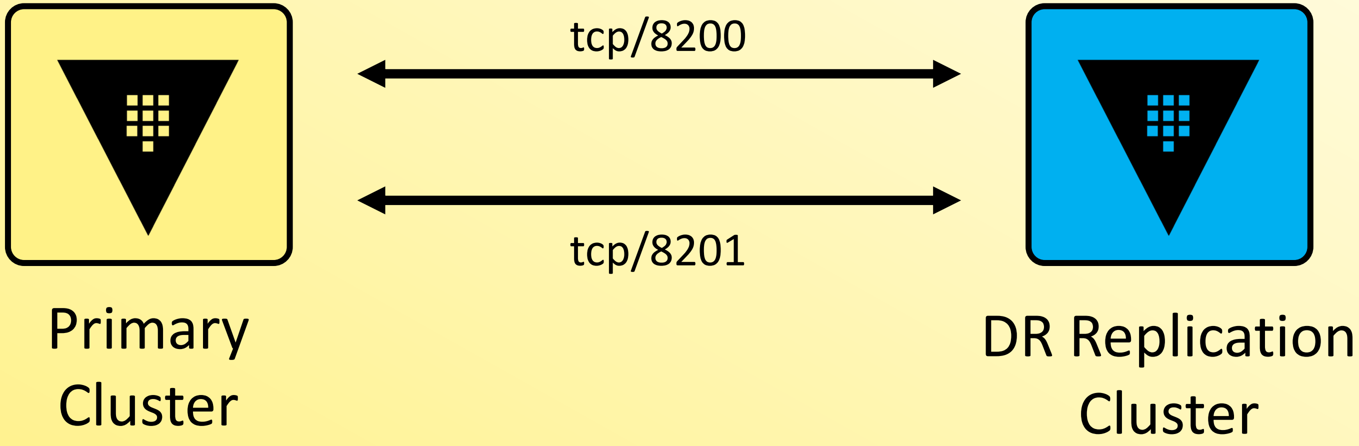


Real-World Customer Example



Networking Requirements

- Communication between clusters must be permitted to allow replication, RPC forwarding, and cluster bootstrapping to work as expected.
- If using DNS, each cluster must be able to resolve the name of the other cluster



Networking Ports

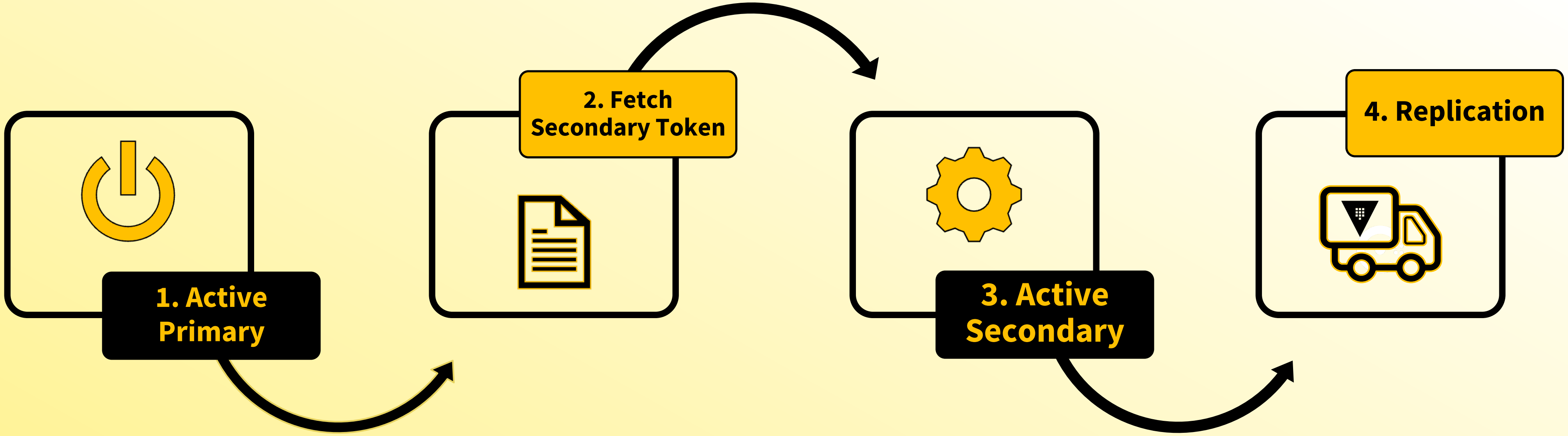
Vault
CERTIFIED
OPERATIONS
PROFESSIONAL

Source	Destination	Port	Protocol	Direction	Purpose
Client Machines	Load balancer	443	tcp	Incoming	Request distribution
Load Balancer	Vault Servers	8200	tcp	Incoming	Vault API
Vault Servers	**Vault Servers**	**8200**	**tcp**	**Bidirectional**	**Cluster bootstrapping**
Vault Servers	**Vault Servers**	**8201**	**tcp**	**Bidirectional**	**Raft, replication, request forwarding**
Vault Servers	External Systems	Various	Various	Various	External APIs

How Do We Set All of this Up?

Vault

CERTIFIED
OPERATIONS
PROFESSIONAL



1. Active Primary

2. Fetch Secondary Token

3. Active Secondary

4. Replication

Activate DR Replication on the Primary as a DR Primary

Create a secondary token on the Primary cluster

Activate DR Replication on the Secondary cluster as a DR secondary

Watch Vault replicated the data from the Primary to the new Secondary cluster

Activating DR Replication

- Replication is NOT enabled by default, so you must enable it on each cluster that will participate in the replica set

- Enables an internal root CA on the primary Vault cluster - creates a root certificate and client cert

- Vault creates a mutual TLS connection between the nodes using self-signed certificates and keys from the internal CA – *NOT the same TLS configured for the listener*

  - If Vault sits behind a load balancer which is terminating TLS, it will break the mutual TLS between the nodes if inter-cluster traffic is forced through the load balancer

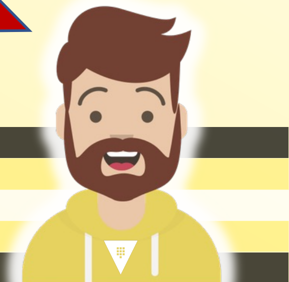
Secondary Token

- A secondary token is required to permit a secondary cluster to replicate from the primary cluster
- Due to its sensitivity, the secondary token is protected with response wrapping
- Multiple people should “have eyes” on the secondary token once it’s been issued until it is submitted to the secondary cluster
- Once the token is successfully used, it is useless (single-use token)
- The secondary token includes information such as:
  - The redirect address of the primary cluster
  - The client certificate and CA certificate

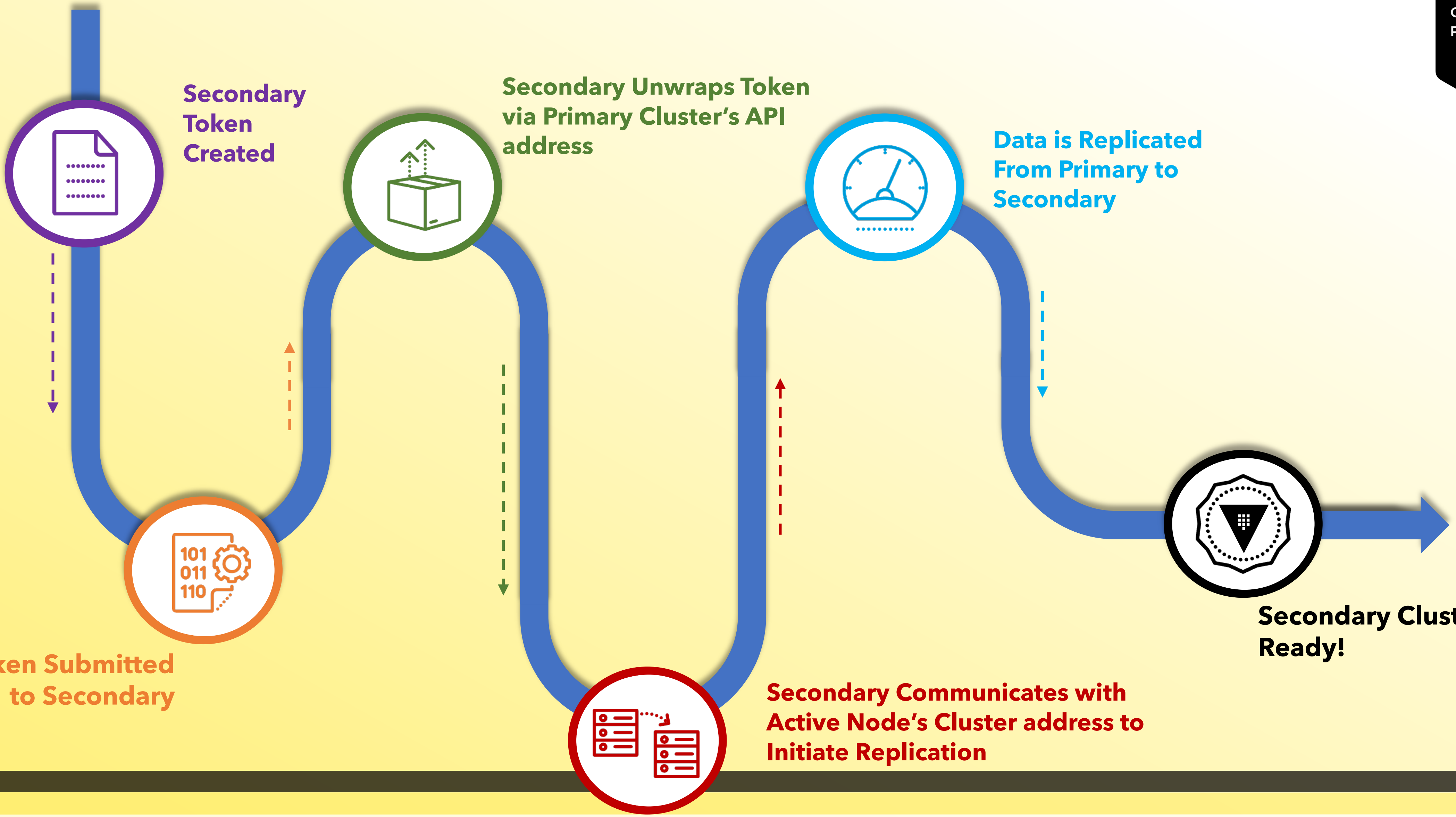
Secondary Token - Unwrapped

```
{
 "request_id": "98d4c7a5-0f00-4872-1cad-6ab8fa35694c",
 "lease_id": "",
 "lease_duration": 0,
 "renewable": false,
 "data": {
  "ca_cert":
"MIICfjCCAd+gAwIBAgIIVQciUMO14jswCgYIKoZIzj0EAwQwMzExMC8GA1UEAxMocmVwLTA3MzQyYTBiLWJhZjktNTRhZC00MjcyLWVlZTE0NTFmMGQyNDAgFw0yMjA1MjMxNzMxMTlaGA8yMDUyMD
UyMzA1MzE0OVowMzExMC8GA1UEAxMocmVwLTA3MzQyYTBiLWJhZjktNTRhZC00MjcyLWVlZTE0NTFmMGQyNDCBmzAQBgcqhkjOPQIB…",
  "client_cert":
"MIICZjCCAcigAwIBAgIIKW4DvMJIDt4wCgYIKoZIzj0EAwQwMzExMC8GA1UEAxMocmVwLTA3MzQyYTBiLWJhZjktNTRhZC00MjcyLWVlZTE0NTFmMGQyNDAgFw0yMjA1MjMxNzMzMjNaGA8yMDUyMD
UyMzA1MzM1M1owLzEtMCsGA1UEAxMkZjYwNmEwMGItMTA0Ny05…",
  "client_key": {
   "d":
1000631355517086513122196214347690053058610203119167515956358237211447177696212705845570913960352147412040118660857971566143956149412938809960381549100740826,
   "type": "p521",
   "x":
6585241467240384151398124142600469244382875941120587428008118368573328804955608918211668669530795701495917170318651699823329298690163971349362335317686304875,
   "y":
4563340717429320656179725289836652789047992587356159319649284729225610938283331963913484853756937351659805499727826936061640752374496368580488067455136501717
  },
  "cluster_id": "0d127970-99ce-152f-0311-3b081d1264d3",
  "encrypted_client_key": null,
  "id": "secondary",
  "mode": 512,
  "nonce": null,
  "primary_cluster_addr": "https://vault-pri.hcvop.com:8201",
  "primary_public_key": null
 },
 "warnings": null
}
```

This is not a normal thing you would do. I simply did it to show you what information the secondary token included



How is the Secondary Token Used?



Secondary Token Created

Token Submitted to Secondary

Secondary Unwraps Token via Primary Cluster's API address

Secondary Communicates with Active Node's Cluster address to Initiate Replication

Data is Replicated From Primary to Secondary

Secondary Cluster Ready!

Configure Replication on the CLI

1 Activate DR Replication

```
primary$ vault write -f sys/replication/dr/primary/enable
```

2 Create the Secondary Token

```
primary$ vault write sys/replication/dr/primary/secondary-token id=<id>
```

Name it what you want

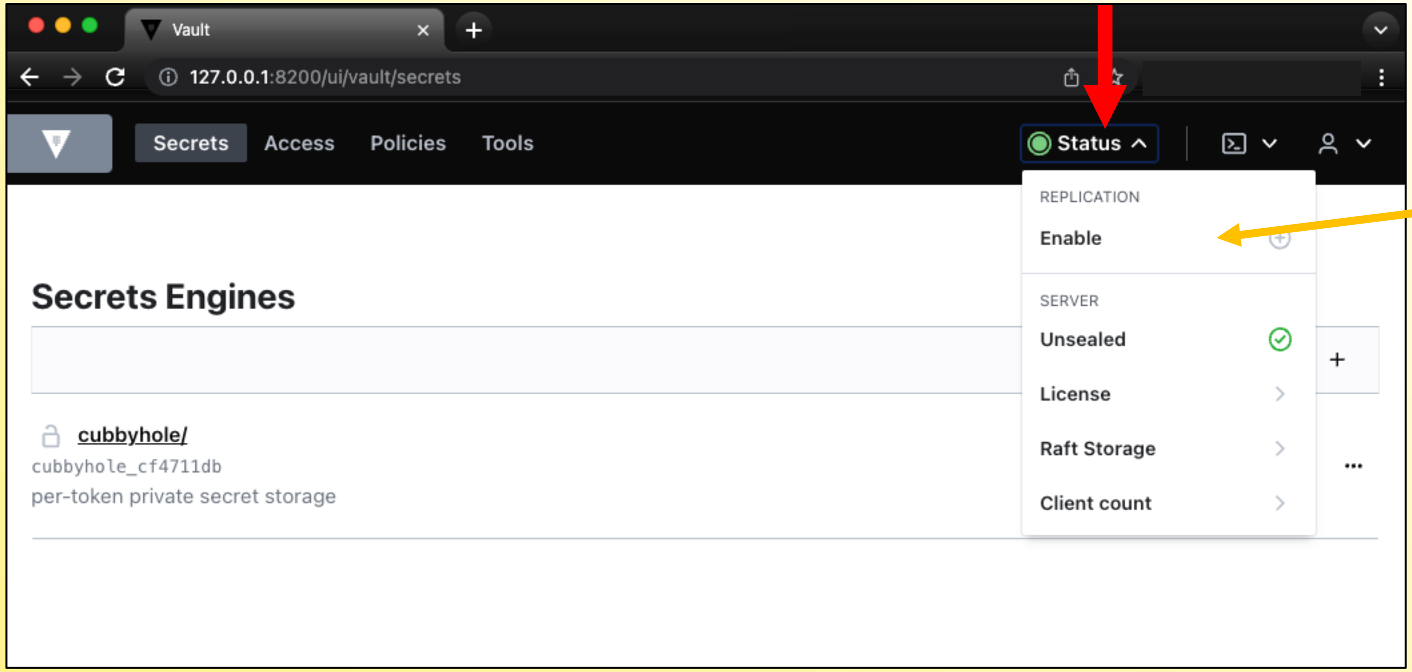
3 Activate the Secondary Cluster

```
secondary$ vault write sys/replication/dr/secondary/enable token=<token>
```

Provide token from primary cluster (command above)

Configure Replication using the UI

Enable Replication on Primary

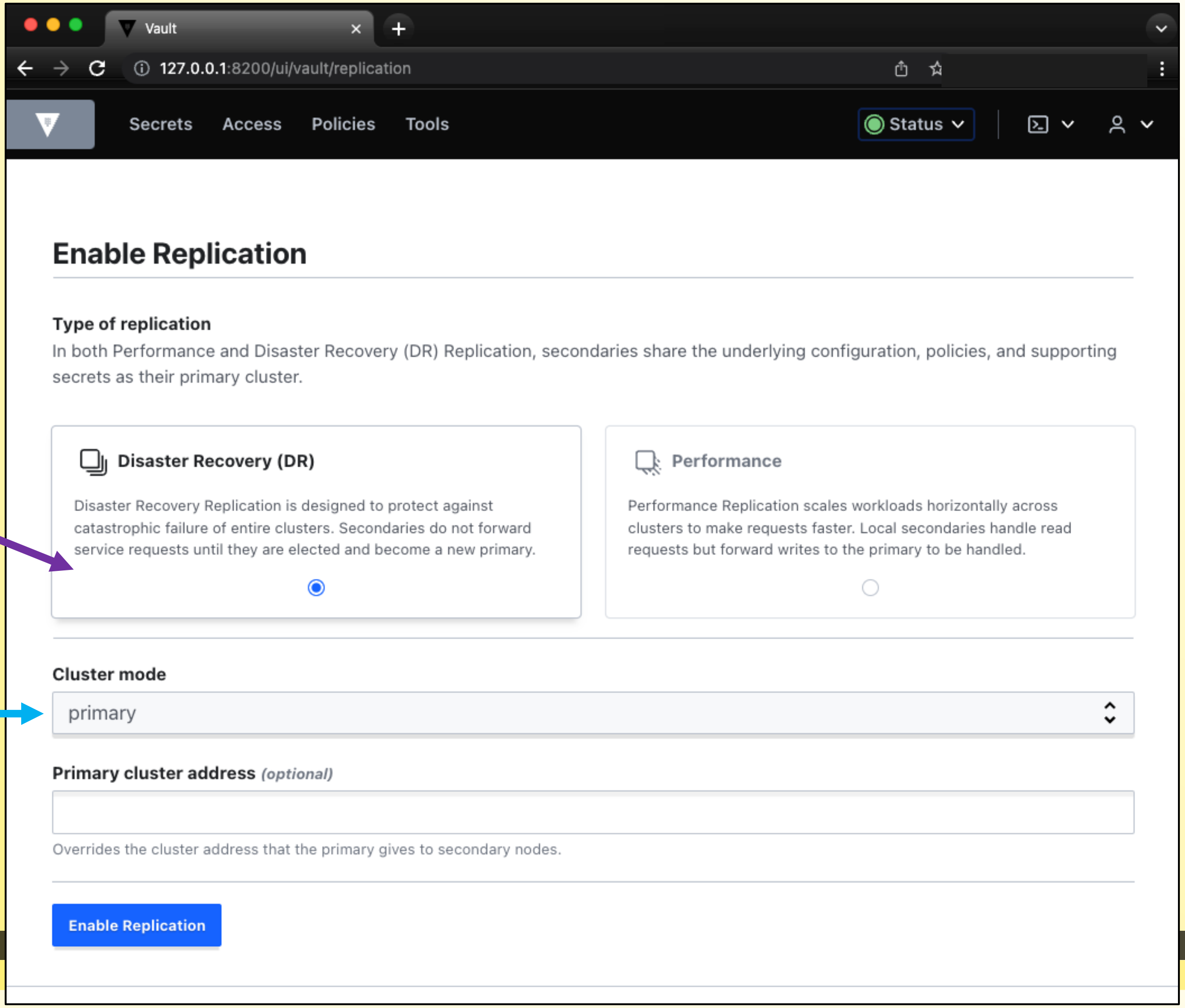


Enable Replication

Configure Replication using the UI

Select Type and Mode on Primary

Vault CERTIFIED OPERATIONS PROFESSIONAL



Enable Replication

Type of replication

In both Performance and Disaster Recovery (DR) Replication, secondaries share the underlying configuration, policies, and supporting secrets as their primary cluster.

Disaster Recovery (DR)

Disaster Recovery Replication is designed to protect against catastrophic failure of entire clusters. Secondaries do not forward service requests until they are elected and become a new primary.

●

Performance

Performance Replication scales workloads horizontally across clusters to make requests faster. Local secondaries handle read requests but forward writes to the primary to be handled.

○

Cluster mode

primary

Primary cluster address *(optional)*

Overrides the cluster address that the primary gives to secondary nodes.

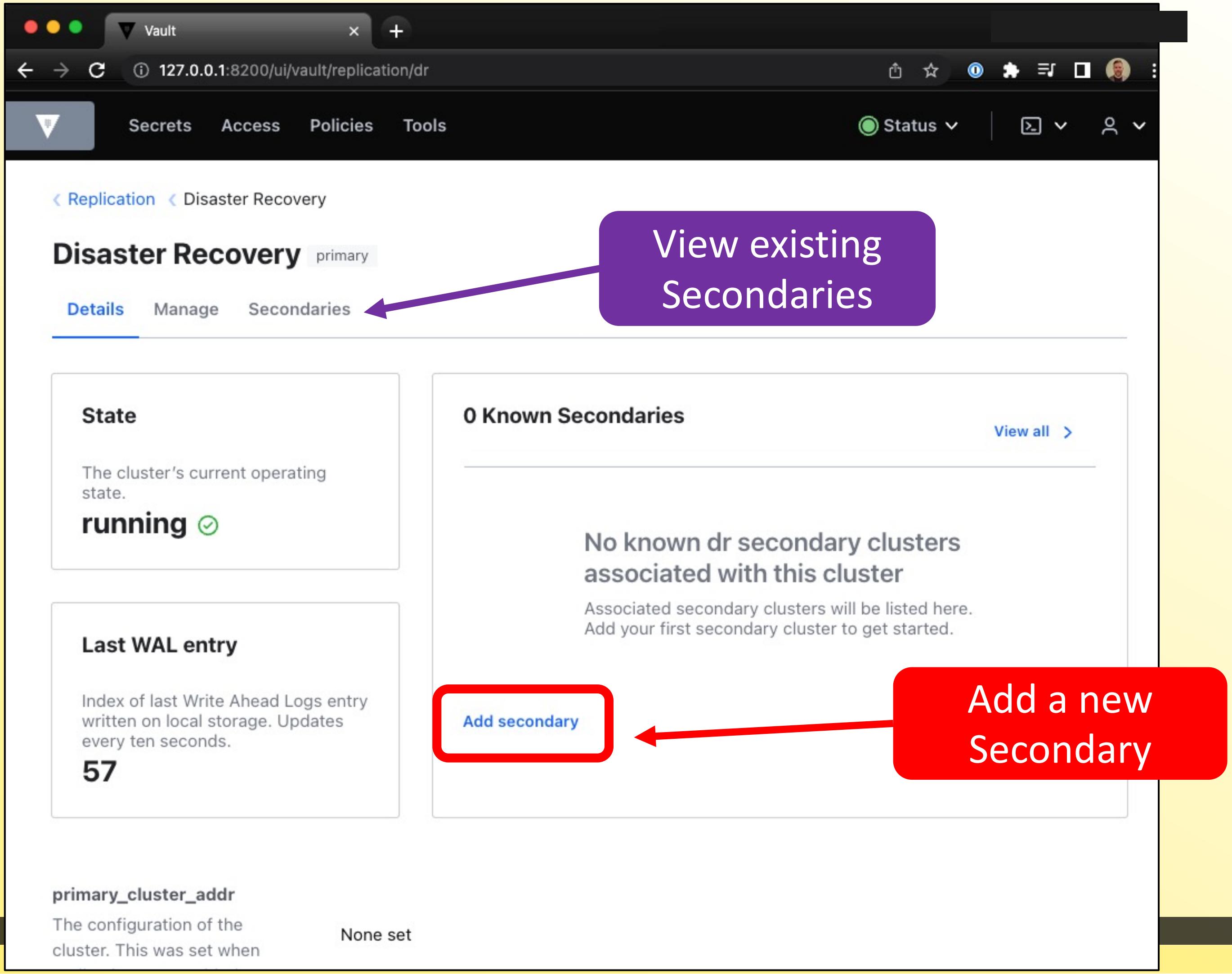
Enable Replication

Select DR Replication

Choose Primary for the DR Primary cluster

Configure Replication using the UI

Add a Secondary



Replication < Disaster Recovery

Disaster Recovery primary

Details | Manage | Secondaries

State

The cluster's current operating state.

running ✓

Last WAL entry

Index of last Write Ahead Logs entry written on local storage. Updates every ten seconds.

57

0 Known Secondaries View all >

No known dr secondary clusters associated with this cluster

Associated secondary clusters will be listed here. Add your first secondary cluster to get started.

Add secondary

primary_cluster_addr

The configuration of the cluster. This was set when

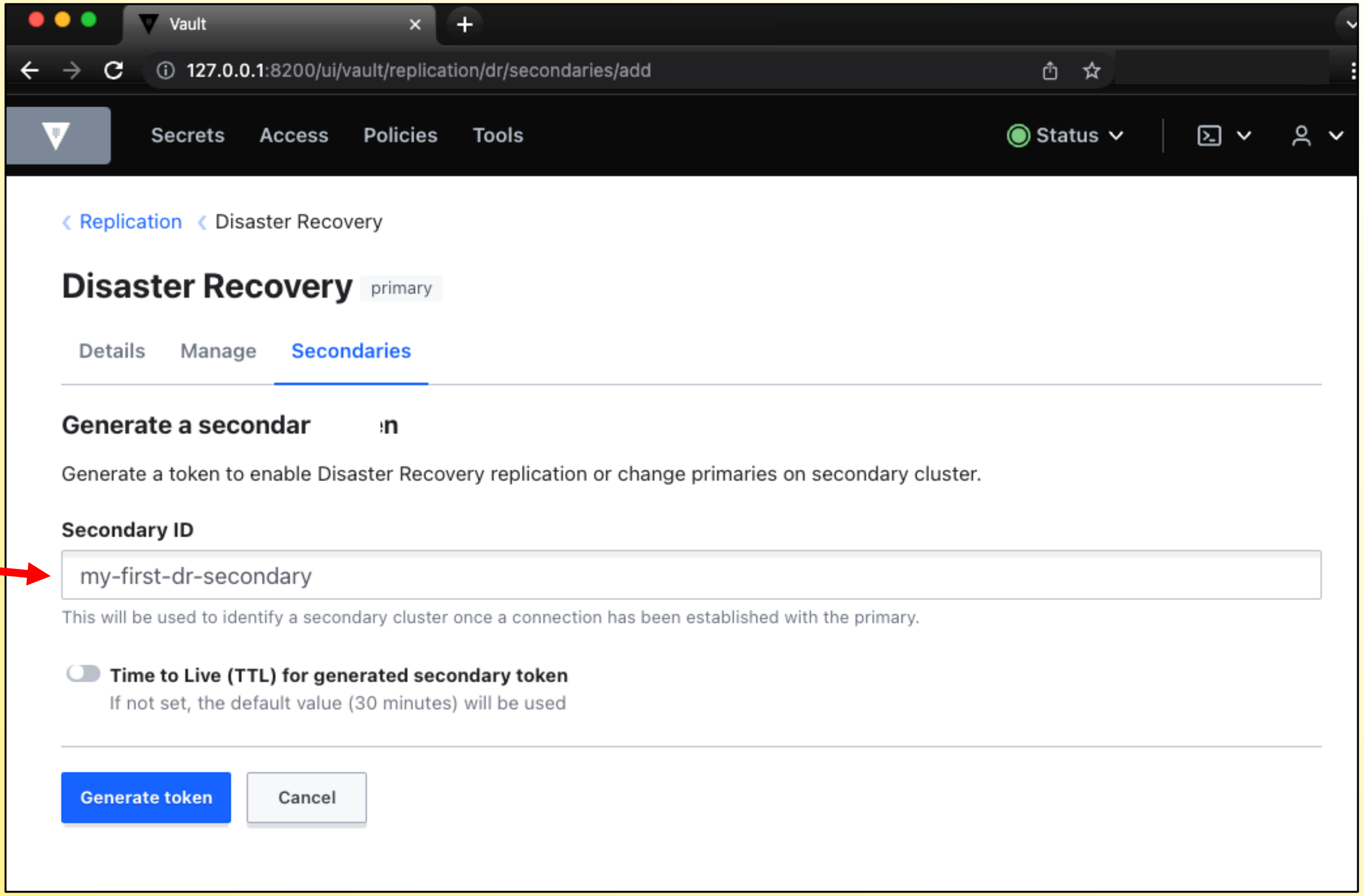
None set

View existing Secondaries

Add a new Secondary

Configure Replication using the UI

Name Secondary and Get Secondary Token



Disaster Recovery primary

Generate a secondar n

Generate a token to enable Disaster Recovery replication or change primaries on secondary cluster.

Secondary ID: my-first-dr-secondary

This will be used to identify a secondary cluster once a connection has been established with the primary.

Time to Live (TTL) for generated secondary token

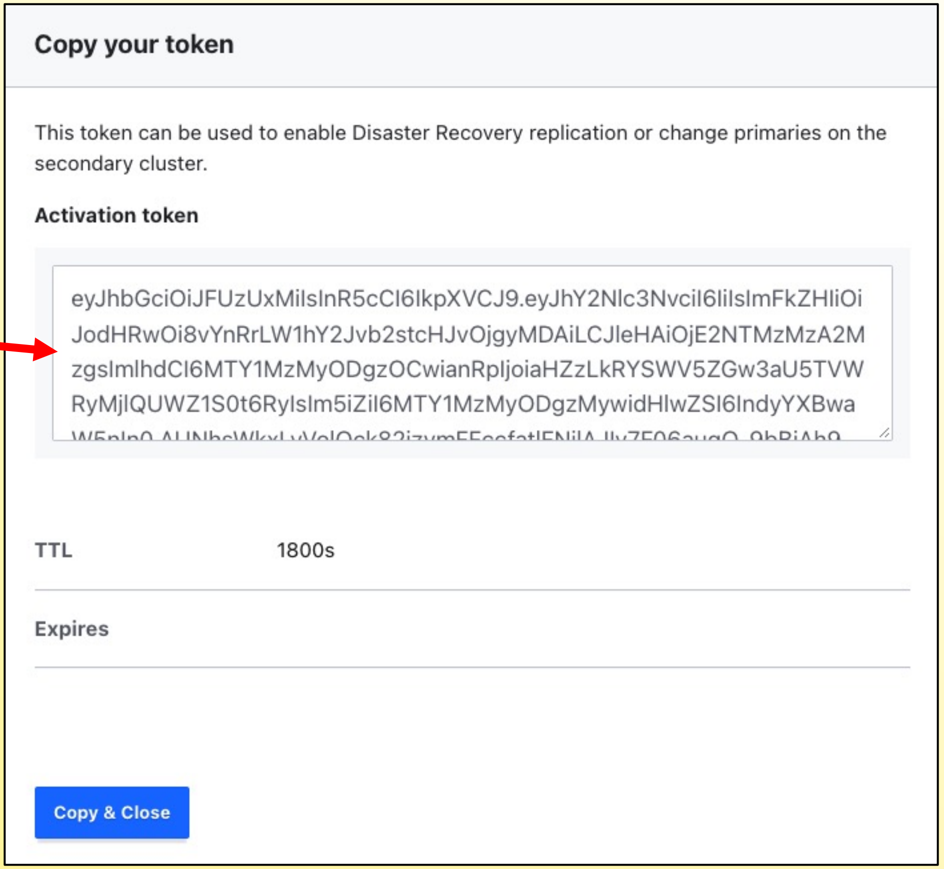
If not set, the default value (30 minutes) will be used

Give it a Name

Configure Replication using the UI

Copy New Secondary Token from Primary Cluster

Secondary Token



Copy your token

This token can be used to enable Disaster Recovery replication or change primaries on the secondary cluster.

Activation token

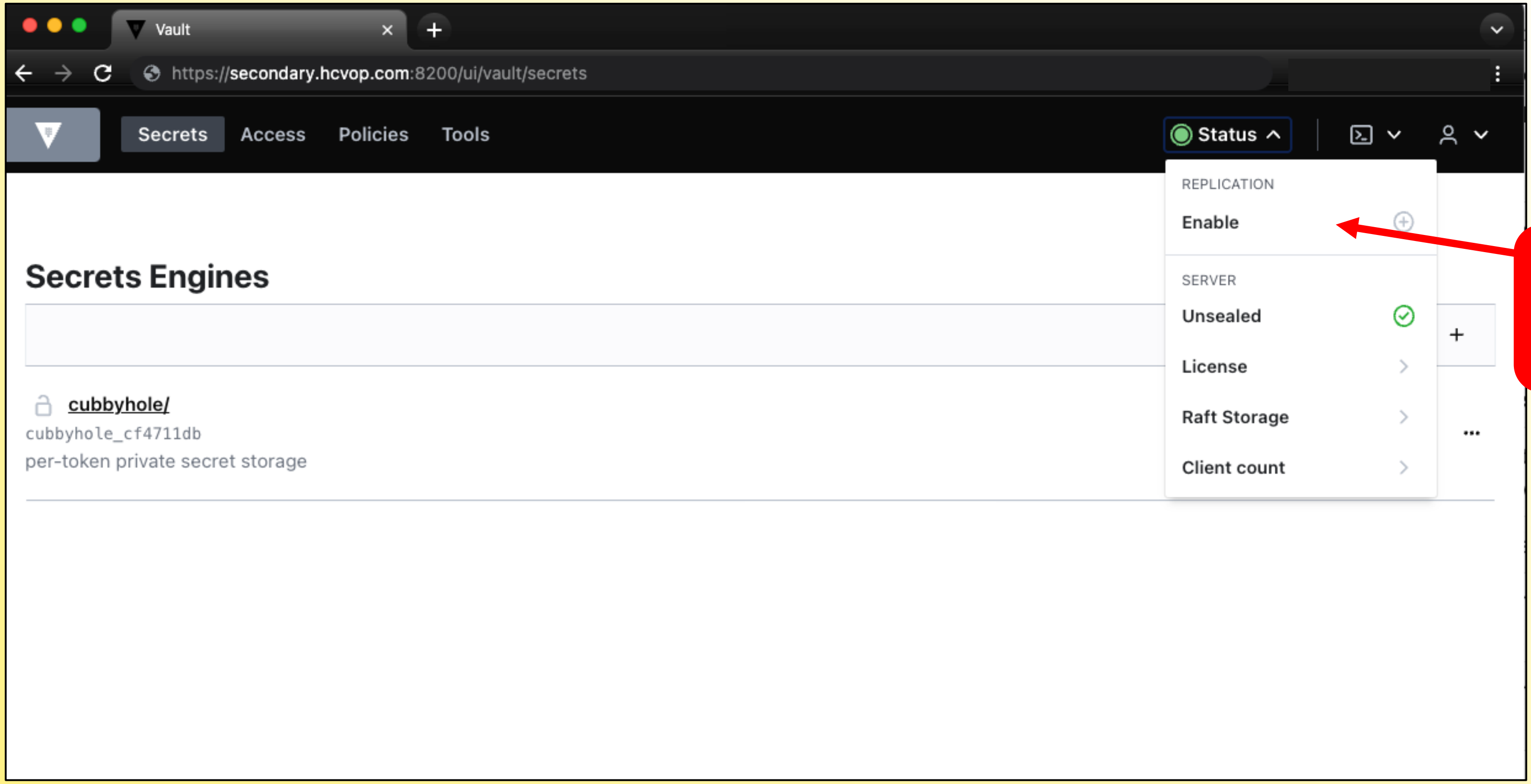
```
eyJhbGciOiJFUzUxMiIsInR5cCI6IkpXVCJ9.eyJhY2Nlc3NvciI6IiIsImFkZHIiOi
JodHRwOi8vYnRrLW1hY2Jvb2stcHJvOjgyMDAiLCJleHAiOjE2NTMzMzA2M
zgsImlhdCI6MTY1MzMyODgzOCwianRpIjoiaHZzLkRYSWV5ZGw3aU5TVW
RyMjlQUWZ1S0t6RyIsIm5iZiI6MTY1MzMyODgzMywidHlwZSI6IndyYXBwa
W5nIn0.AUNbcWkxLvVelQck82izvmFEccfatlENilA-Ilv7E06augO_9bRiAh9
```

TTL 1800s

Expires

Configure Replication using the UI

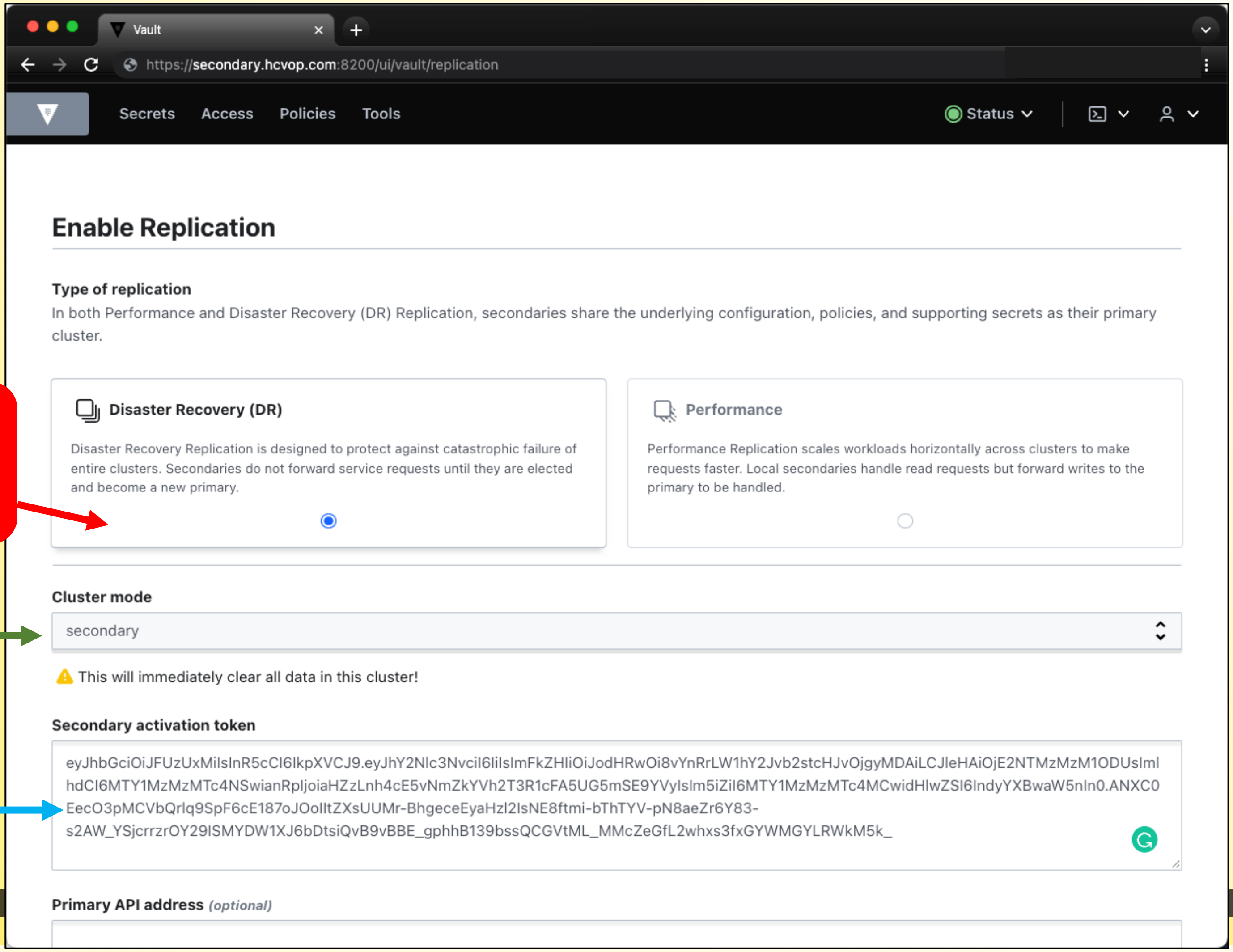
Enable Replication on Secondary Cluster



Enable Replication on Secondary

Configure Replication using the UI

Configure Secondary Cluster for Replication as a Secondary



Enable Replication

Type of replication

In both Performance and Disaster Recovery (DR) Replication, secondaries share the underlying configuration, policies, and supporting secrets as their primary cluster.

Disaster Recovery (DR) ●

Disaster Recovery Replication is designed to protect against catastrophic failure of entire clusters. Secondaries do not forward service requests until they are elected and become a new primary.

Performance ○

Performance Replication scales workloads horizontally across clusters to make requests faster. Local secondaries handle read requests but forward writes to the primary to be handled.

Cluster mode

secondary

⚠ This will immediately clear all data in this cluster!

Secondary activation token

```
eyJhbGciOiJFUzUxMiIsInR5cCI6IkpXVCJ9.eyJhY2Nlc3NvciI6IiIsImFkZHIiOiJodHRwOi8vYnRrLW1hY2Jvb2stcHJvOjgyMDAiLCJleHAiOjE2NTMzMzM1ODUsImlhdCI6MTY1MzMzMTc4NSwianRpIjoiaHZzLnh4cE5vNmZkYVh2T3R1cFA5UG5mSE9YVylsIm5iZiI6MTY1MzMzMTc4MCwidHlwZSI6IndyYXBwaW5nIn0.ANXC0EecO3pMCVbQrlq9SpF6cE187oJOolltZXsUUMr-BhgeceEyaHzl2IsNE8ftmi-bThTYV-pN8aeZr6Y83-s2AW_YSjcrrzrOY29ISMYDW1XJ6bDtsiQvB9vBBE_gphhB139bssQCGVtML_MMcZeGfL2whxs3fxGYWMGYLRWkM5k_
```

Primary API address *(optional)*

Select Disaster Recovery Replication

Select Secondary

Paste Secondary Token Here

Monitor Replication

Check Status of ALL Replication

```
$ vault read -format=json sys/replication/status
```

Check Status of Performance Replication

```
$ vault read -format=json sys/replication/performance/status
```

Performance Replication Only

Check Status of DR Replication

```
$ vault read -format=json sys/replication/dr/status
```

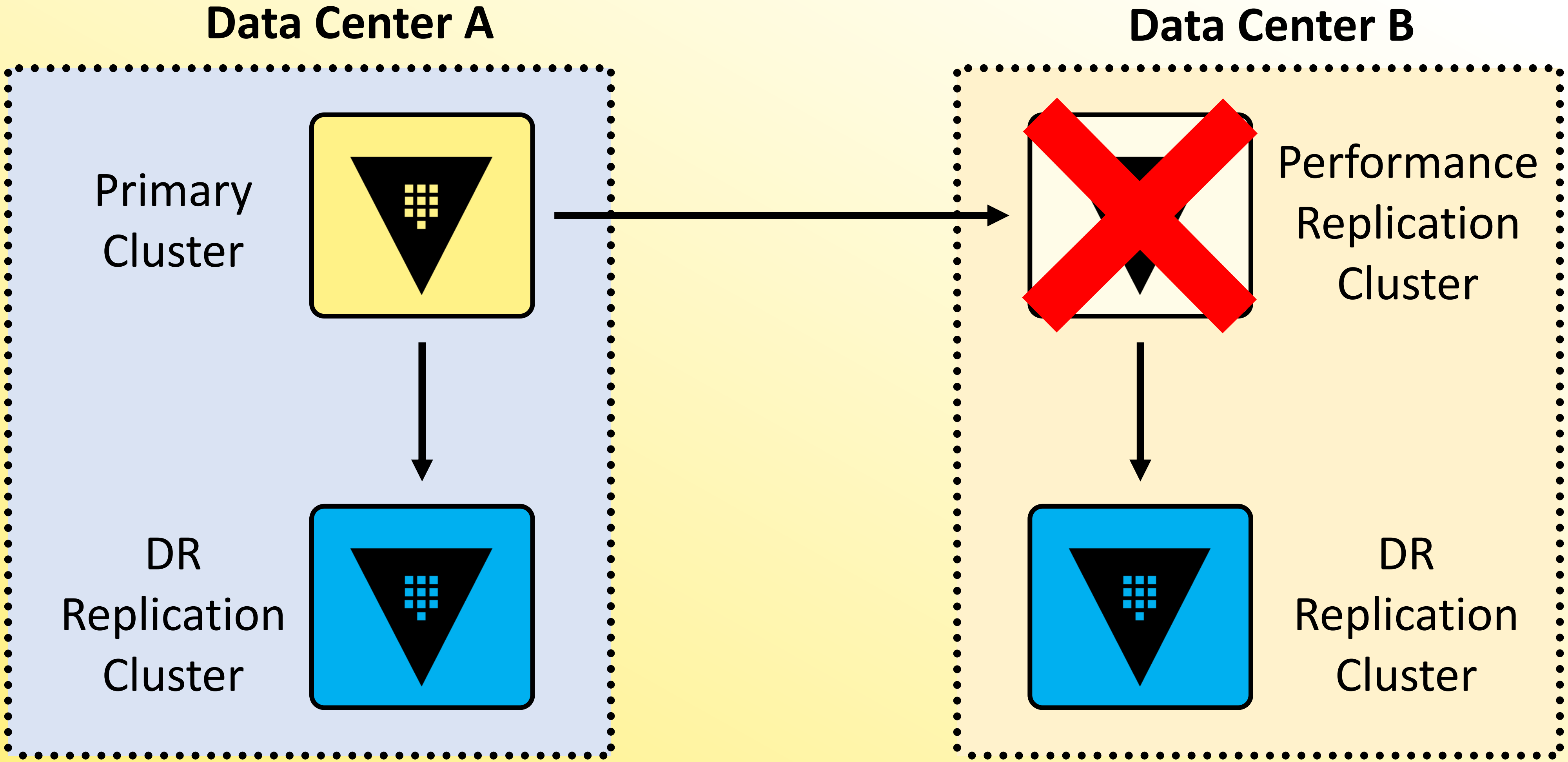
DR Replication Only

Vault

CERTIFIED
OPERATIONS
PROFESSIONAL

Promote a Secondary Cluster

Oh No…Our Cluster…It's Broken



Data Center A

Primary Cluster

DR Replication Cluster

Data Center B

Performance Replication Cluster

DR Replication Cluster

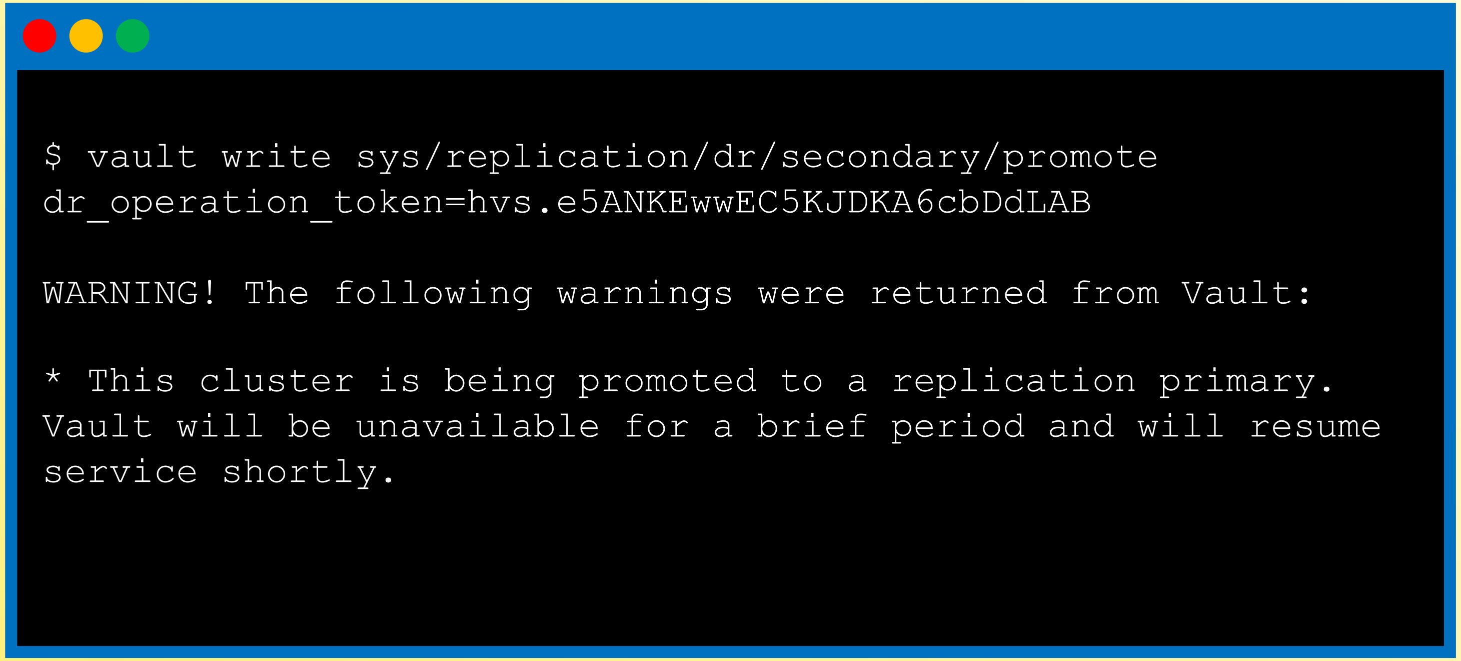
Promote a Secondary to a Primary

- Promotion of a DR cluster requires a DR Operation Token
  - This is generated directly on the DR cluster using the unseal/recovery keys
  - Process is similar to generating a root token (requires threshold of keys)

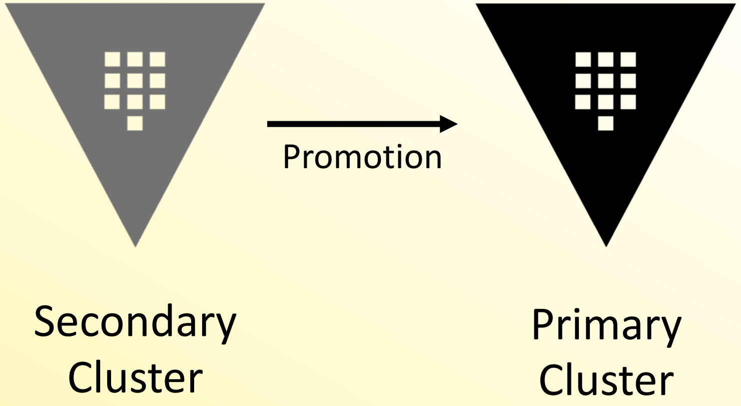
- Alternatively, you can create a DR Operation Batch Token on the primary BEFORE the failure
  - The idea is to have a valid token ready in the event of a failure
  - Reduces time to generate a DR Operation Token
  - BUT….you need to ensure the token TTL is valid

Promoting a Cluster

Once you have a token, you can use the token to promote the cluster

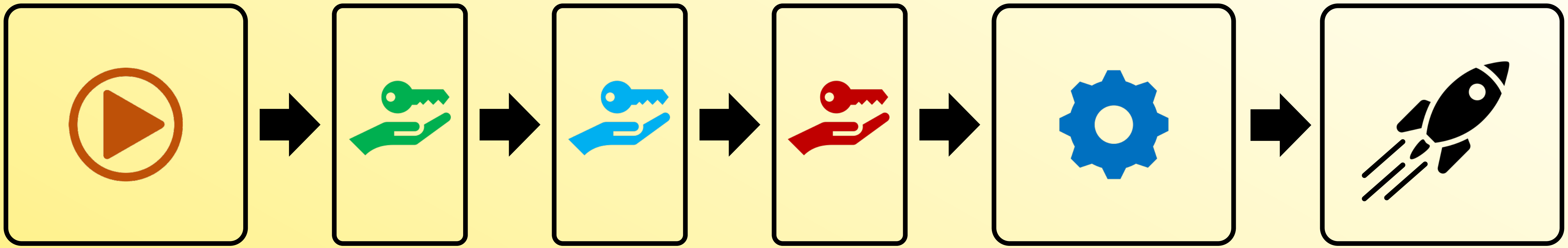


```
$ vault write sys/replication/dr/secondary/promote
dr_operation_token=hvs.e5ANKEwwEC5KJDKA6cbDdLAB

WARNING! The following warnings were returned from Vault:

* This cluster is being promoted to a replication primary.
Vault will be unavailable for a brief period and will resume
service shortly.
```



How Do I Get a DR Operation Token?

Similar to a `generate-root` process, the process must be initialized, and each key holder will need to provide their key to meet the configured threshold



Initialize the DR Token Generation

Key Holder Provides Key

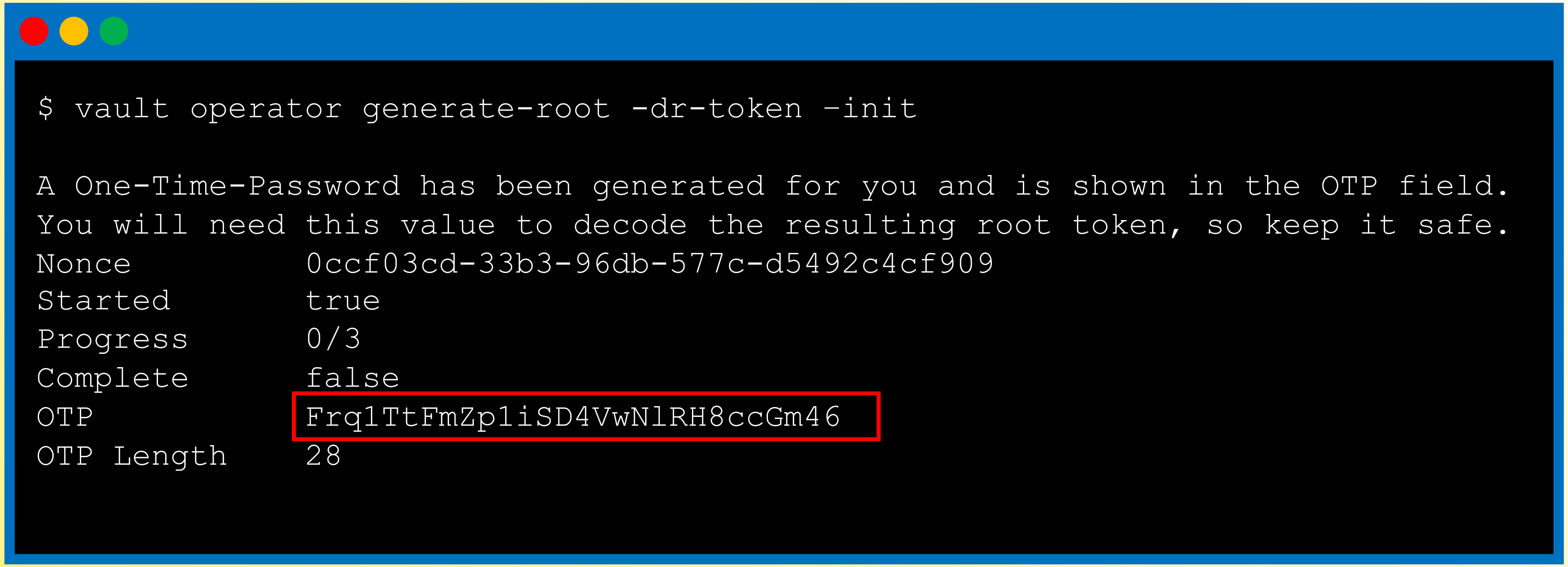
Key Holder Provides Key

Key Holder Provides Key

Decode using OTP to get DR Token

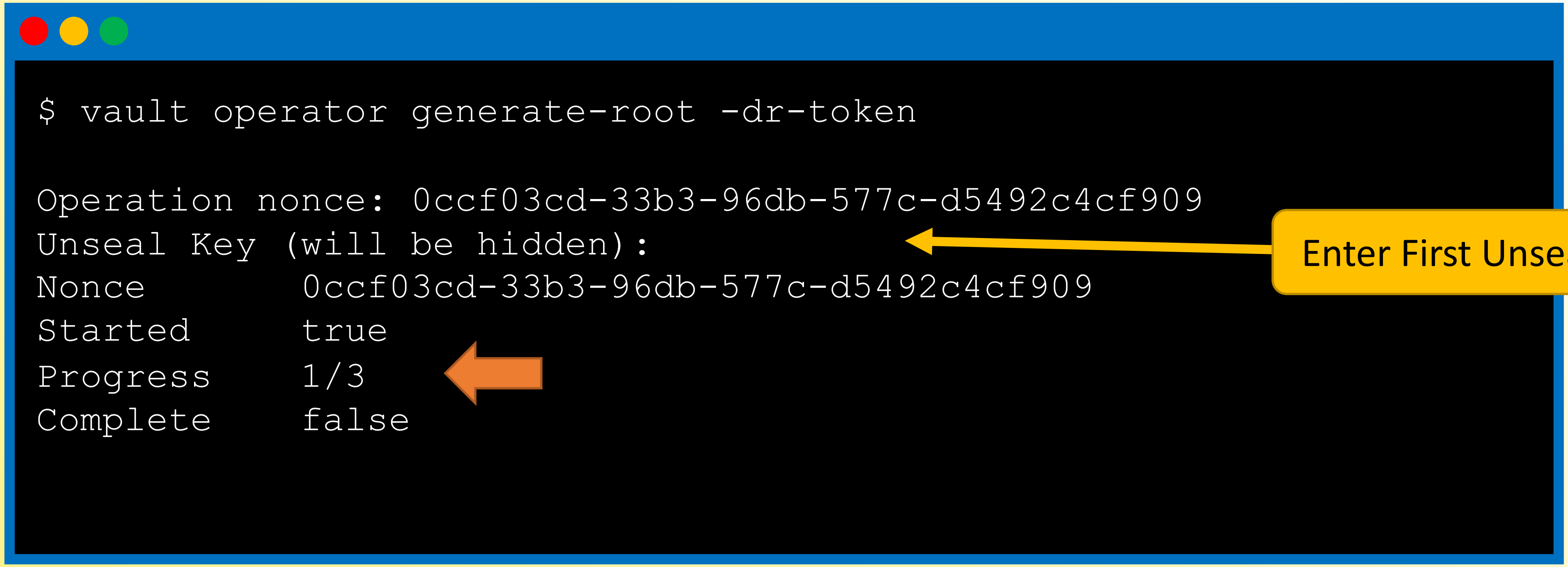
Promote Cluster

Initialize The Process



```
$ vault operator generate-root -dr-token -init

A One-Time-Password has been generated for you and is shown in the OTP field.
You will need this value to decode the resulting root token, so keep it safe.
Nonce          0ccf03cd-33b3-96db-577c-d5492c4cf909
Started        true
Progress       0/3
Complete       false
OTP            Frq1TtFmZp1iSD4VwNlRH8ccGm46
OTP Length     28
```

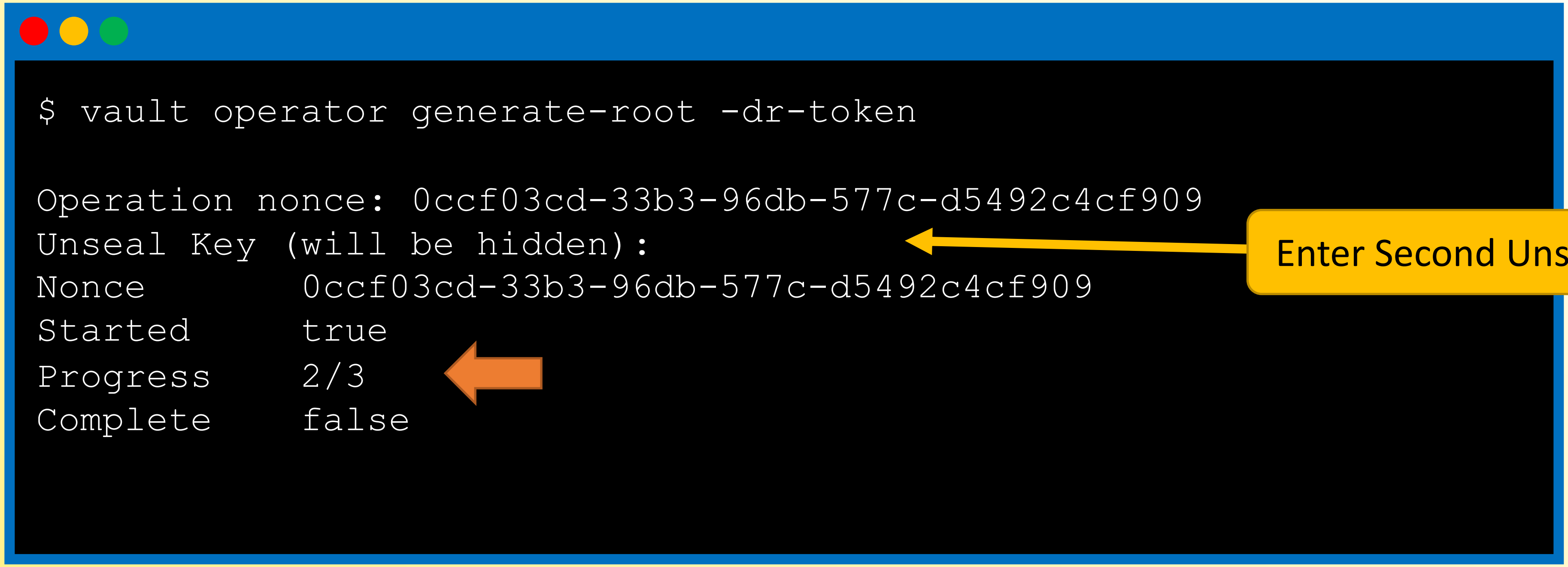
Provide the Keys



```
$ vault operator generate-root -dr-token

Operation nonce: 0ccf03cd-33b3-96db-577c-d5492c4cf909
Unseal Key (will be hidden):
Nonce          0ccf03cd-33b3-96db-577c-d5492c4cf909
Started        true
Progress       1/3
Complete       false
```

Enter First Unseal/Recovery Key Here

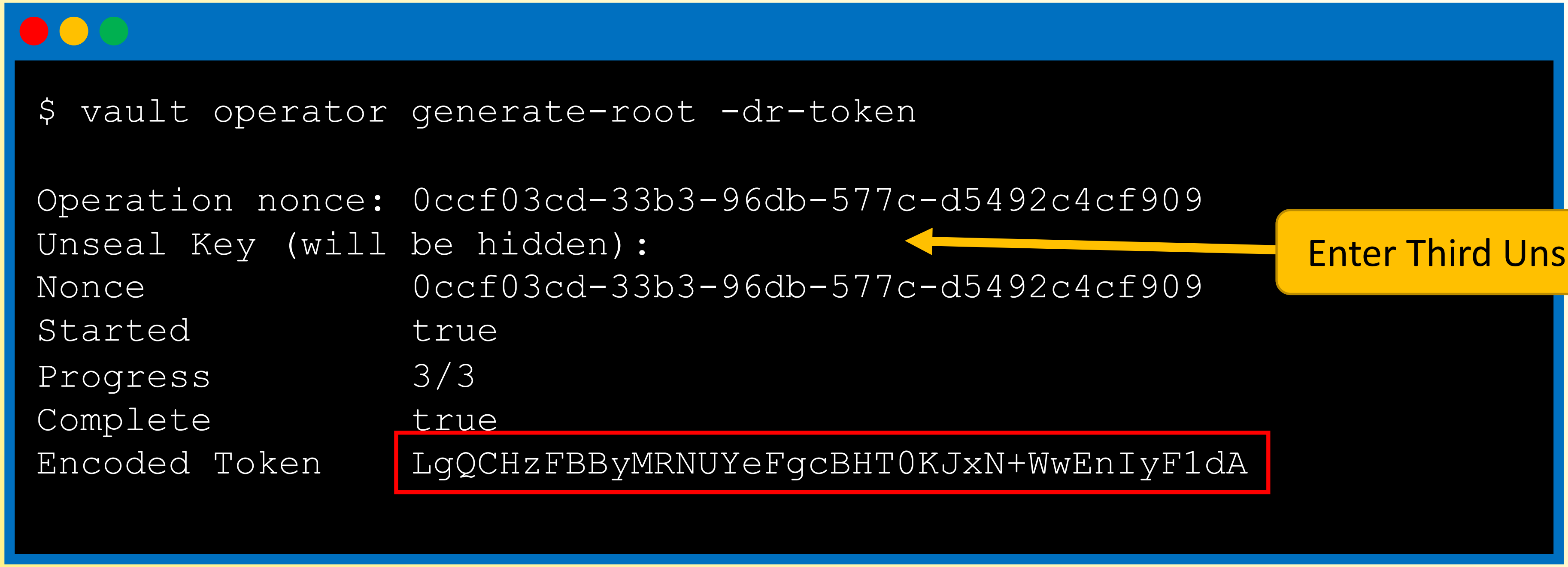
Provide the Keys



```
$ vault operator generate-root -dr-token

Operation nonce: 0ccf03cd-33b3-96db-577c-d5492c4cf909
Unseal Key (will be hidden):
Nonce           0ccf03cd-33b3-96db-577c-d5492c4cf909
Started         true
Progress        2/3
Complete        false
```

Enter Second Unseal/Recovery Key Here

Provide the Keys

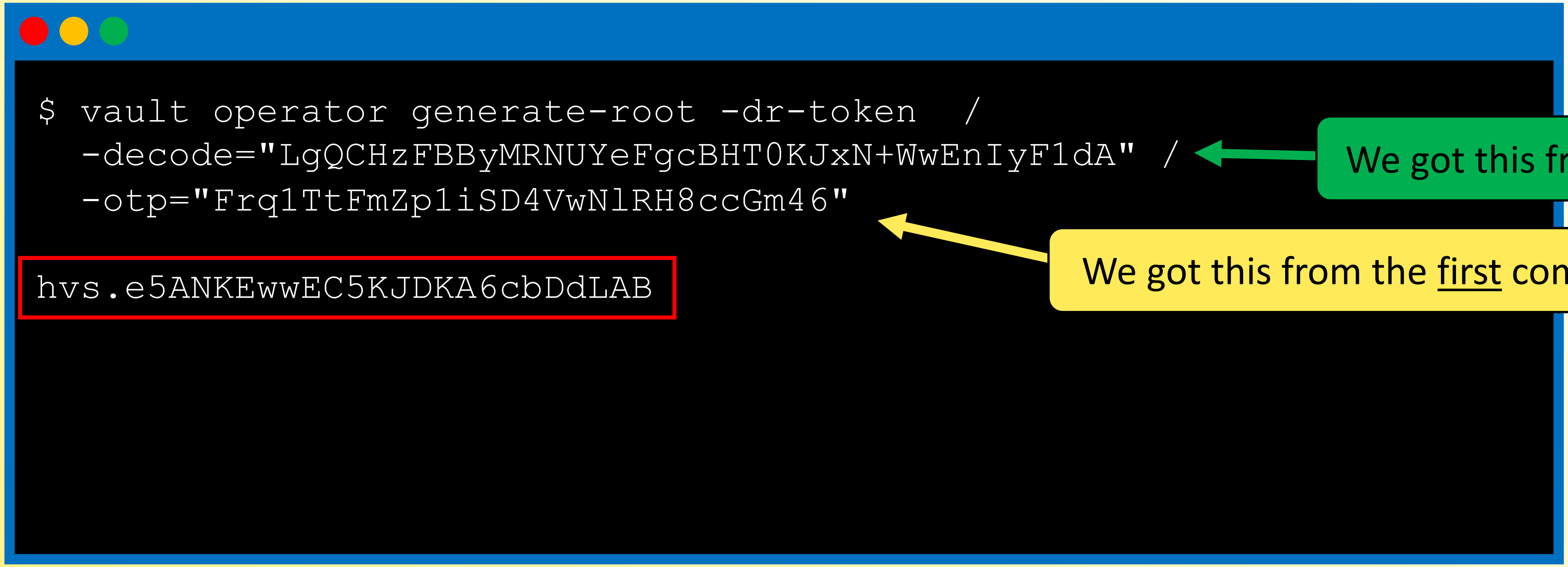


```
$ vault operator generate-root -dr-token

Operation nonce: 0ccf03cd-33b3-96db-577c-d5492c4cf909
Unseal Key (will be hidden):
Nonce            0ccf03cd-33b3-96db-577c-d5492c4cf909
Started          true
Progress         3/3
Complete         true
Encoded Token    LgQCHzFBByMRNUYeFgcBHT0KJxN+WwEnIyF1dA
```

Enter Third Unseal/Recovery Key Here

Decode the DR Operation Token

Vault
CERTIFIED
OPERATIONS
PROFESSIONAL



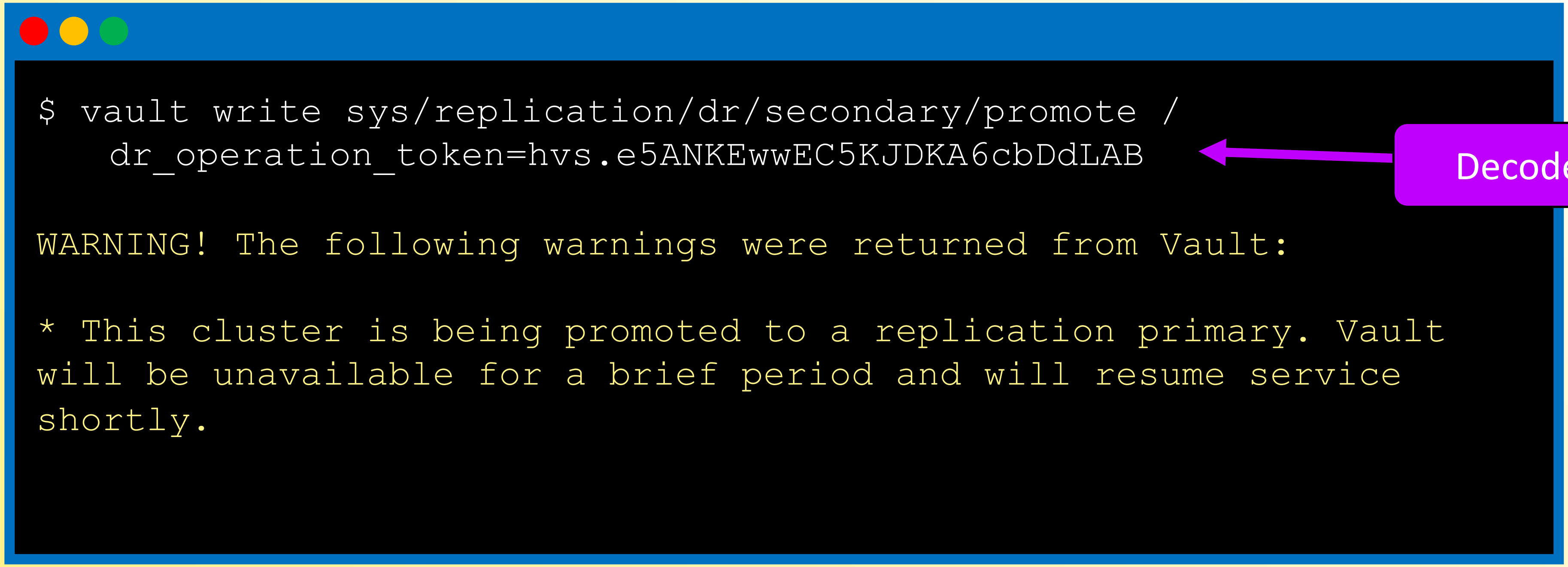
```
$ vault operator generate-root -dr-token  /
  -decode="LgQCHzFBByMRNUYeFgcBHT0KJxN+WwEnIyF1dA" /
  -otp="Frq1TtFmZp1iSD4VwNlRH8ccGm46"

hvs.e5ANKEwwEC5KJDKA6cbDdLAB
```

We got this from the final command

We got this from the first command

Promote the Cluster

Vault CERTIFIED OPERATIONS PROFESSIONAL



```
$ vault write sys/replication/dr/secondary/promote /
    dr_operation_token=hvs.e5ANKEwwEC5KJDKA6cbDdLAB

WARNING! The following warnings were returned from Vault:

* This cluster is being promoted to a replication primary. Vault
will be unavailable for a brief period and will resume service
shortly.
```

Decoded Token