HashiCorp Vault

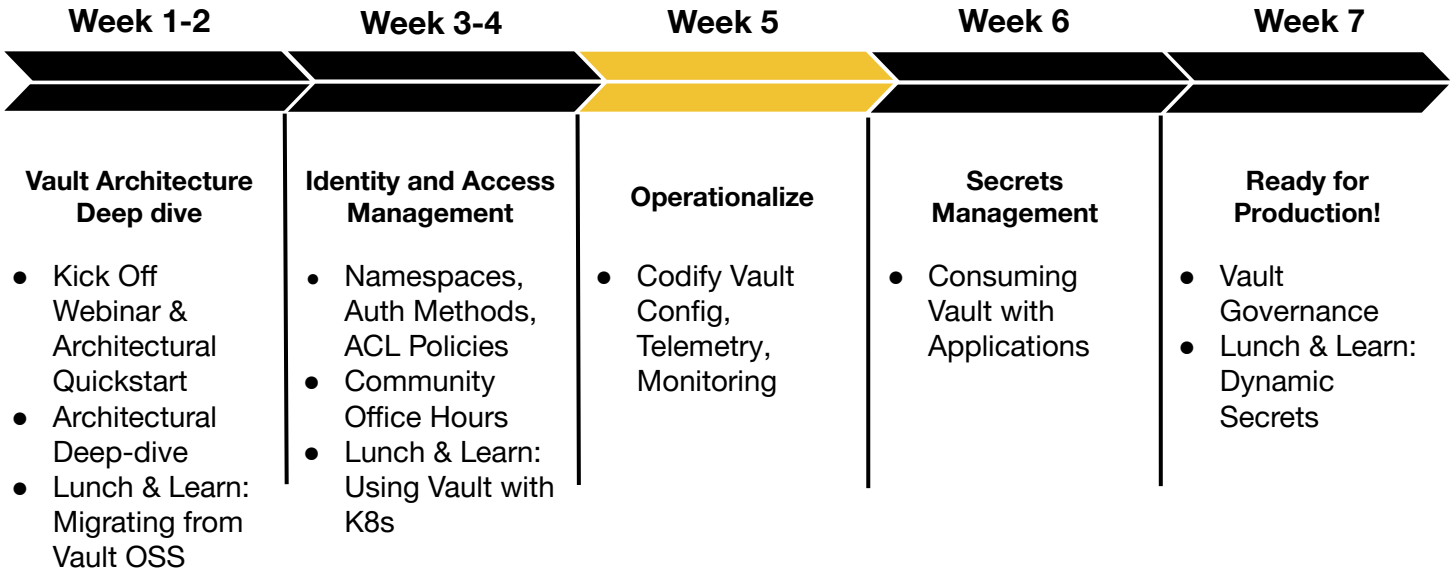
Operating Vault Enterprise

August 2022

Agenda

- Telemetry & Monitoring
- DR Operations
- Vault Runbooks
- Next Steps
- Q & A

Vault Enterprise Path to Production



01

Telemetry & Monitoring

Monitoring Patterns

- Vault is typically classified as a Tier 0 application based upon critical applications having it as a upstream dependency
- Monitoring the health of Vault clusters in Production Environments should include all three of the following patterns
    1. Time-series Telemetry Data
    2. Log Analytics
    3. Active Health Checks

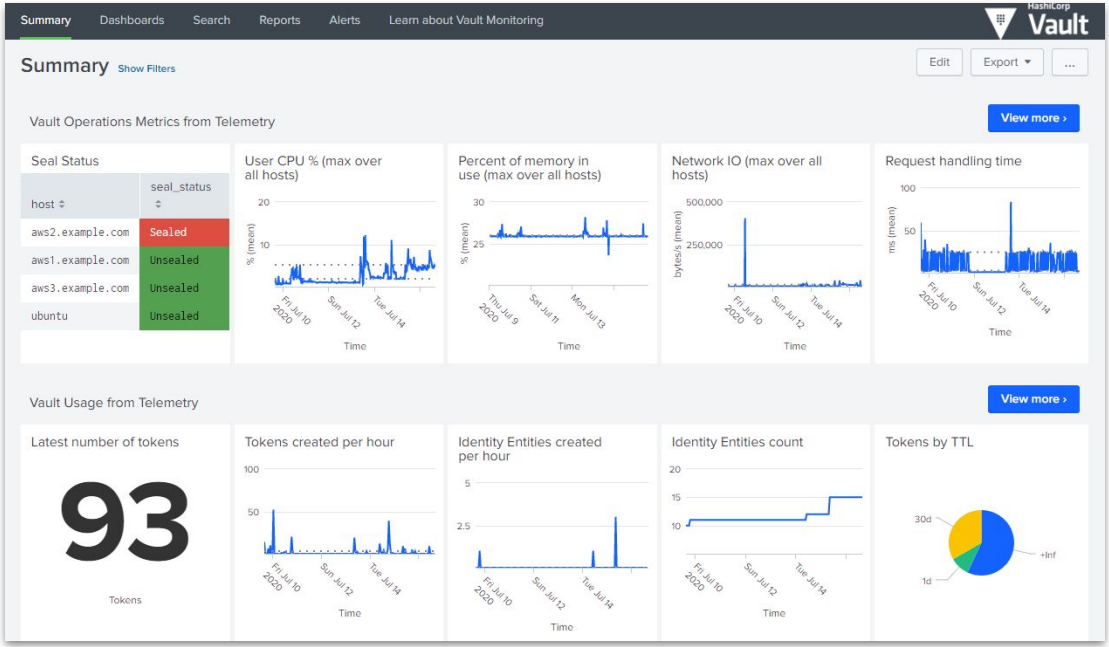
Vault Telemetry

- Vault uses the Golang go-metrics package to export Telemetry Metrics to an upstream service, specified in the telemetry stanza
- Supported sinks include:
  - Circonus
  - DogStatsd
  - Prometheus
  - Statsd
  - Statsite

Vault Metrics

- Run Time Metrics are aggregated across a 10 second interval and retained in memory for 1 minute

- Telemetry from Vault is best stored in a metrics aggregation platform to collect durable metrics and visualize trends



Metric Types

[C] Counter

Cumulative metrics that increment when an event occurs and are reset at the end of the reporting interval

[G] Gauge

Provides measurements of current values

[S] Summary

Provide sample observations of values. Commonly used to measure timing duration of discrete events in the reporting interval.

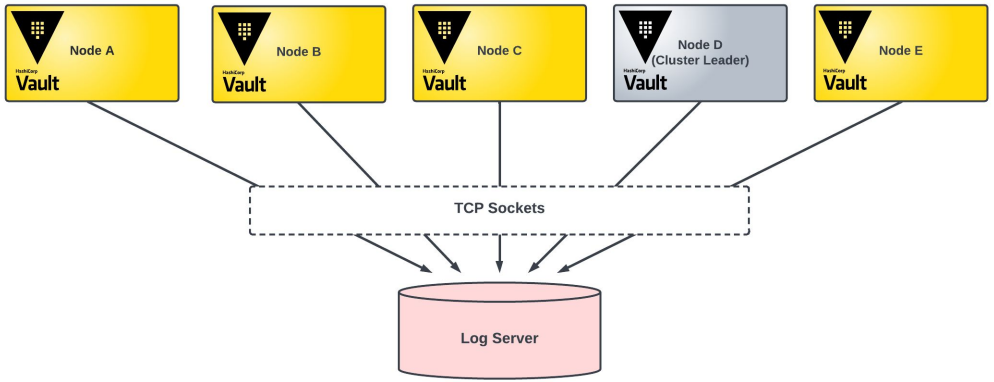
Vault Logging

- Vault generates two types of logs
  - Vault **server logs** contain operational data, plugin errors, debug dumps, and critical security events
  - JSON **audit logs** keep a detailed log of *every authenticated* interaction with Vault, including errors
- Multiple audits devices can (and should) be enabled and Vault will send audit logs to all of them
  - Redundant log copies can be used to check for data tampering in log files
  - Vault considers a request successful if it can log to at least one configured audit device
- If you have only one audit device enabled and it is blocked (network issue, full disk, etc) then Vault will seal itself and cease operations ***by design***

Vault Logging

Socket-to-remote Log Server

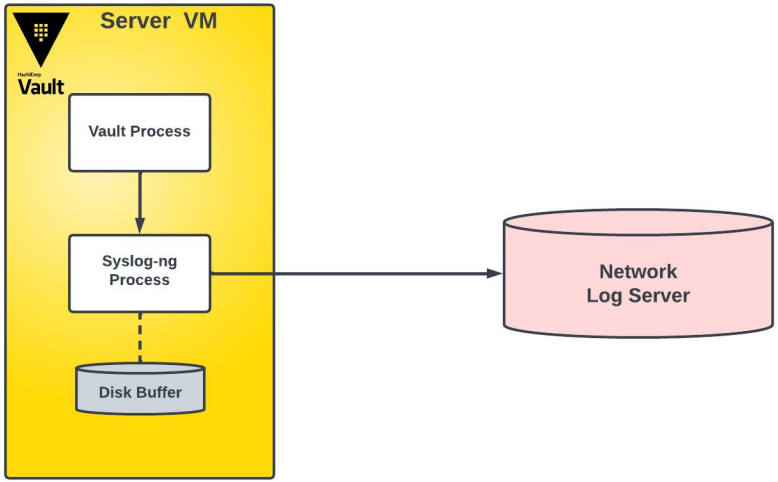
- Vault's socket audit device writes to a TCP, UDP, or UNIX socket
- Using a network log server retains logs in JSON format for easy parsing, and eliminates the potential to fill a local filesystem
- Use with a second audit method or with multiple instances to prevent downtime caused by log server failure



Vault Logging

Syslog-ng daemon

- Utilize syslog-ng daemon to forward audit logs directly to a centralized log server
- Any analysis or resource intensive tasks are performed on the remote log server while keeping Vault simple and continuing to service requests
- Very reliable solution because of its simplicity



Enable Telemetry

Specify upstream monitoring service in telemetry stanza in Vault Server configuration

[Telemetry - Configuration]

```
CODE EDITOR

telemetry {
  statsd_address = "statsd.company.local:8125"
}
```

Contributing Factors in Performance

- Know the expected workload
- Vault System Resources (CPU, MEM, Disk)
- Complexity of the Vault Policies
- Audit Logging
- Network for all the things

Contributing Factors in Performance

Vault CPU

Select the host VMs to handle the concurrent workload

Policies

Never run load test using root token

Network

Know all the systems that are involved and connectivity between them

Audit Device

Be sure that the write location won't become the bottleneck

Storage Backend

- Determine appropriate TTLs for tokens and leases
- Leverage batch tokens and Vault Agent Caching

Key System Metrics

Metric	Description	What to look for?
cpu.user_cpu	Percentage of CPU being used by user processes	Look for high Vault CPU consumption
cpu.iowait_cpu	Percentage of CPU time spent waiting for I/O tasks to complete	Look for `cpu.iowait_cpu` greater than 10%
net.bytes_recv	Bytes received on each network interface	Look for sudden large changes to the net metrics (greater than 50% deviation from baseline)
net.bytes_sent	Bytes transmitted on each network interface	
linux_sysctl_fs.file-nr	Number of file handles being used across all processes on the host	If the `file-nr` reaches 80% of the `file-max` then you should alert and investigate
linux_sysctl_fs.file-max	Total number of available file handles	

Key Integrated Storage Metrics

Metric	Description	What to look for?
vault.runtime.total_gc_pause_ns	Number of nanoseconds consumed by stop-the-world garbage collection (GC) pauses since Vault started	This would be considered a _warning_ if `total_gc_pause_ns` exceeds 2 seconds/minute and _critical_ if it exceeds 5 seconds/minute
vault.raft.leader.lastContact`	Time to retrieve a value for a path	Look for candidate > 0, or leader > 0, or `lastContact` greater than 200ms which indicates that consensus is unhealthy
vault.raft.state.candidate	Time to insert a log entry to the persist path	
vault.raft.state.leader	This increments whenever a raft server becomes a leader	
diskio.read_bytes	Bytes read from each block device	You will want to monitor for large changes to the diskio metrics for greater than 50% deviation from baseline, or more than 3 deviations from your standard baseline. Then you will want to monitor for over 80% utilization on block device mount points on which Vault data are persisted.
diskio.write_bytes	Bytes written to each block device	
disk.used_percent	Per-mount-point block device utilization	

Key Usage Metrics

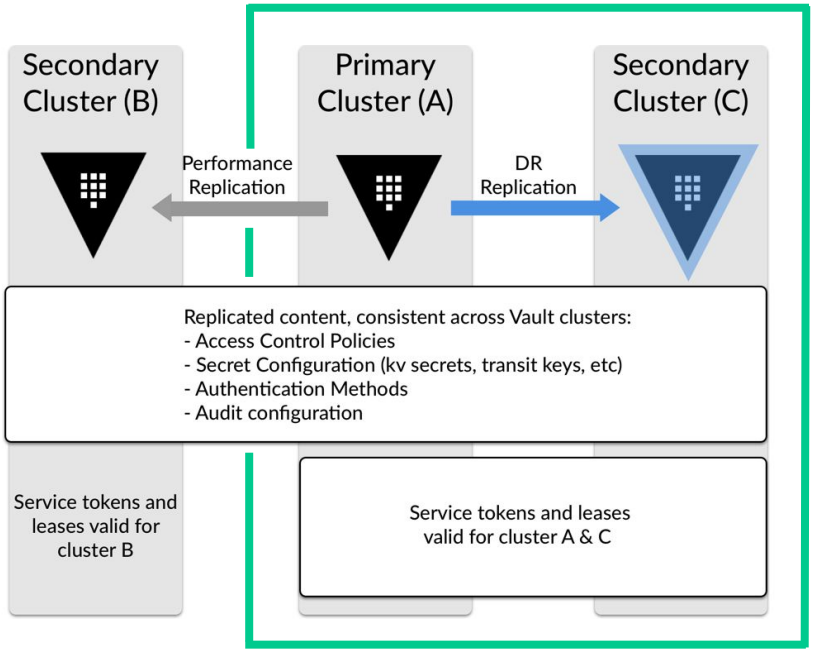
Metric	Description
vault.token.creation	A new service or batch token was created
vault.token.count	Number of service tokens available for use.
vault.token.count.by_auth	Number of existing tokens broken down by the auth method used to create them.
vault.token.count.by_policy	Number of existing tokens, counted in each policy assigned.
vault.token.count.by_ttl	Number of existing tokens, aggregated by their TTL at creation.
vault.secret.kv.count	Count of secrets in key-value stores.
vault.secret.lease.creation	Count of leases created by a secret engine (excluding leases created internally for token expiration.)

02

DR Operations

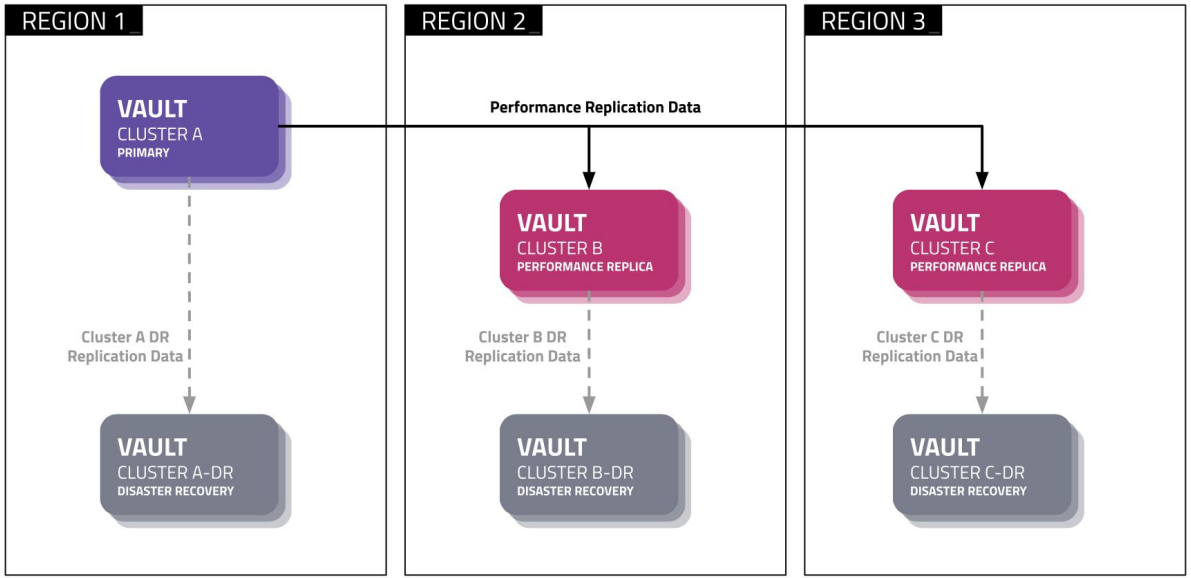
Vault Disaster Recovery Replication

- Vault Enterprise offers two modes of replication for high availability
- Many organizations leverage combinations of both DR & Performance replication to meet resilience objectives
- DR clusters do not forward read or write requests until promoted to primary, they are warm standbys



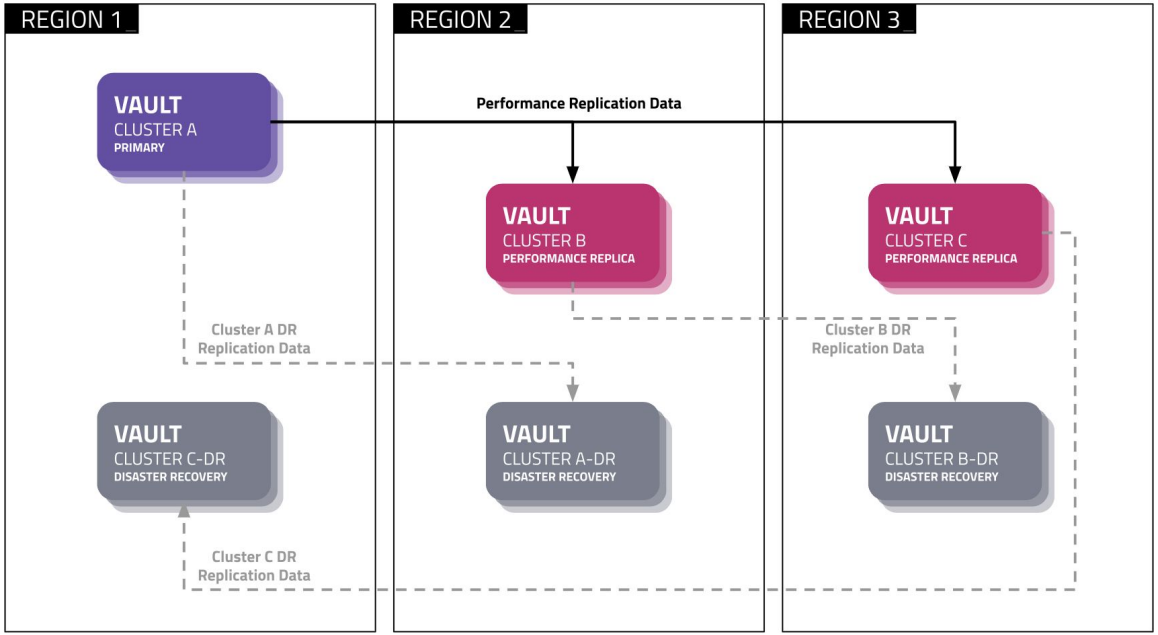
Resilience Against Cluster Failure

Leveraging Disaster Recovery and Performance Replication

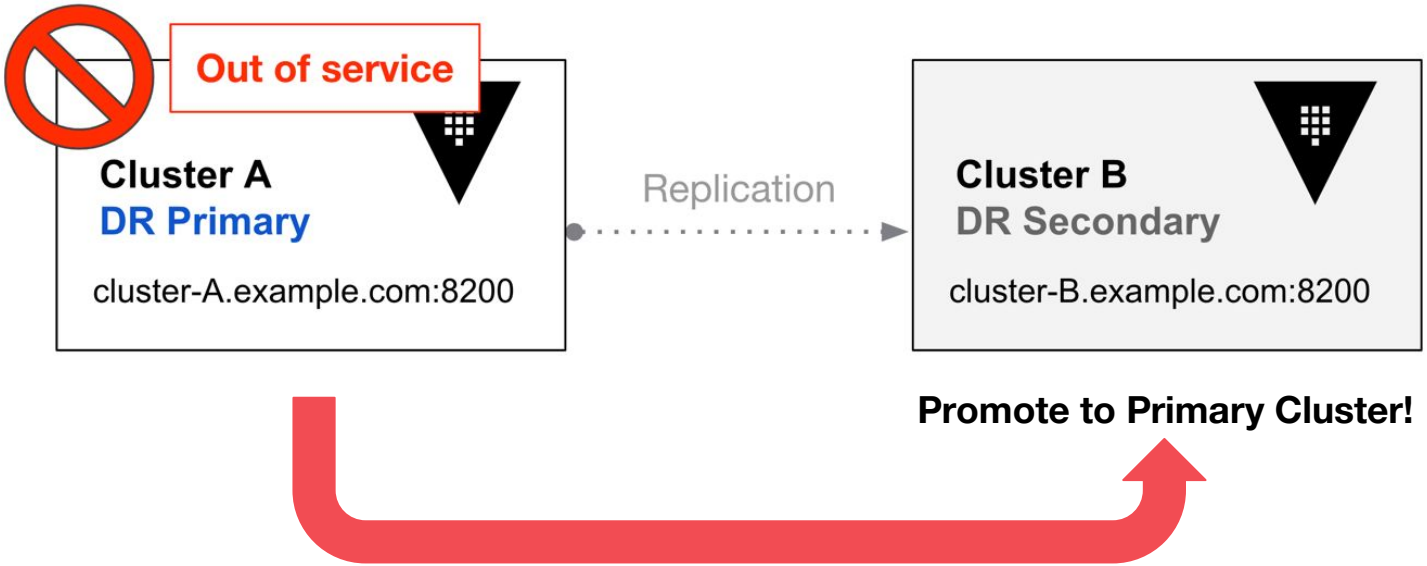


Resilience Against Region Failure

Leveraging Disaster Recovery and Performance Replication



DR Operation



DR Operation Token Strategy

DR Operation Token

- Requires a quorum of unseal/recovery keys to generate
- Can introduce delays in an emergency while gathering a quorum of key holders

Batch DR Operations Token

- Available in Vault 1.4 and higher
- Allows Vault Operator to generating a batch DR operation token ahead of time to prepare for a DR situation
- Have a fixed TTL and require management of the token lifecycle
- If expired will require the generation of a standard DR Operations Token

Batch DR Operations Token Policy

CODE EDITOR

```
path "sys/replication/dr/secondary/promote" {
  capabilities = [ "update" ]
}

# To update the primary to connect
path "sys/replication/dr/secondary/update-primary" {
    capabilities = [ "update" ]
}

# Only if using integrated storage (raft) as the storage
backend
# To read the current autopilot status
path "sys/storage/raft/autopilot/state" {
    capabilities = [ "update" , "read" ]
}
```

Generating Batch DR Operations Token

TERMINAL

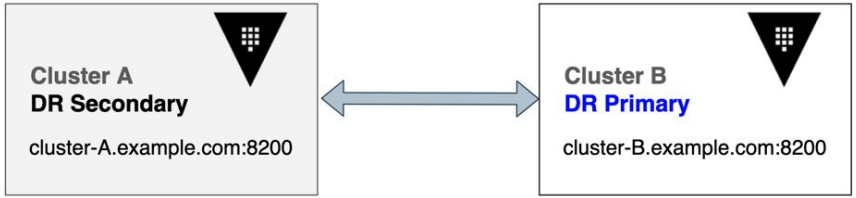
```
> vault write auth/token/roles/failover-handler \
    allowed_policies=dr-secondary-promotion \
    orphan=true \
    renewable=false \
    token_type=batch

> vault token create -role=failover-handler -ttl=72h
```

DR Failback

After a DR event, failing back to the original primary Vault Cluster is done by reversing the Promotion procedure:

1. The original primary (Cluster A) needs to be set as a DR Secondary
2. Cluster A syncs against Cluster B (current Primary)
3. Once synced, promote Cluster A to new Primary
4. Demote Cluster B back to DR Secondary



Automated DR Failover

Vault does not support an automatic failover/promotion of a DR secondary cluster, and this is a deliberate choice due to the difficulty in accurately evaluating why a failover should or shouldn't happen. For example, imagine a DR secondary loses its connection to the primary. Is it because the primary is down, or is it because networking between the two has failed?

If the DR secondary promotes itself and clients start connecting to it, you now have two active clusters whose data sets will immediately start diverging. There's no way to understand simply from one perspective or the other which one of them is right.

Are the DR Clusters in Sync

When the clusters are fully in sync, you can expect to see:

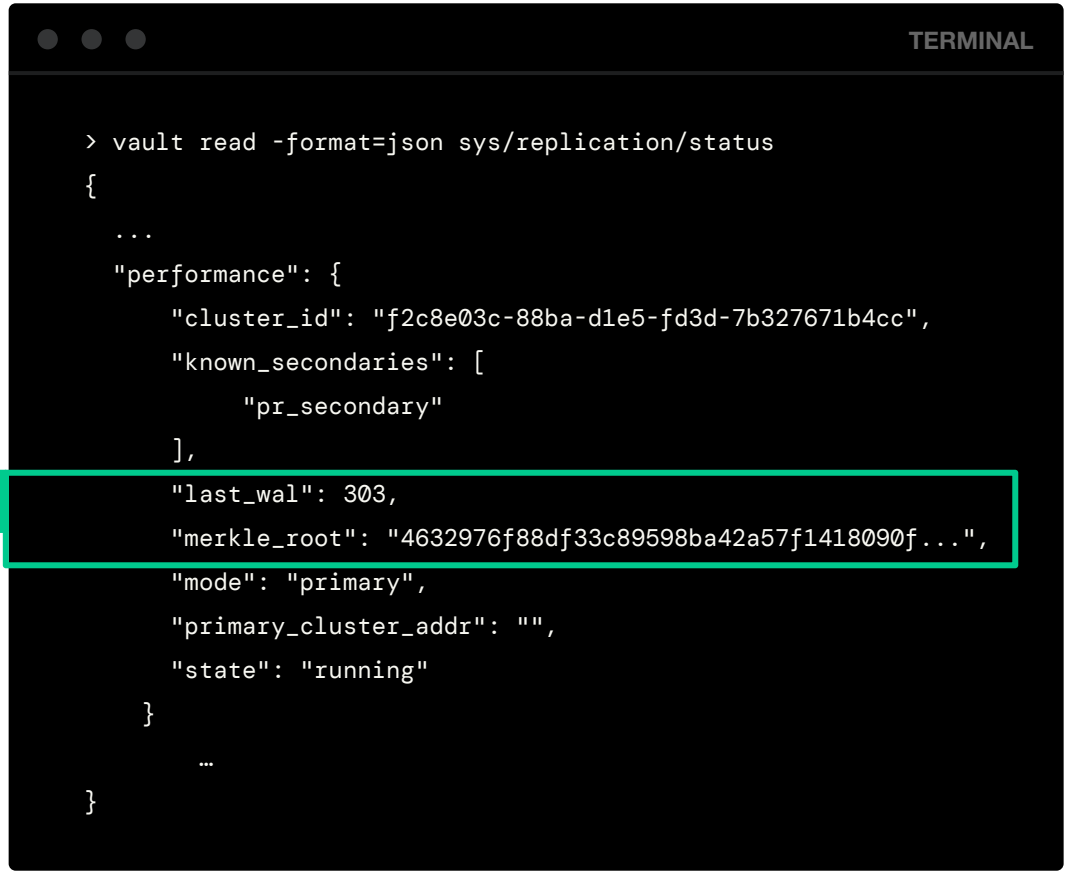
- **state** of secondary will be **stream-wals**
- **last_remote_wal** on the secondary should be very close to the **last_wal** on the primary
- **merkle_root** on the primary and secondary will match

Status Parameters on Secondaries

State	Description
stream-wals	Normal streaming (this is the value you want to see)
merkle-diff	The cluster is **determining** the sync status to see if a merkle sync is required
merkle-sync	The cluster is **syncing** - the secondary is too far behind the primary to use WALs to catch up (blocking operation)
idle	Indicates an **issue**. Needs investigation!

Verify Replication Status

Performance **Primary**



Verify Replication Status

Performance **Secondary**

TERMINAL

```
> vault read -format=json sys/replication/status
{
  ...
  "performance": {
      "cluster_id": "f2c8e03c-88ba-d1e5-fd3d-7b327671b4cc",
      "known_primary_cluster_addrs": [
          "https://perf-primary.example.com:8201"
      ],
      "last_remote_wal": 303,
      "merkle_root": "4632976f88df33c89598ba42a57f1418090f...",
      "mode": "secondary",
      "primary_cluster_addr":
"https://perf-primary.example.com:8201",
      "secondary_id": "pr_secondary",
      "state": "stream-wals"
    }
      …
}
```

WAL Metrics

Metric	Description
vault.wal_persistwals	Time taken to persist a WAL to storage
vault.wal_flushready	Time taken to flush a ready WAL to storage
wal.gc.total	Total number of WAL on disk
wal.gc.deleted	Number of WAL deleted during each garbage collection run

Replication Metrics

Metric	Description
logshipper.streamWALs.missing_guard	Number of missing guards: the Merkle tree index used to begin streaming WAL entries is not found/missing
logshipper.streamWALs.guard_found	Number of found guards
replication.fetchRemoteKeys	Time taken (in milliseconds) to perform a Merkle tree based delta generation between the clusters
replication.merkleDiff	Time taken (in milliseconds) to perform a Merkle tree based delta generation between the clusters
replication.merkleSync	Time taken (in milliseconds) to perform a Merkle tree based synchronization using the last delta generated between the clusters

Replication State Management

WAL Replays

WALs are replayed at server startup as well as during a reindex. At startup, the WAL replay blocks the incoming requests (no reads or writes). If replication is in a bad state or data has been removed from the storage backend without Vault's knowledge, reindex the Merkle tree via sys/replication/reindex endpoint

Merkle Tree

Vault uses a Merkle Tree to replicate data consistent across the Primary and Secondary Clusters.

- Use sys/replication endpoint to restore the replication state
- If replication is in an adverse state:

  ```
  vault write -f sys/replication/recover
  ```
- Manually reindex the local data storage:

  ```
  vault write -f sys/replication/reindex
  ```

03

Runbooks

Runbooks

As you proceed towards production, runbooks should be created for the operations involved in managing the lifecycle of your Vault cluster

Common runbooks include:

1. Backup/Restore (from snapshots)
2. DR Operations & Testing
3. Upgrade Procedures

Developing Runbook

1. Identify scenarios
2. Identify RACI
3. Document tasks
4. Test in non-production
5. Implement Runbook in production
6. Lifecycle management

Upgrades

Vault 1.11.X added automated upgrades to Autopilot

- Autopilot will upgrade a Vault cluster automatically whenever newer servers join the cluster
- When enabled Autopilot will **promote the newer versioned nodes to voters** and **demote the older versioned nodes to non-voters**
- When leadership transition is complete, the older version nodes are removed
- Vault Autopilot configuration

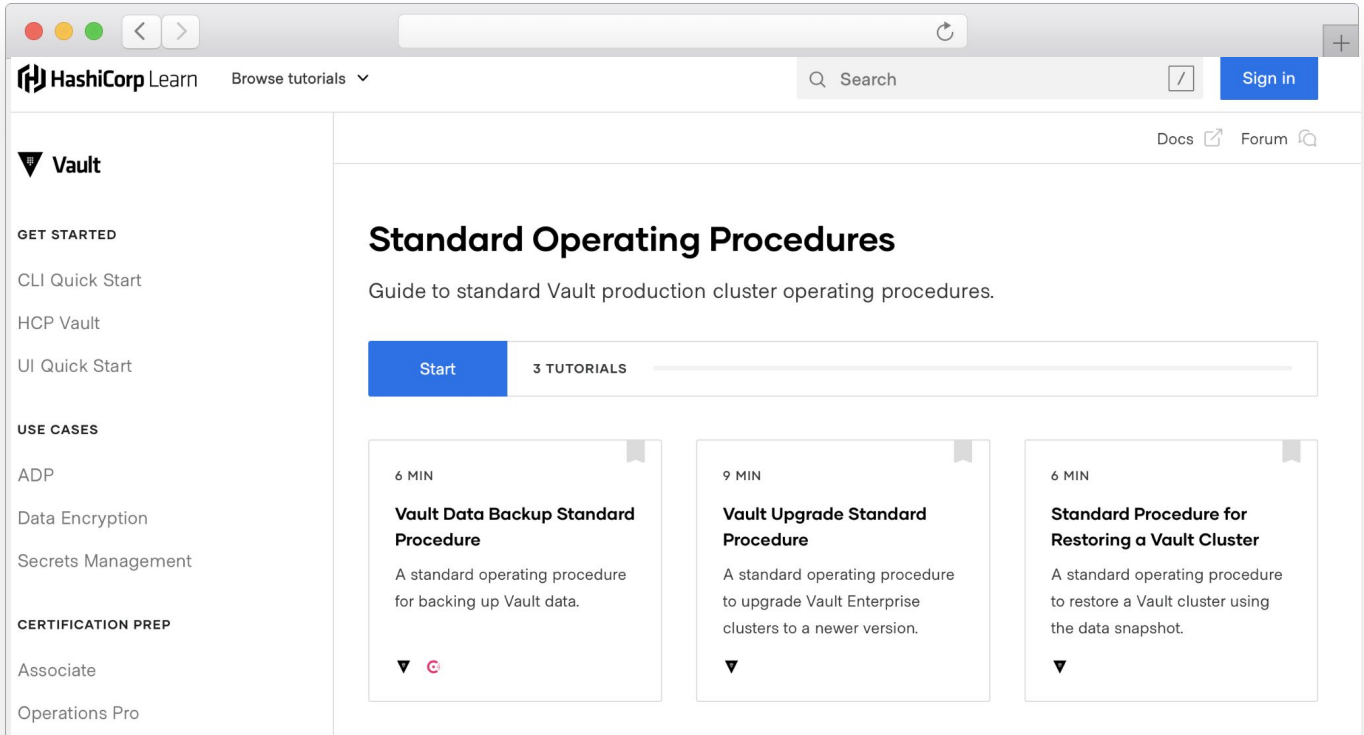
Upgrades

If not using Autopilot for automatic upgrades, or upgrading Vault versions prior to 1.11.0

- Take a snapshot of the Vault cluster
- Perform a “rolling upgrade” of the cluster
  - Stop Vault on a follower node
  - Replace the binary with the newer version
  - Restart Vault on the updated node
  - Verify logs for errors
  - Repeat on all following nodes in cluster
  - Stop Vault on the leader (active) node which will force a leadership change
  - Replace the Vault binary and restart the process

Vault SOP

https://learn.hashicorp.com/collections/vault/standard-procedures



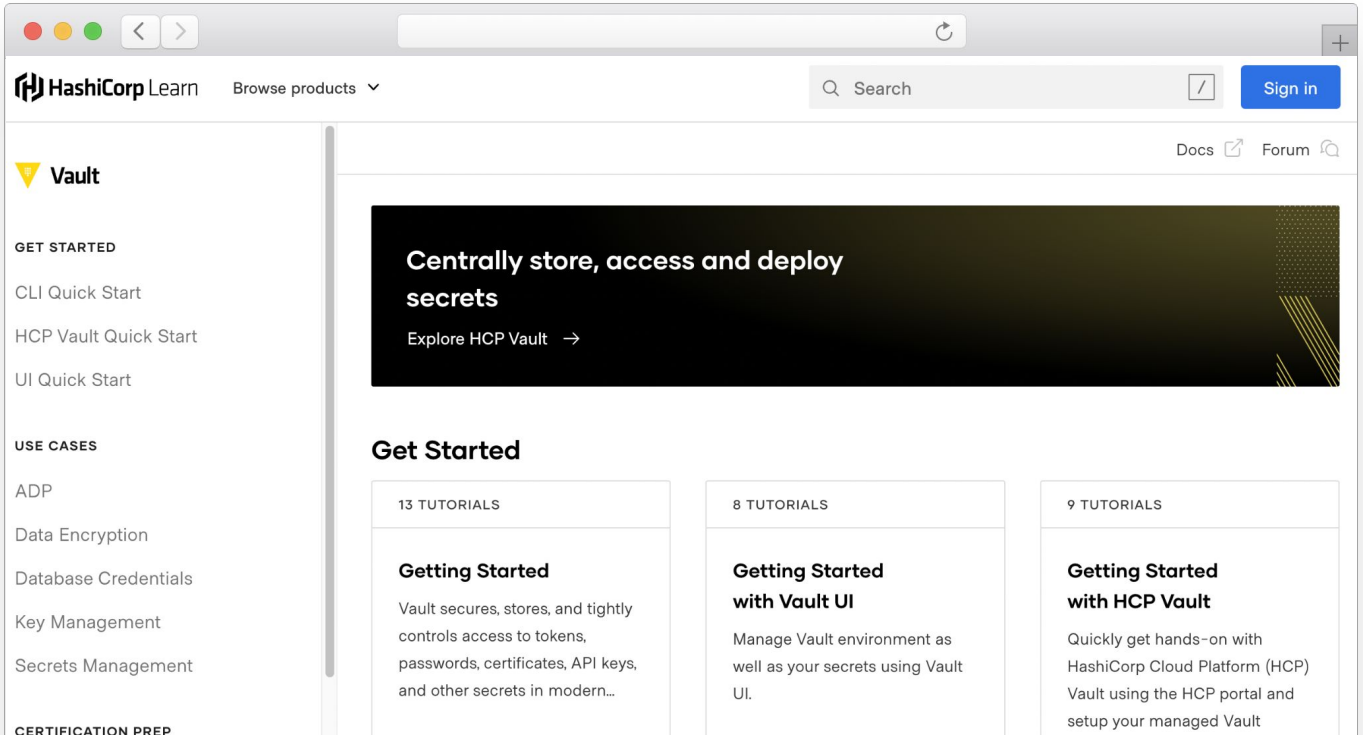
04

Next Steps

Learn

https://learn.hashicorp.com/vault

Step-by-step guides to accelerate deployment of Vault



Resources

- Vault Alerting & Logging on Day 1 (Blog)
- Monitor Telemetry & Audit Logs with Splunk
- Monitor Telemetry with Prometheus & Grafana
- Vault Telemetry Reference
- Disaster Recovery Replication Setup
- Monitoring Vault Replication
- How to Perform a Vault DR Drill
- Emergency Break-Glass Features

Need Additional Help?

Customer Success

Contact our Customer Success Management team with any questions. We will help coordinate the right resources for you to get your questions answered.

customer.success@hashicorp.com

Technical Support

Something not working quite right? Engage with HashiCorp Technical Support by opening a ticket for your issue at support.hashicorp.com.

Discuss

Engage with the HashiCorp Cloud community including HashiCorp Architects and Engineers

discuss.hashicorp.com

COBRA  Vault Onboarding Journey

Up Next…

- Webinar: Consuming Vault with your Applications
- Lunch & Learn: Vault Dynamic Secrets

05

Q & A

Thank You

customer.success@hashicorp.com
www.hashicorp.com