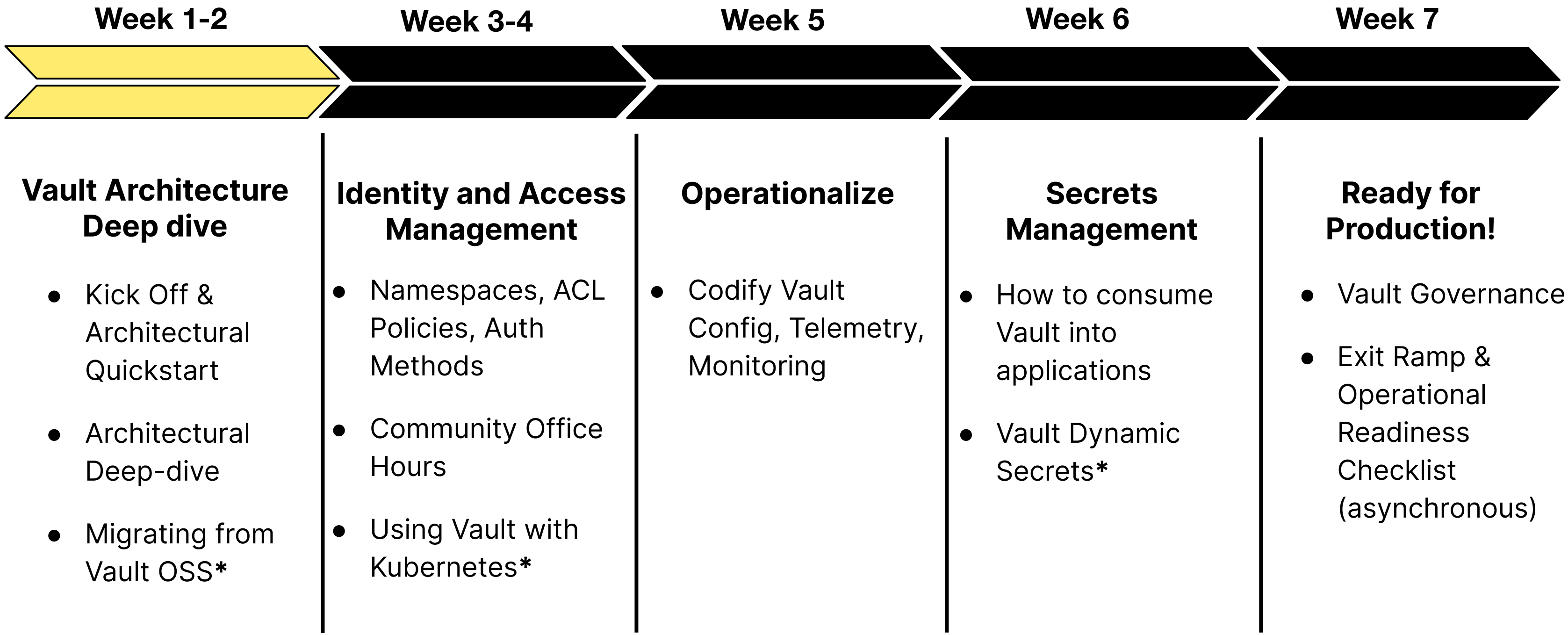
# Vault Enterprise Technical Overview & Architectural Deep-Dive

# Agenda

| | |
|---|---|
| Overview | 01 |
| Architecture | 02 |
| Deployment Patterns | 03 |
| Operations | 04 |

# Vault Onboarding Program

A 7 week guided community environment
Assisting customers with onboarding and adoption



**Week 1-2**

**Vault Architecture Deep dive**

- Kick Off & Architectural Quickstart
- Architectural Deep-dive
- Migrating from Vault OSS*

**Week 3-4**

**Identity and Access Management**

- Namespaces, ACL Policies, Auth Methods
- Community Office Hours
- Using Vault with Kubernetes*

**Week 5**

**Operationalize**

- Codify Vault Config, Telemetry, Monitoring

**Week 6**

**Secrets Management**

- How to consume Vault into applications
- Vault Dynamic Secrets*

**Week 7**

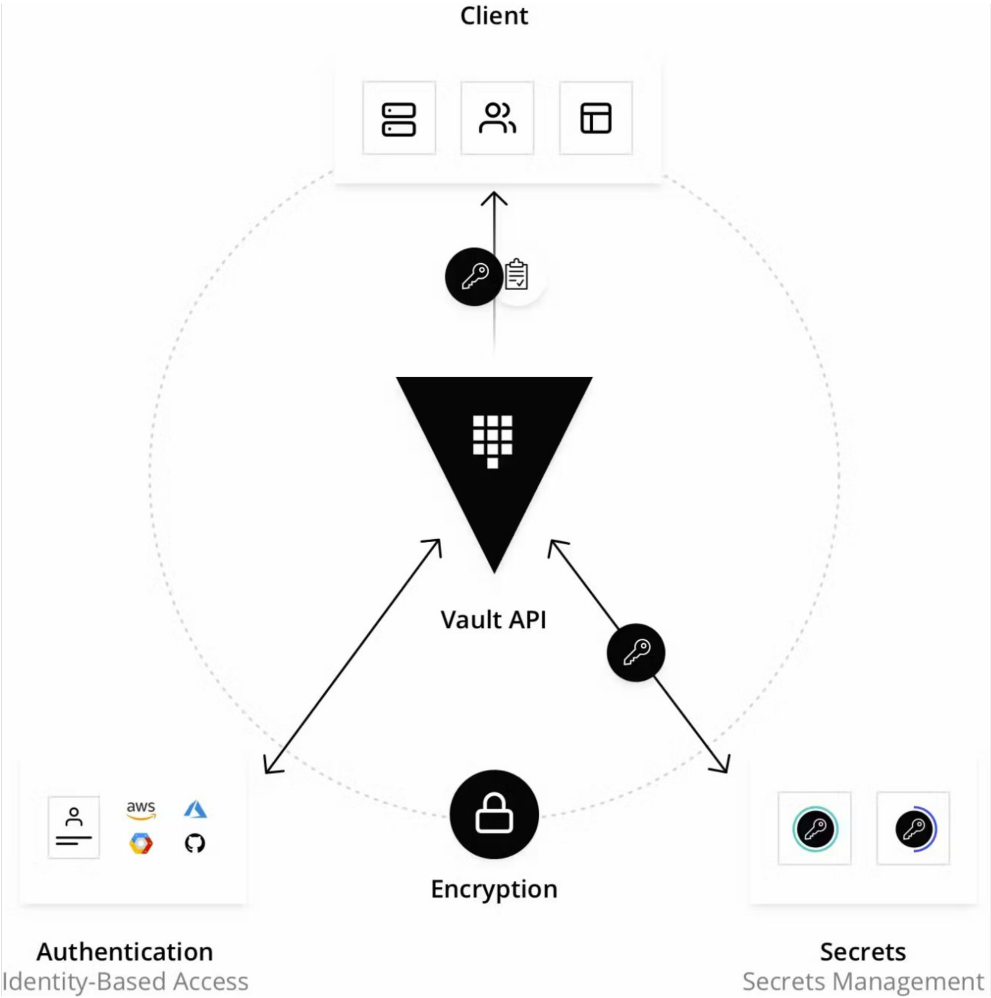
**Ready for Production!**

- Vault Governance
- Exit Ramp & Operational Readiness Checklist (asynchronous)

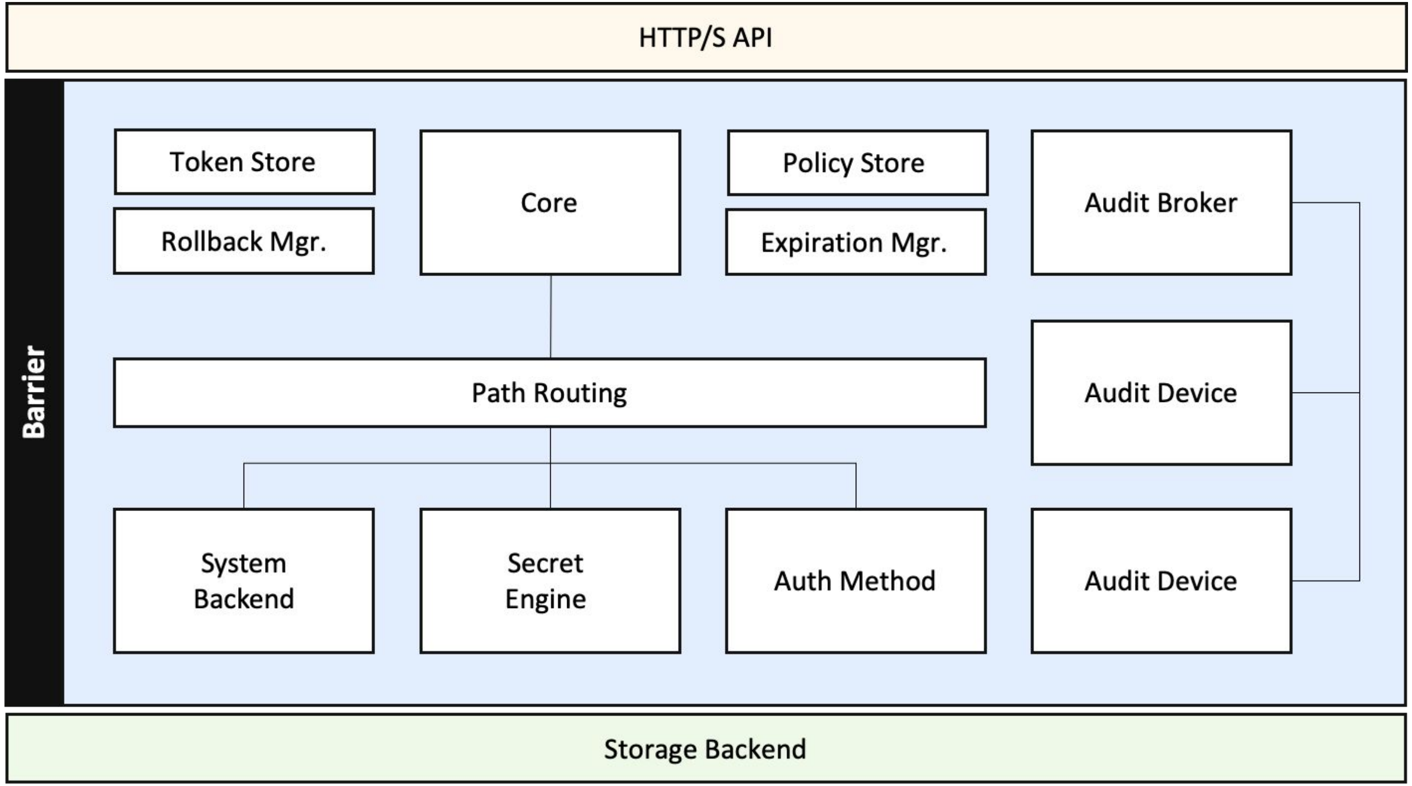
01

# Overview

# Overview


Client
Vault API
Encryption
Authentication
Identity-Based Access
Secrets
Secrets Management
aws

# Architecture & Cryptographic Barrier



# Vault Security Model

- It’s all about access to the Encryption Key
- Configuring “**`cap_ipc_lock=+ep`**”, **`LimitNOFILE`**, and **`LimitMEMLOCK`** prevent Memory Swapping to Disk, so secrets are not written in plain text to disk
- The Vault Encryption Key is stored in memory in **PLAIN TEXT**
    - This is done for performance
    - Root access to an unlocked vault server could compromise this
    - Isolation technologies which allow reading of memory could compromise this (VM issues, but principally Kubernetes)
- Master Root Key protects the Encryption key, so it also must be secure
- Auto-Unseal is a recommended pattern as it shifts the risk profile

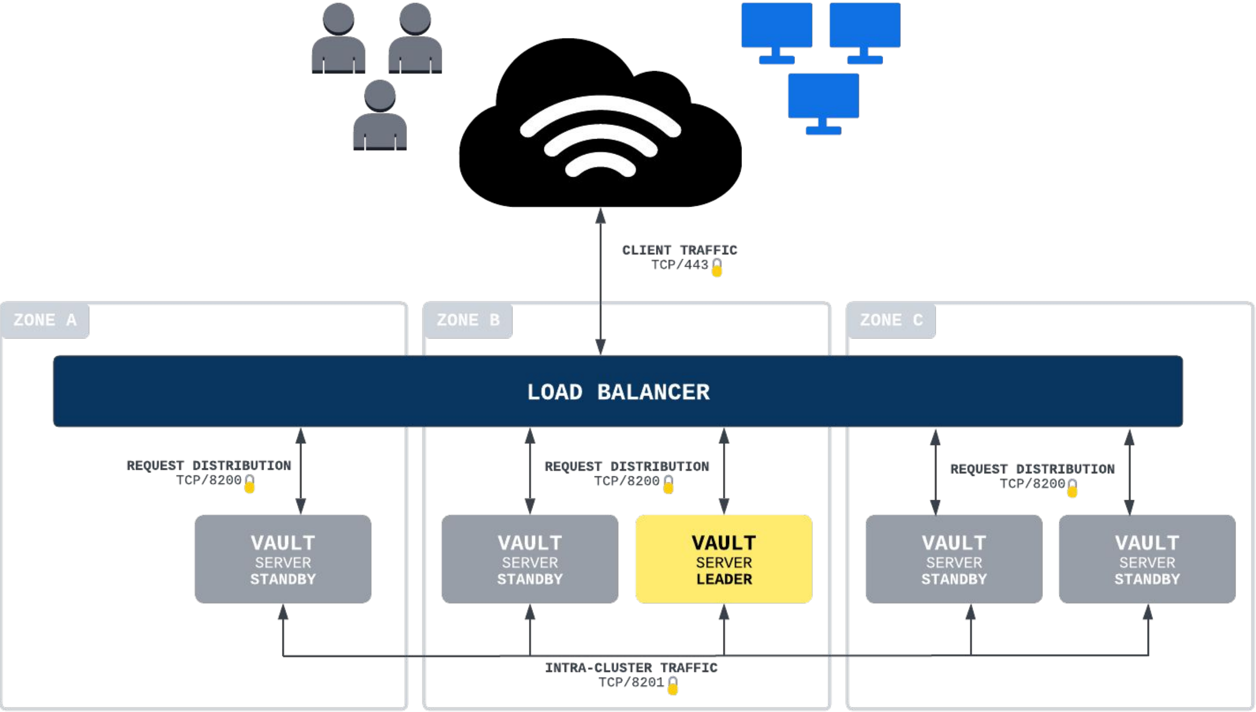
# Cryptography Security Model

- **Vault uses publicly available cryptographic technologies**
- P vs NP - Good cryptographic algorithms are exponential in difficulty to solve but polynomial in difficulty to validate answers for
- Numerous algorithms (SHA1) were exposed to have defects that allowed them, or a subset of them to be reduced to polynomial difficulty problems
- Short encryption keys and faster computers has made brute-forcing older encryption standards possible
- Software based random number generations suffer from a lack of randomness

02

# Architecture

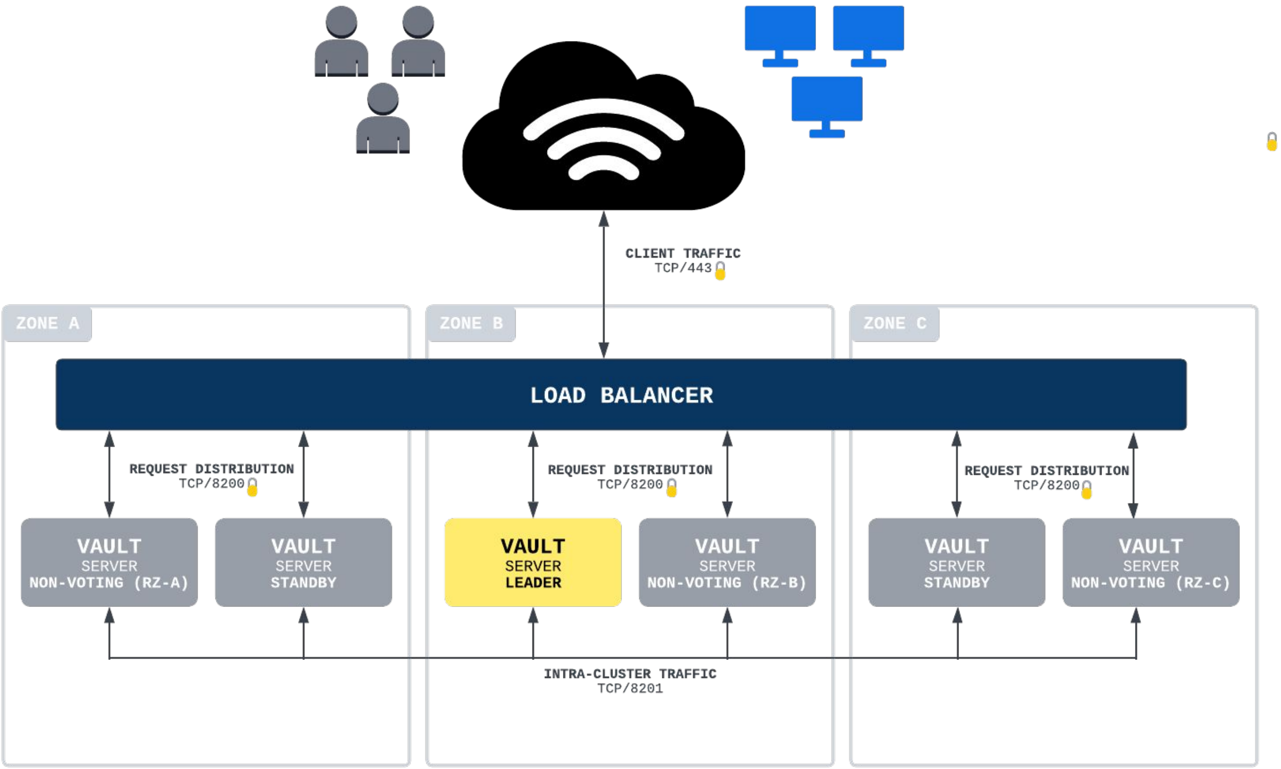
# Integrated Storage Reference Architecture



# Vault Integrated Storage Architecture

- **Integrated Storage Autopilot**
  - Monitors node health status
  - Server stabilization - prevent quorum disruption from an unstable node
  - Dead server cleanup
  - Enabled by default in Vault 1.7.0 and higher
- **Vault 1.11.0+ new features**
  - Automated upgrades promotes new versioned nodes to voter nodes removed old versioned nodes
  - Redundancy zones allows for deployment of non-voter nodes in an AZ with automatic promotion if a node is lost

# Integrated Storage with Redundancy Zones



# Sizing

Per instance sizing recommendations

| | Small<br>(Dev/Test/Staging/QA) | Large<br>(Production) |
|---|---|---|
| **CPU** | 2 - 4 Core | 4 - 8 Core |
| **Memory** | 8 - 16 GB RAM | 32 - 64 GB RAM |
| **Disk Capacity** | 100+ GB | 200+ GB |
| **Disk IO** | 3000+ IOPS | 10000+ IOPS |
| **Disk Throughput** | 75+ MB/s | 250+ MB/s |
| **AWS** | m5.large,<br>m5.xlarge | m5.2xlarge,<br>m5.4xlarge |
| **Azure** | standard_d2s_v3,<br>standard_d4s_v3 | standard_d8s_v3,<br>standard_d16s_v3 |
| **GCP** | n2-standard-2,<br>n2-standard-4 | n2-standard-8,<br>n2-standard-16 |

# Performance Considerations

## Profile Workloads

- As Vault adoption scales throughout an organization there will be varying workloads utilizing Vault
- Different workloads have varying impacts to resources (RAM, CPU, I/O)
- Leverage telemetry monitoring to ensure an understanding of implications to Vault Cluster resources usage
- As new applications/services/teams/users are onboarded to Vault, profile the usage patterns to ensure optimal authentication and consumption patterns are used

# Performance Considerations

## External Systems

- Authentication Methods & Secrets Engines have external systems dependencies that can impact Vault’s ability to process requests
- Ensure telemetry is enabled on those systems and services and proactively monitor for performance issues

# Networking Considerations

**Integrated Storage is network latency dependent**

- <8ms RT network connection required to ensure Raft Storage remains consistent across all Vault Nodes
- Restrict communication to only required ports and CIDRs
- Standard HTTPS TLS encryption should be used to protect network traffic

# Networking Requirements

| 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 |

# Load Balancing

**Vault does not include built in load balancing capabilities**

- To ensure Vault availability and reliability either an external load balancer or Consul should be used to distribute client requests
- **TLS should terminate at Vault** and not the load balancer to ensure end-to-end encryption
- Use Vault's health endpoint to determine active node and node health https://<vaultnode>:8200/v1/sys/health

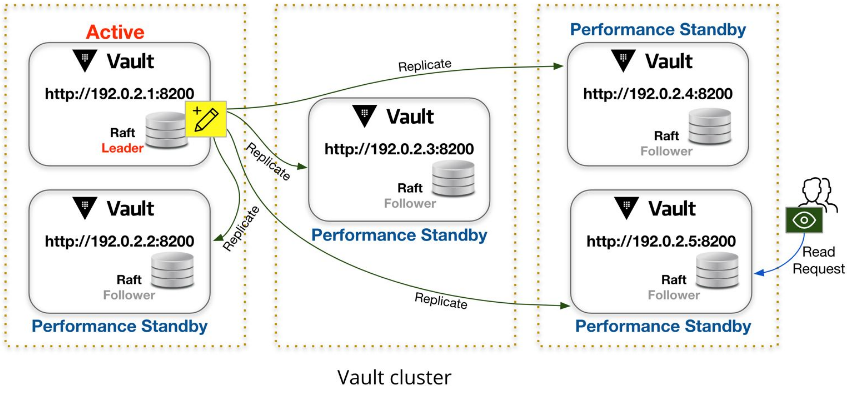
# Scaling Considerations

Managed scaling services should be leveraged when deploying in a cloud environment to ensure the Vault cluster remains populated with health nodes

- Additional nodes will not increase performance
- Do not replace all instances at once in a scaling group otherwise data-loss will occur

| Cloud | Managed Auto Scaling Service |
|---|---|
| AWS | Auto Scaling Group (ASG) |
| Azure | Virtual Machine Scale Set (VMSS) |
| GCP | Managed Instance Group (MIG) |

# Performance Standby Nodes

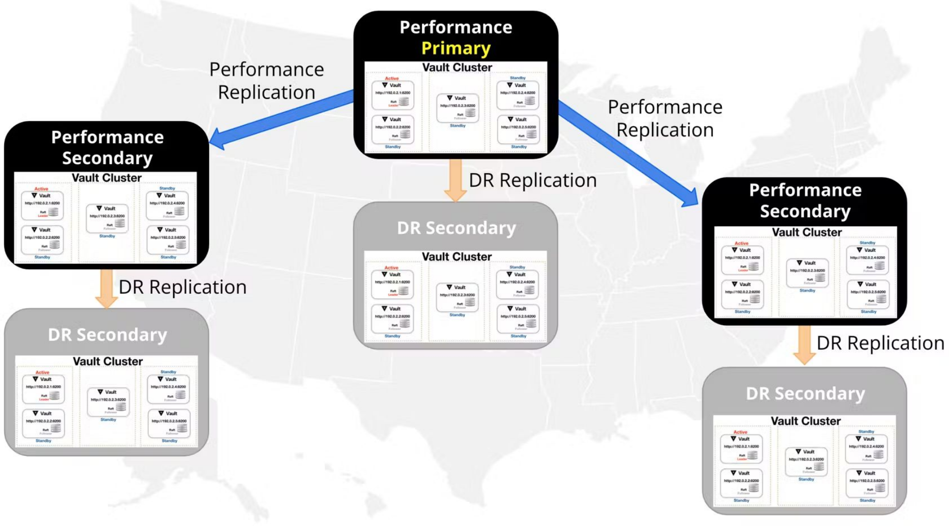


## Horizontal scalability for read requests

- Performance Standby Nodes can be used to respond to read-only requests
- Performance Standby Nodes are enabled by default and process read-only requests locally
- Write requests are forwarded to the Active Node
- Integrated Storage uses eventual consistency and data may not be available across all nodes immediately
- Vault 1.7+ includes multiple methods to control how requests are handled

# Vault Replication

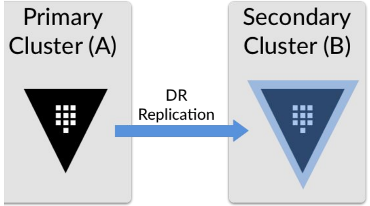
- Vault can be extended to multiple regions using replication
- The primary cluster uses asynchronous replication to ship data to the secondaries
- Multiple replication modes can be combined to provide resilience and performance



# Replication Types

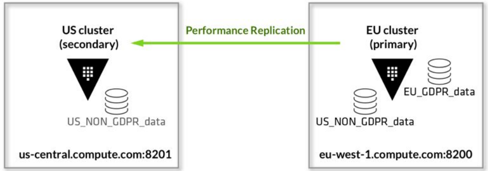
## Disaster Recovery Replication

- Active-Passive model that provides a warm standby cluster containing all data from the primary Vault cluster
- Valuable tool to achieve RPO/RTO requirements
- **It is strongly recommended to deploy at least one DR cluster**



## Performance Replication

- Provides an active Vault cluster with shared state of the primary
- Replicates secrets, auth methods, policies, & other shared data, tokens & leases are not replicated
- Common use cases include: latency reduction, compliance & data sovereignty, segmentation of workload types



03

# Deployment Patterns

# Recommended Patterns

## Immutable Builds

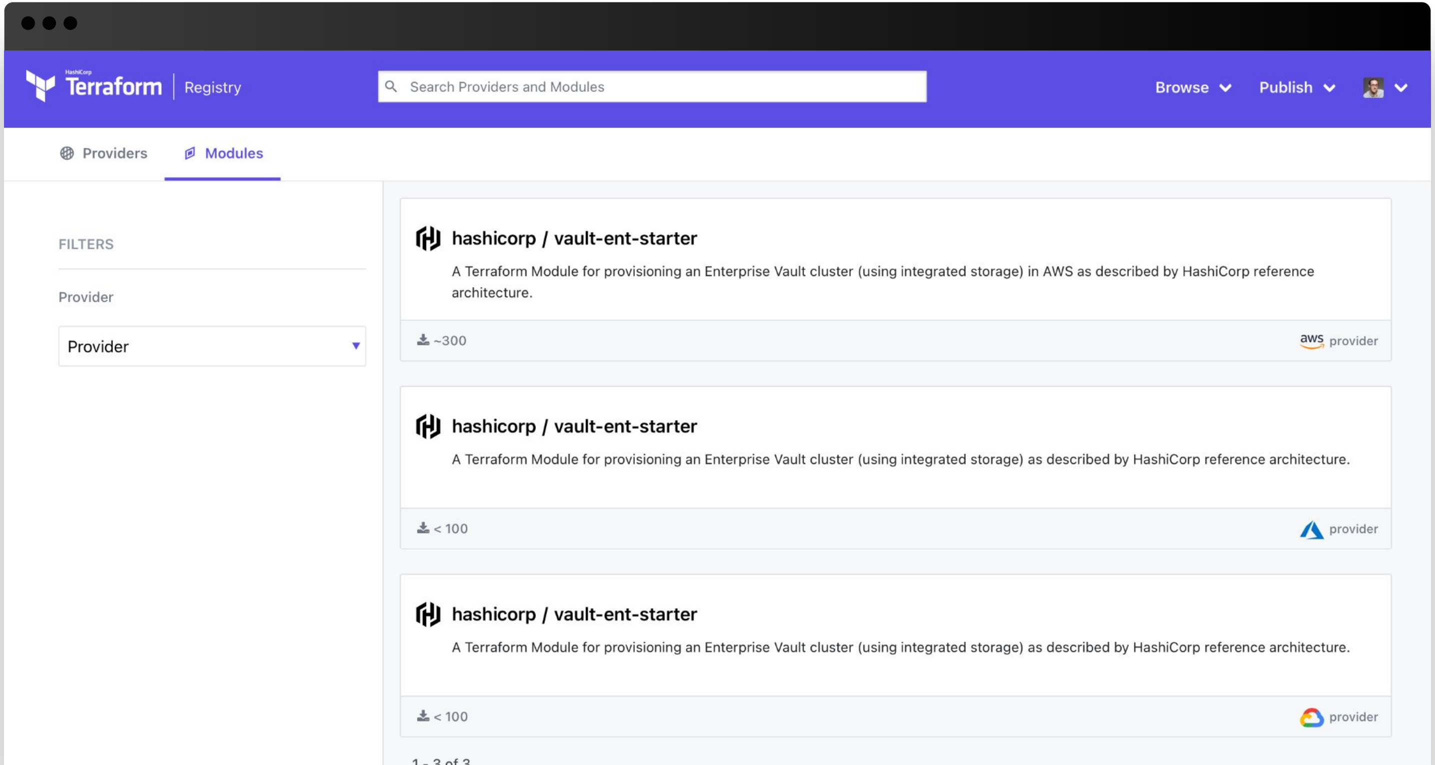
- Tools like Packer can be used to build immutable machine for blue/green deployment using existing CI/CD orchestration
- This can streamline lifecycle processes for managing Vault
- When using this pattern with Integrated Storage, ensure measures are taken to ensure quorum is maintained as new image versions are introduced to the cluster

## Configuration Management

- Configuration Management tools and patterns can be used  for installation, upgrade, and configuration of Vault
- Autopilot can be leveraged for in-place upgrades (Vault 1.11.0+)

# Terraform Modules

Quickly deploy Vault cluster based on reference architecture



**hashicorp / vault-ent-starter**

A Terraform Module for provisioning an Enterprise Vault cluster (using integrated storage) in AWS as described by HashiCorp reference architecture.

~300 · aws provider

**hashicorp / vault-ent-starter**

A Terraform Module for provisioning an Enterprise Vault cluster (using integrated storage) as described by HashiCorp reference architecture.

< 100 · provider

**hashicorp / vault-ent-starter**

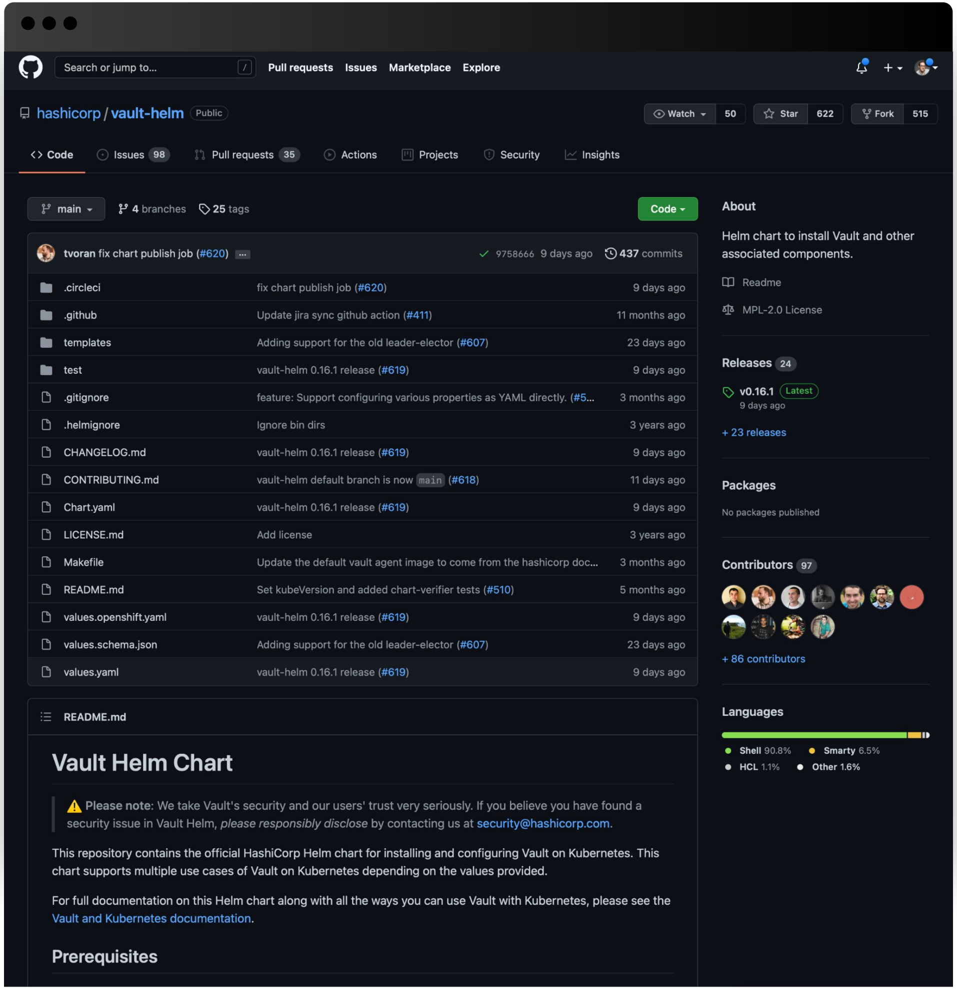
A Terraform Module for provisioning an Enterprise Vault cluster (using integrated storage) as described by HashiCorp reference architecture.

< 100 · provider

1 - 3 of 3

# Vault Helm Chart

[Deploy Vault Reference Architecture inside Kubernetes]



tvoran fix chart publish job (#620) ... ✓ 9758666 9 days ago 437 commits

| | | |
|---|---|---|
| .circleci | fix chart publish job (#620) | 9 days ago |
| .github | Update jira sync github action (#411) | 11 months ago |
| templates | Adding support for the old leader-elector (#607) | 23 days ago |
| test | vault-helm 0.16.1 release (#619) | 9 days ago |
| .gitignore | feature: Support configuring various properties as YAML directly. (#5... | 3 months ago |
| .helmignore | Ignore bin dirs | 3 years ago |
| CHANGELOG.md | vault-helm 0.16.1 release (#619) | 9 days ago |
| CONTRIBUTING.md | vault-helm default branch is now `main` (#618) | 11 days ago |
| Chart.yaml | vault-helm 0.16.1 release (#619) | 9 days ago |
| LICENSE.md | Add license | 3 years ago |
| Makefile | Update the default vault agent image to come from the hashicorp doc... | 3 months ago |
| README.md | Set kubeVersion and added chart-verifier tests (#510) | 5 months ago |
| values.openshift.yaml | vault-helm 0.16.1 release (#619) | 9 days ago |
| values.schema.json | Adding support for the old leader-elector (#607) | 23 days ago |
| values.yaml | vault-helm 0.16.1 release (#619) | 9 days ago |

README.md

## Vault Helm Chart

> ⚠ **Please note**: We take Vault's security and our users' trust very seriously. If you believe you have found a security issue in Vault Helm, *please responsibly disclose* by contacting us at security@hashicorp.com.

This repository contains the official HashiCorp Helm chart for installing and configuring Vault on Kubernetes. This chart supports multiple use cases of Vault on Kubernetes depending on the values provided.

For full documentation on this Helm chart along with all the ways you can use Vault with Kubernetes, please see the Vault and Kubernetes documentation.

## Prerequisites

**About**

Helm chart to install Vault and other associated components.

Readme

MPL-2.0 License

**Releases** 24

v0.16.1 Latest
9 days ago

+ 23 releases

**Packages**

No packages published

**Contributors** 97

+ 86 contributors

**Languages**

● Shell 90.8% ● Smarty 6.5%
● HCL 1.1% ● Other 1.6%

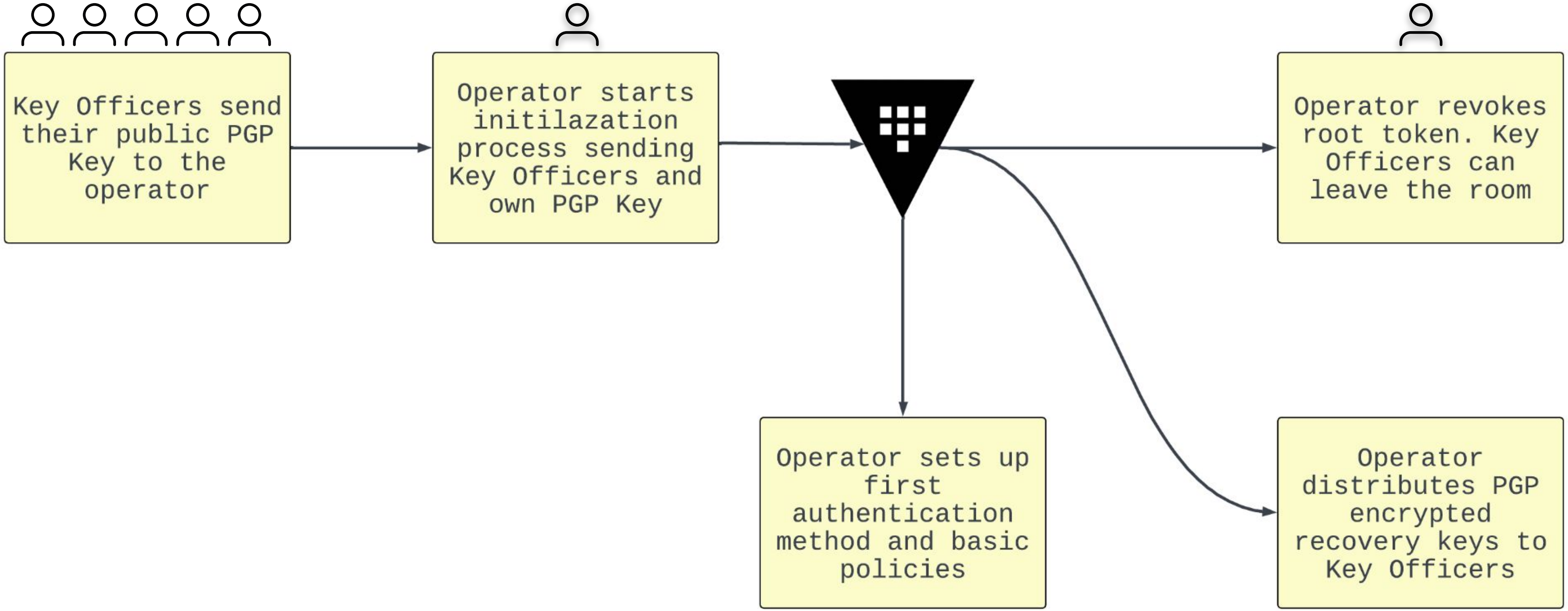
# Upgrades

- Major upgrades should occur **at least 2X per year** to stay within **N-2 major releases** version support window
- Automation of the update process is recommended to ensure ease of operations and keep Consul patched with current updates
- Prior to a production upgrade:
  - Review version specific upgrade guide
  - Review changelog
  - Test version in QA environment
  - Take a snapshot prior to any upgrade

04

# Operations

# Vault Cluster Initialization



Key Officers send their public PGP Key to the operator

Operator starts initilazation process sending Key Officers and own PGP Key

Operator revokes root token. Key Officers can leave the room

Operator sets up first authentication method and basic policies

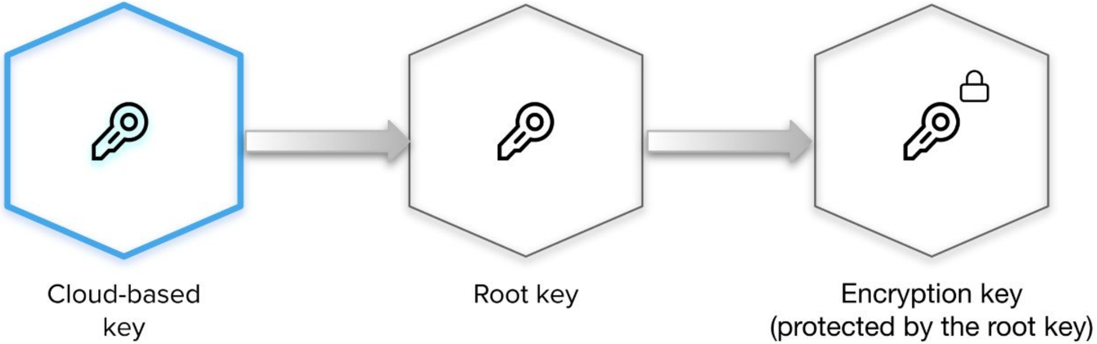
Operator distributes PGP encrypted recovery keys to Key Officers

# Auto-unseal

Unsealing is the process of constructing the master key necessary to decrypt the data encryption key because by default, Vault needs to be unsealed before any operation can be performed

Vault supports auto-unseal from:

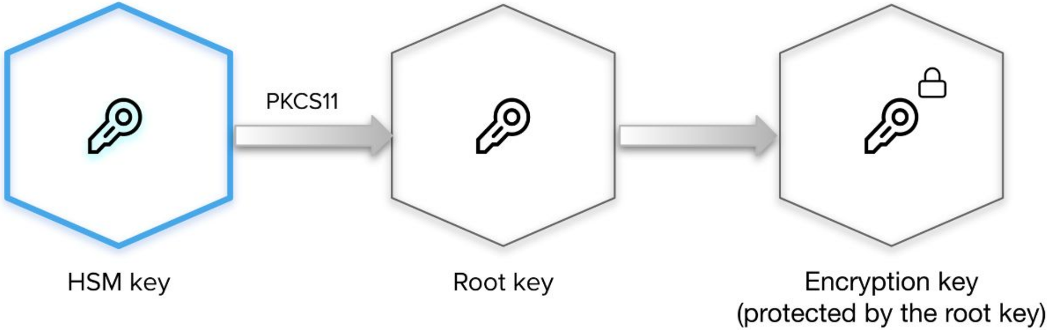
- HSM
- AliCloud KMS
- AWS KMS
- Azure Key Vault
- Google Cloud KMS
- OCI KMS



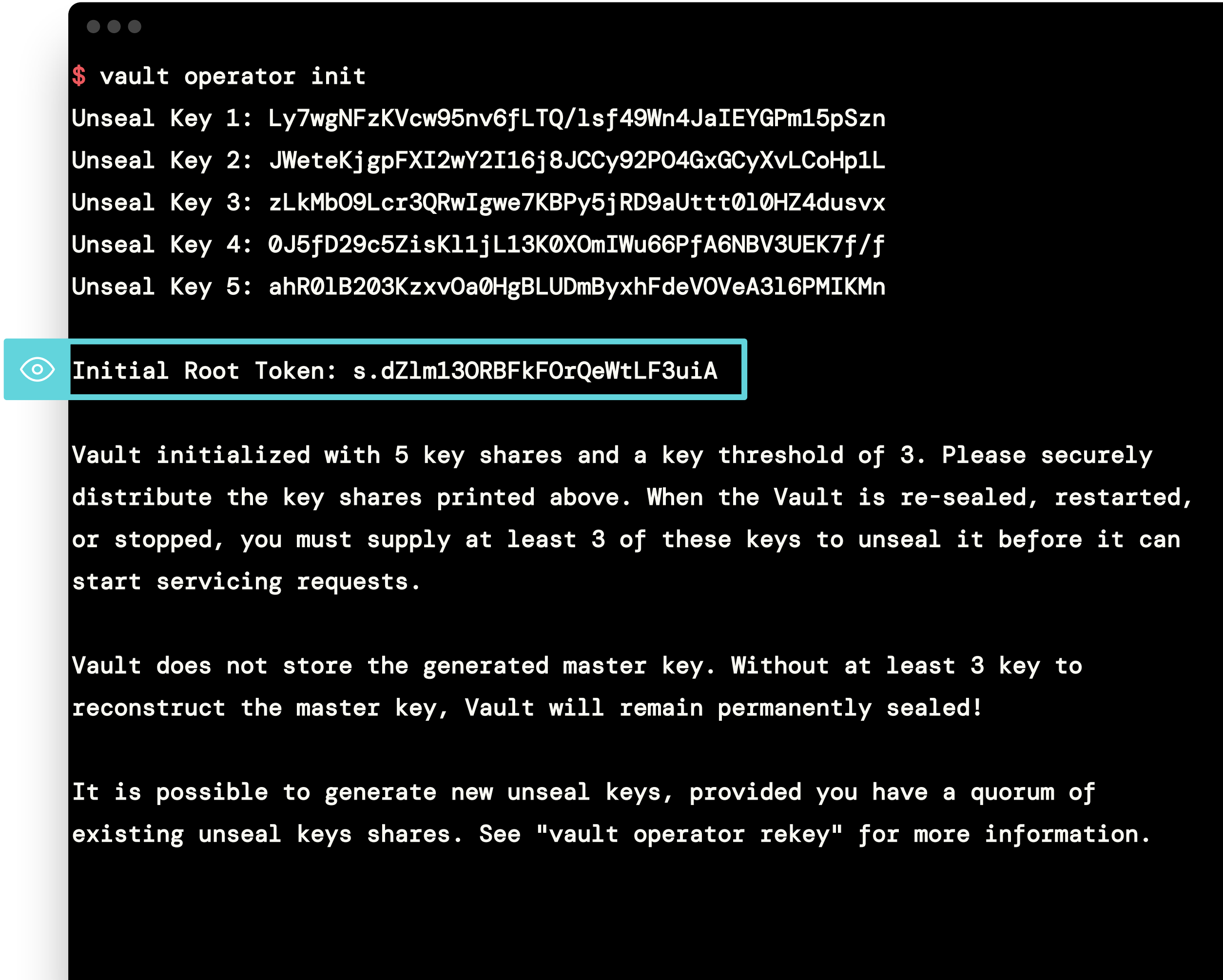
# HSM Integration

- Integrate Vault with FIPS 140-2 certified HSM (Hardware Security Module) and enable the Seal Wrap feature to protect your data.

- Vault encrypts secrets using 256-bit AES in GCM mode with a randomly generated nonce prior to writing them to its persistent storage. When you enable seal wrap, Vault wraps your secrets with an extra layer of encryption leveraging the HSM encryption and decryption.



# Root Token Generation



```
$ vault operator init
Unseal Key 1: Ly7wgNFzKVcw95nv6fLTQ/lsf49Wn4JaIEYGPm15pSzn
Unseal Key 2: JWeteKjgpFXI2wY2I16j8JCCy92PO4GxGCyXvLCoHp1L
Unseal Key 3: zLkMbO9Lcr3QRwIgwe7KBPy5jRD9aUttt0l0HZ4dusvx
Unseal Key 4: 0J5fD29c5ZisKl1jL13K0XOmIWu66PfA6NBV3UEK7f/f
Unseal Key 5: ahR0lB203KzxvOa0HgBLUDmByxhFdeVOVeA3l6PMIKMn

Initial Root Token: s.dZlm13ORBFkFOrQeWtLF3uiA

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 master key. Without at least 3 key to
reconstruct the master 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.
```

# Root Token Handling Practices

The root token is returned to the operator during the initialization ceremony. This token can do **anything** in Vault and its usage should be closely monitored.

- Once operator has configured a secondary authentication method and granted administrators sudo access, almost all operations can be performed
- Best practice is **NOT** persisting the root token
- Generate a root token only when absolutely necessary

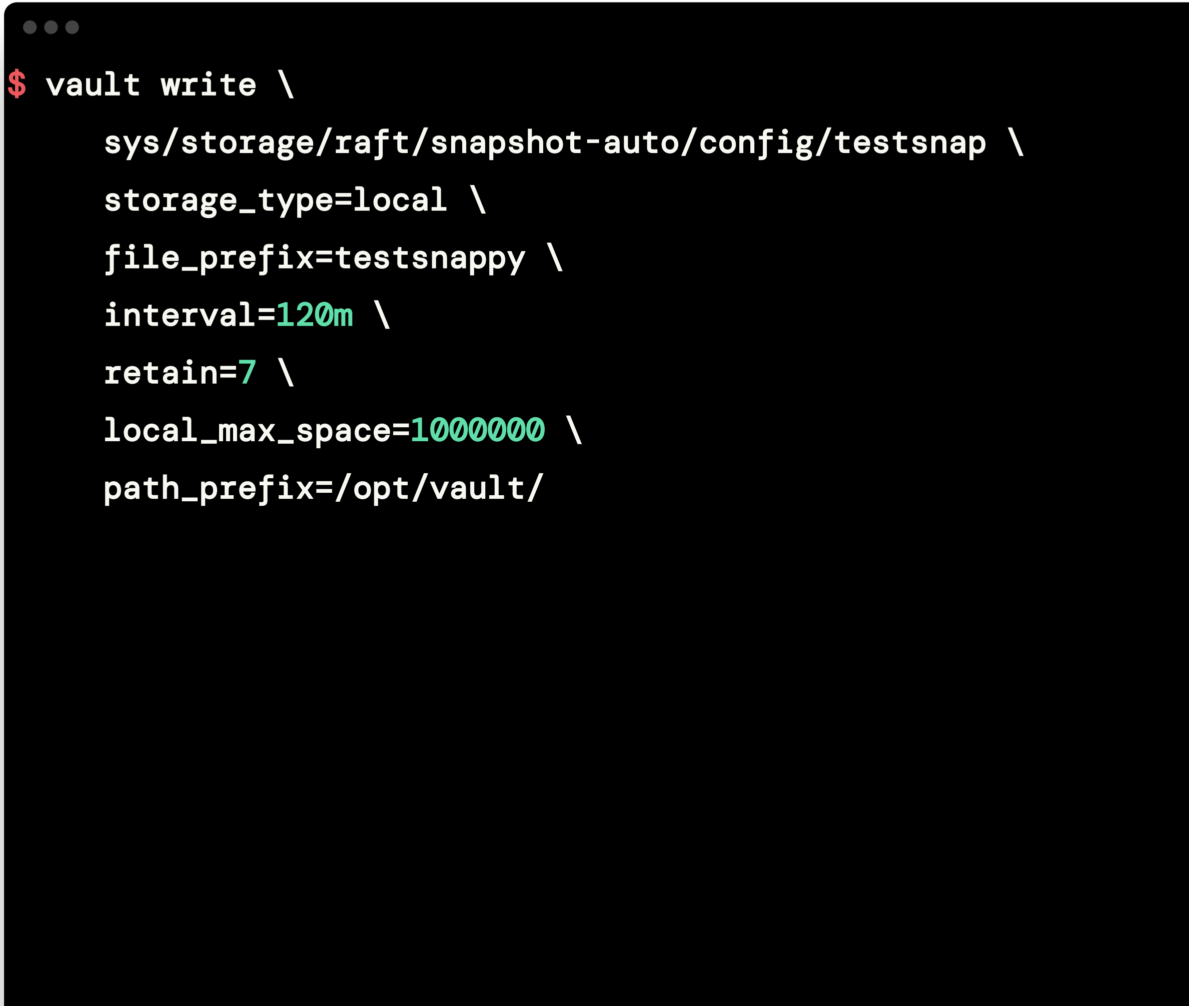
# Backup

Critical item to have in place before production go-live

Automated Snapshots, a Vault Enterprise feature takes periodic snapshots of Vault’s data

- Determine where snapshot files will be stored
- Configure based off your RPO/RTO requirements
- If snapshot is restored, the unseal keys that were valid at the time of the snapshot will be used to unseal

# Automated Integrated Storage Snapshots



```
$ vault write \
    sys/storage/raft/snapshot-auto/config/testsnap \
    storage_type=local \
    file_prefix=testsnappy \
    interval=120m \
    retain=7 \
    local_max_space=1000000 \
    path_prefix=/opt/vault/
```

# Monitoring

Critical item to have in place before production go-live

Vault should be monitored closely to ensure the service remains healthy and available in production

- Telemetry - Export telemetry data to solution that can analyze and identify trends overtime
- Log Analytics - Capture app logs and system logs and perform analysis on the log files for useful signals
- Active Health Checks - Query health endpoints to get the health of nodes and route traffic to active node

# Audit Logs

Critical item to have in place before production go-live

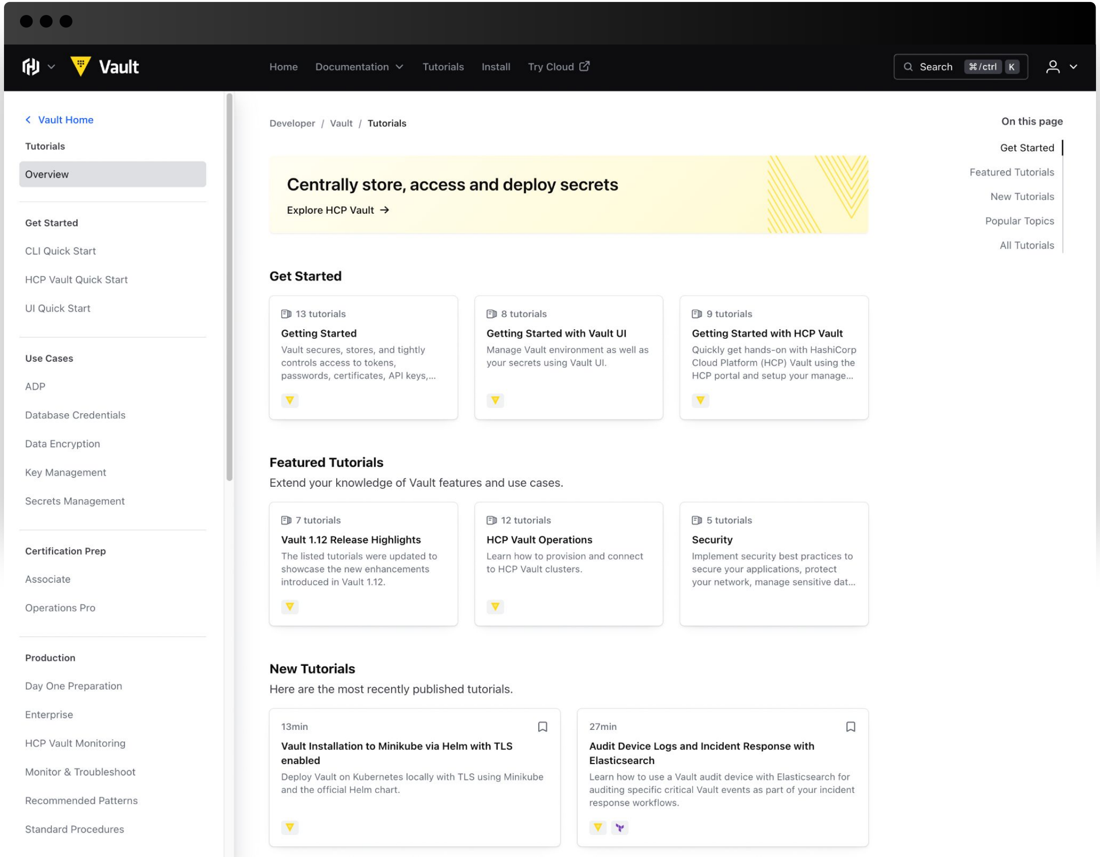
Vault sends audit information to a SIEM system or logging backend

- Determine audit devices that will be used
- Vault will not respond if the audit device is unavailable, use multiple audit devices to ensure Vault remains available
- Sensitive fields are HMAC, Selectively determine if any HMAC fields need to be exposed

# Next Steps

# Tutorials

Step-by-step guides to accelerate deployment of Vault



< Vault Home

Tutorials

Overview

Get Started

CLI Quick Start

HCP Vault Quick Start

UI Quick Start

Use Cases

ADP

Database Credentials

Data Encryption

Key Management

Secrets Management

Certification Prep

Associate

Operations Pro

Production

Day One Preparation

Enterprise

HCP Vault Monitoring

Monitor & Troubleshoot

Recommended Patterns

Standard Procedures

Developer / Vault / Tutorials

Centrally store, access and deploy secrets

Explore HCP Vault →

On this page

Get Started

Featured Tutorials

New Tutorials

Popular Topics

All Tutorials

Get Started

13 tutorials
**Getting Started**
Vault secures, stores, and tightly controls access to tokens, passwords, certificates, API keys,...

8 tutorials
**Getting Started with Vault UI**
Manage Vault environment as well as your secrets using Vault UI.

9 tutorials
**Getting Started with HCP Vault**
Quickly get hands-on with HashiCorp Cloud Platform (HCP) Vault using the HCP portal and setup your manage...

Featured Tutorials

Extend your knowledge of Vault features and use cases.

7 tutorials
**Vault 1.12 Release Highlights**
The listed tutorials were updated to showcase the new enhancements introduced in Vault 1.12.

12 tutorials
**HCP Vault Operations**
Learn how to provision and connect to HCP Vault clusters.

5 tutorials
**Security**
Implement security best practices to secure your applications, protect your network, manage sensitive dat...

New Tutorials

Here are the most recently published tutorials.

13min
**Vault Installation to Minikube via Helm with TLS enabled**
Deploy Vault on Kubernetes locally with TLS using Minikube and the official Helm chart.

27min
**Audit Device Logs and Incident Response with Elasticsearch**
Learn how to use a Vault audit device with Elasticsearch for auditing specific critical Vault events as part of your incident response workflows.

https://developer.hashicorp.com/vault/tutorials

# Resources

- Vault Internal Architecture
- Vault Security Model
- Vault Reference Architecture
- Vault Redundancy Zones (1.11.0+)
- Terraform Starter Code
- Disaster Recovery Replication Setup
- Performance Replication Setup
- Vault Eventual Consistency and Controls

# 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

# Upcoming Webinars

## Using Vault with Kubernetes

This Lunch & Learn (separate link) covers the best practices for integrating Vault Enterprise with Kubernetes and

## Namespaces, Authentication, and Policies

Learn best practices for deploying Vault Namespaces, Authentication Methods, and Vault policy

## Office Hours

An open forum with Vault Subject Matter Experts to answer questions that have arisen during the program and your deployment

# Action Items

- Share to customer.success@hashicorp.com
  - Authorized technical contacts for support
  - Stakeholders contact information (name and email addresses)
- Email customer.success@hashicorp.com with a brief summary of Vault Enterprise use case(s), goals, and project milestones
- Determine Vault pattern and begin deployment of first cluster(s)

# Q&A

# Thank you

customer.success@hashicorp.com

www.hashicorp.com/customer-success