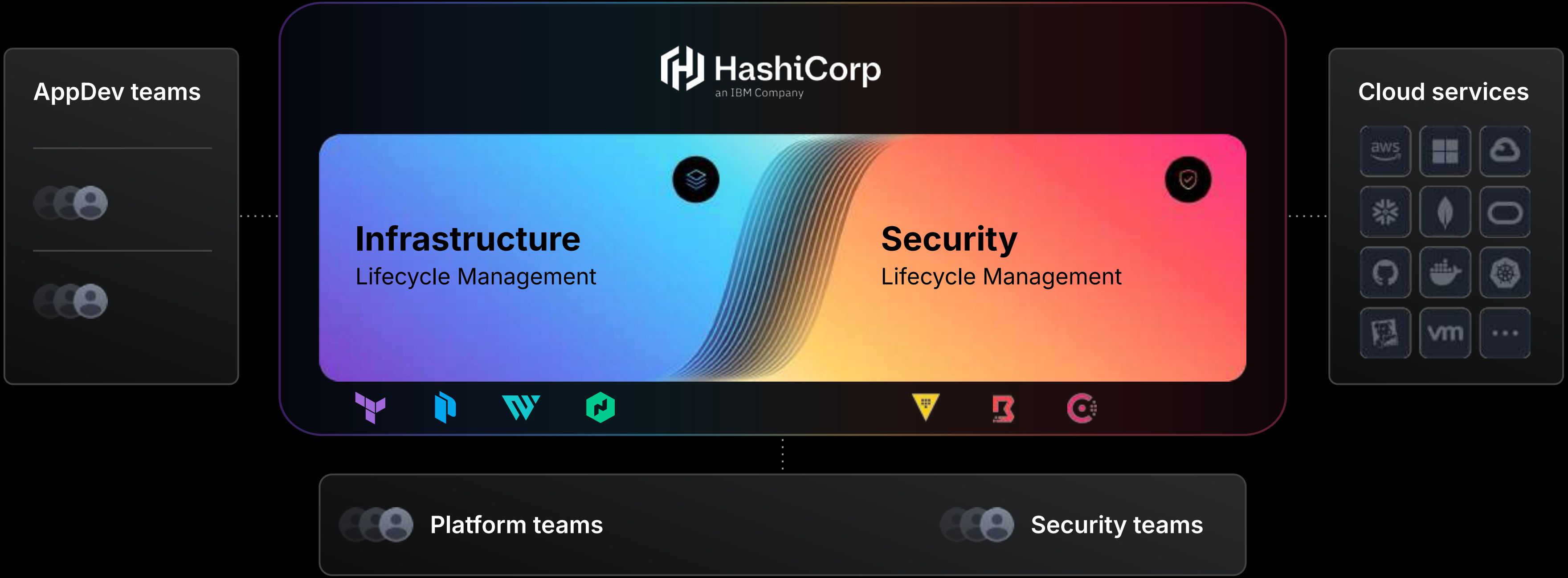
HashiCorp
an IBM Company

HashiCorp Vault and OpenShift: Discover Security and Speed in Perfect Harmony

Ruud Zwakenberg
Red Hat

Cojan van Ballegooijen
HashiCorp

Doing Cloud Right



AppDev teams

HashiCorp
an IBM Company

Infrastructure
Lifecycle Management

Security
Lifecycle Management

Cloud services

Platform teams

Security teams

SECURE PORTFOLIO

Security Lifecycle Management

HashiCorp Cloud Platform
an IBM Company

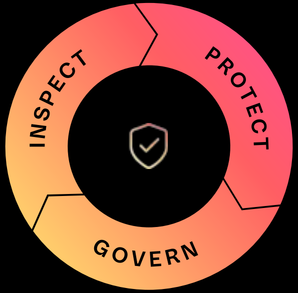
Security
Lifecycle Management

Vault
Machine identity management

Vault Radar
Secret discovery and remediation

Boundary
Secure remote access

Consul
Service-based networking



Identity-based secure access to machines, people, and services

MARKET TRENDS

Kubernetes lacks built-in security

No secret encryption

⚠ CHALLENGE

By default, secrets are stored in base64 encoded plain text, presenting targets for attackers

Default open access

⚠ CHALLENGE

Misconfigured access control can allow unauthorized entities to access secrets within the namespace

Manual key rotation

⚠ CHALLENGE

Manual and inconsistent key rotation can lead to stale or compromised credentials across clusters

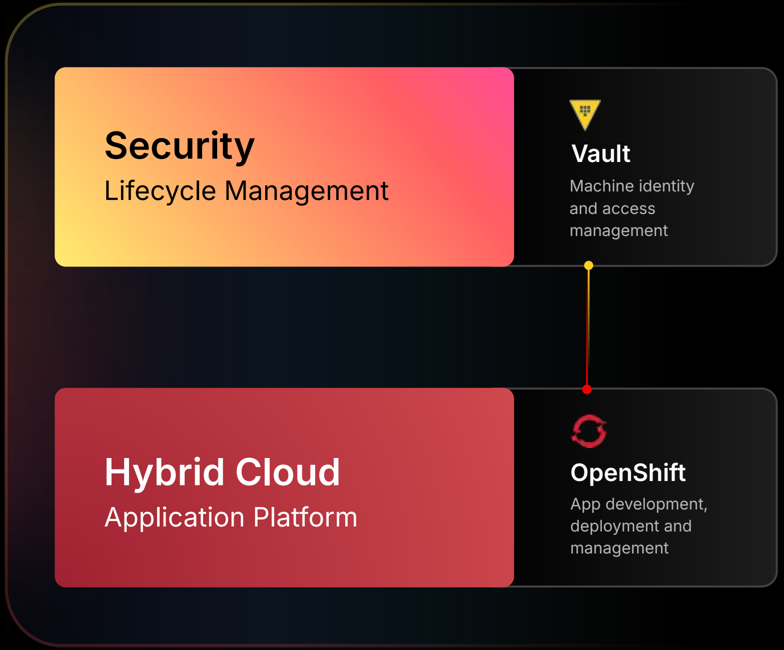
Database key storage

⚠ CHALLENGE

Unencrypted keys stored in etcd are vulnerable if the etcd database is compromised and are accessible to cluster admins

BETTER TOGETHER

Protect hybrid applications from credential theft



Security
Lifecycle Management

Vault
Machine identity and access management

Hybrid Cloud
Application Platform

OpenShift
App development, deployment and management

Reduce risk and streamline hybrid operations with Vault and OpenShift

Build, manage, and secure hybrid applications on a single platform

Enforce identity-based authorization and security policies consistently across environments

Encrypt, rotate, and inject credentials into OpenShift containers and CI/CD workflows

KEYS TO CLOUD SUCCESS

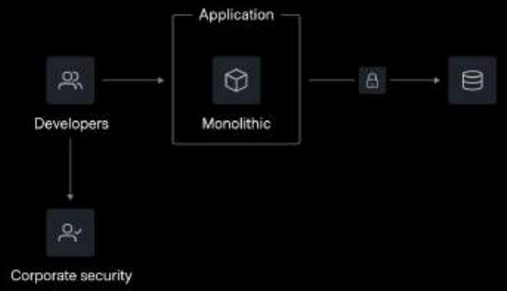
Challenges implementing lifecycle management

	DAY 0	DAY 1	DAY 2 … DAY N
	INSPECT	**PROTECT**	**GOVERN**
Secrets administrator	**Secrets architecture Is inconsistent** • No centralized secrets strategy • Duplicate workflows across clusters • Lacks namespace-level policy isolation	**Manual processes and policy gaps** • Hand-built pipelines for secret delivery • No fine-grained access controls • Lacks cross-team visibility and governance	**Manual credential handling blocks development velocity** • Manual credential rotation burdens ops • Static secrets increase exposure window • Limited auditing across environments
Developers	**Legacy secrets management doesn't scale** • Static secrets across environments multiply • No standardization across microservices • Manual duplication creates security risk	**Security requirements break dev workflows** • Rotation policies disrupt CI/CD • Secrets blocked by brittle tooling • No GitOps-native secret delivery	**Manual credential handling blocks velocity:** • Wait on secrets from ops • Frequent manual rotation required • Delays environment rebuilds and testing

Basic secrets management

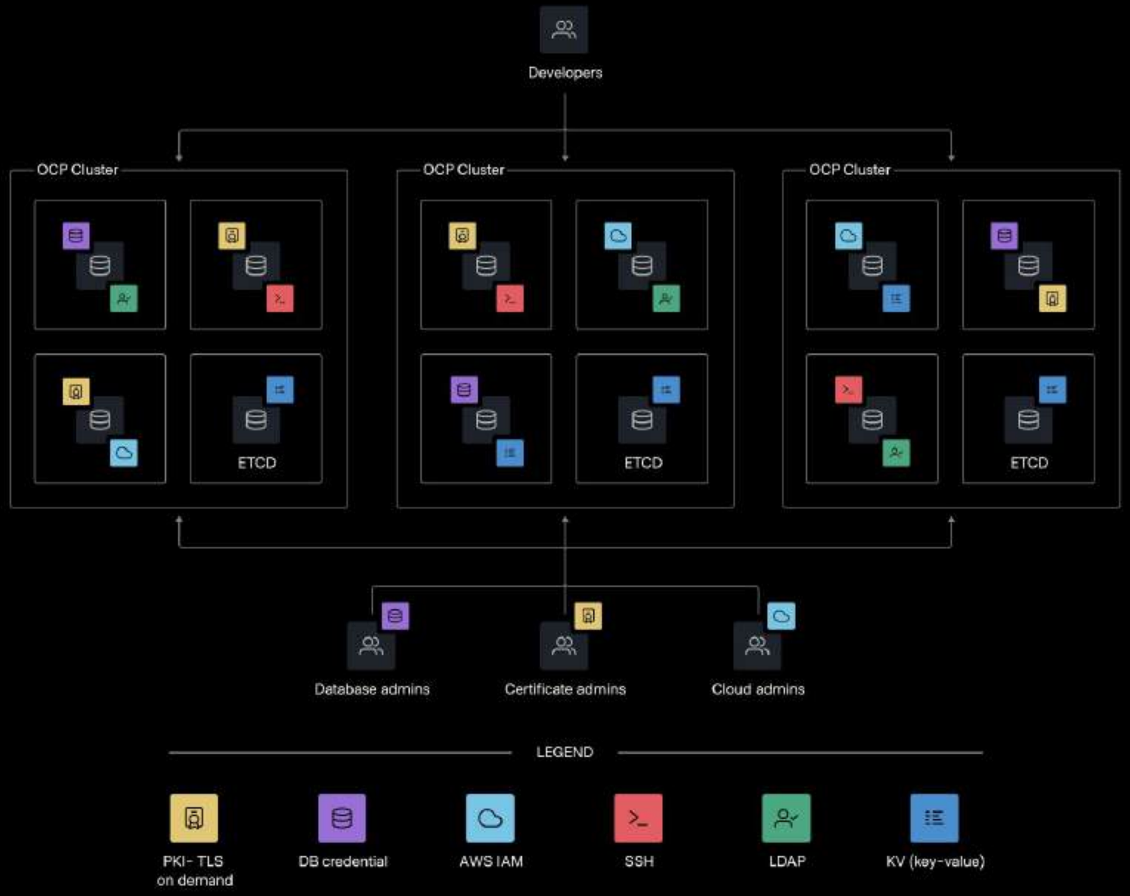
Applications require multiple forms of sensitive material. This could be database credentials for web applications, cloud credentials for access to cloud native services, or even the ability to interact with sensitive data types.

Traditional approaches are manually managed through Identity or Information Security teams to maintain chain of custody of sensitive information.



OpenShift secrets management

- Secret Management not centralized
- Administrative overhead is spread out
- Tracking down sprawl of different secrets...

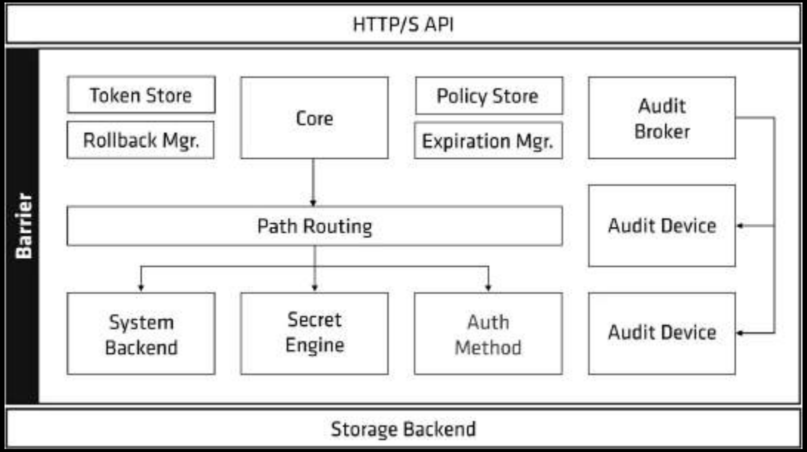


How Vault works

ARCHITECTURE

HashiCorp Vault Components

- Storage backends
- Secrets Engines
- Auth methods
- Audit devices
- HTTP/API



HTTP/S API

Barrier

Token Store

Rollback Mgr.

Core

Policy Store

Expiration Mgr.

Audit Broker

Path Routing

Audit Device

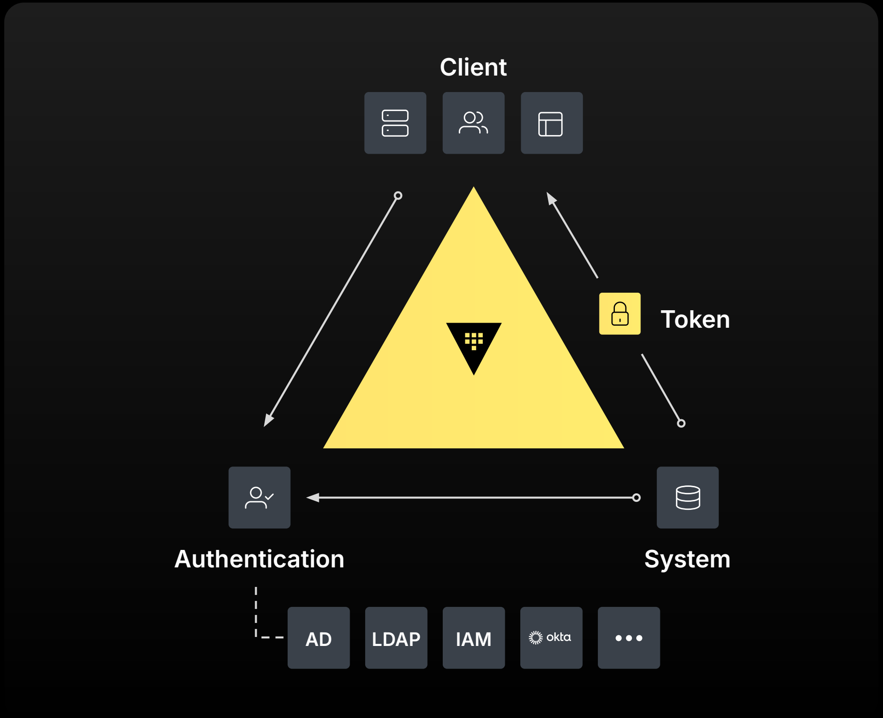
System Backend

Secret Engine

Auth Method

Audit Device

Storage Backend



Client

Token

Authentication

System

AD

LDAP

IAM

okta

•••

ARCHITECTURE

HashiCorp Vault workflow

1. A client provides credentials (ID) to Vault requesting access.

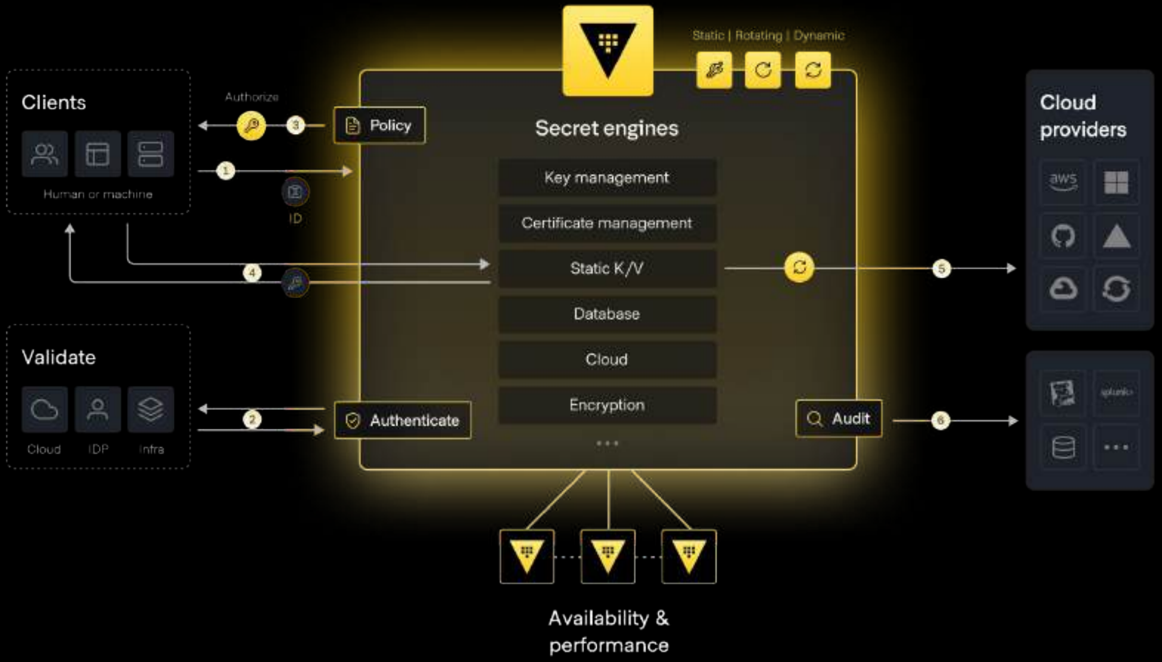
2. Vault uses authentication plugins to validate the client against the appropriate trusted third-party resource, such as GitHub, LDAP, CSP, or others.

3. Vault grants access to secrets and encryption capabilities by issuing a token tied to policies associated with the client's identity.

4. Client uses policy-based access to retrieve secrets, keys, and certificates, and perform other operations like data encryption.

5. Static secrets can be centrally managed and automatically synced to destination sources.

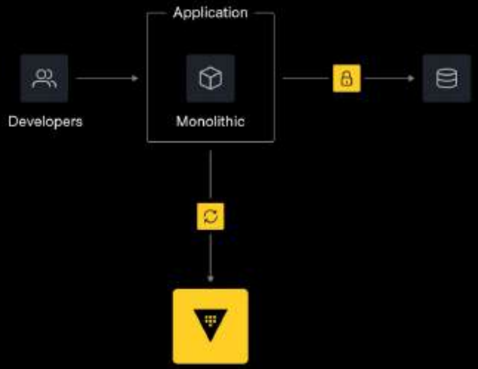
6. Detailed logs retained for monitoring and compliance.



Basic secrets management

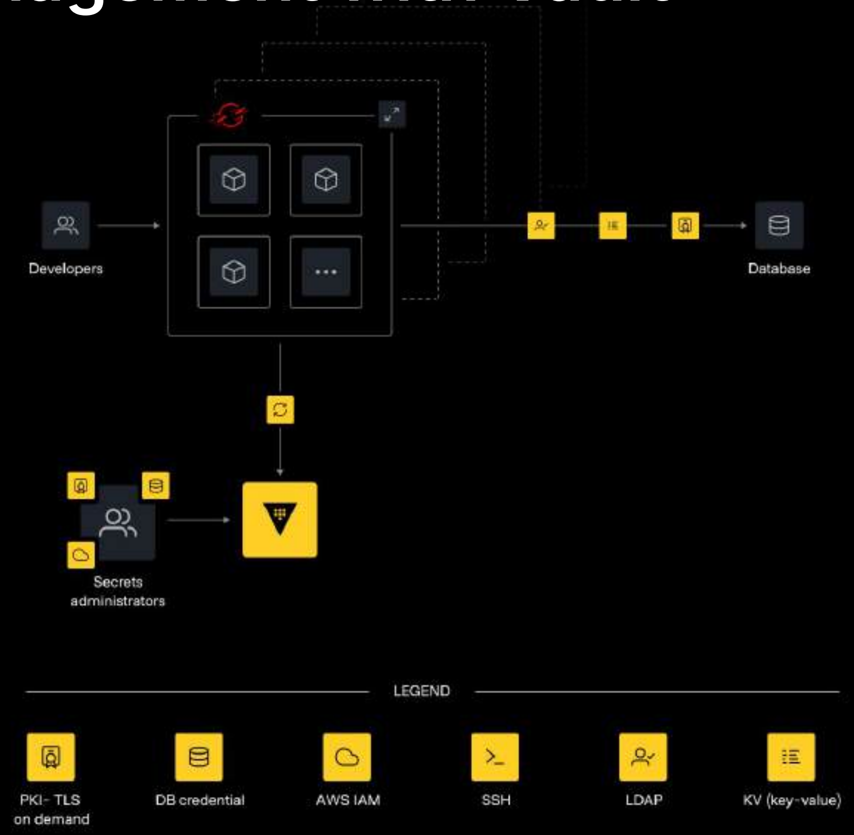
Vault can enable frequent iterative development with self-service while increasing security posture and maintaining your rigorous compliance requirements.

Decoupling the human element and integrating into common workflows reduces friction in the software development lifecycle, increases speed of delivery, and removes operational overhead.



OpenShift secrets management with Vault

- Automated secret injection at runtime
- Centralized identity and access policies
- Consistent secret delivery to all workloads



OpenShift secrets management with Vault

INTEGRATION OPTIONS

Vault Secrets Operator

- Provides secret data to Pods from synced K8s Secrets
- Secret data is cached
- Syncs Vault secret data

Vault Agent Injector

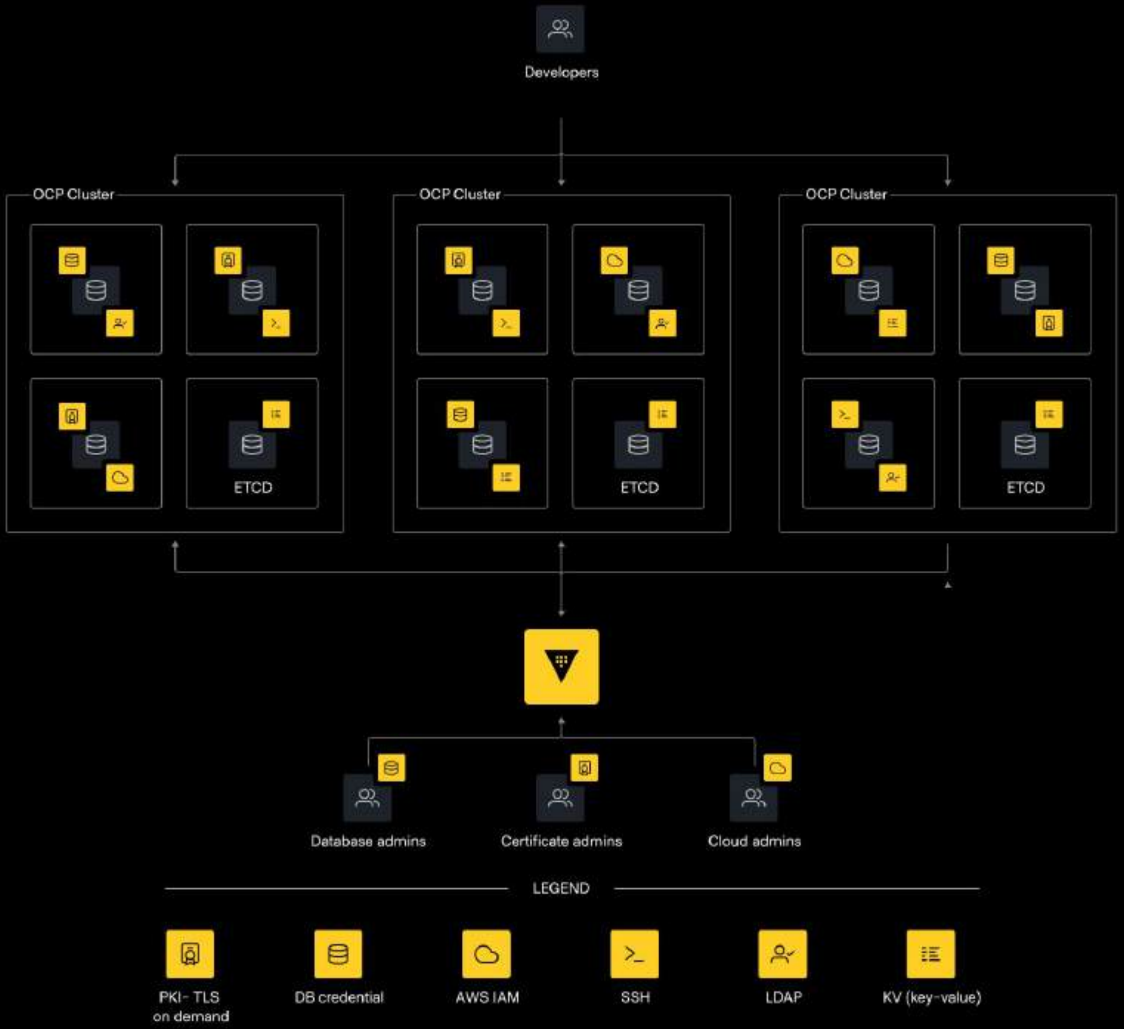
- Stores secrets in ephemeral Volumes
- Depends on Vault being up during Pod scaling
- Utilizes the agent sidecar strategy to inject secrets into Pods

Vault CSI Provider

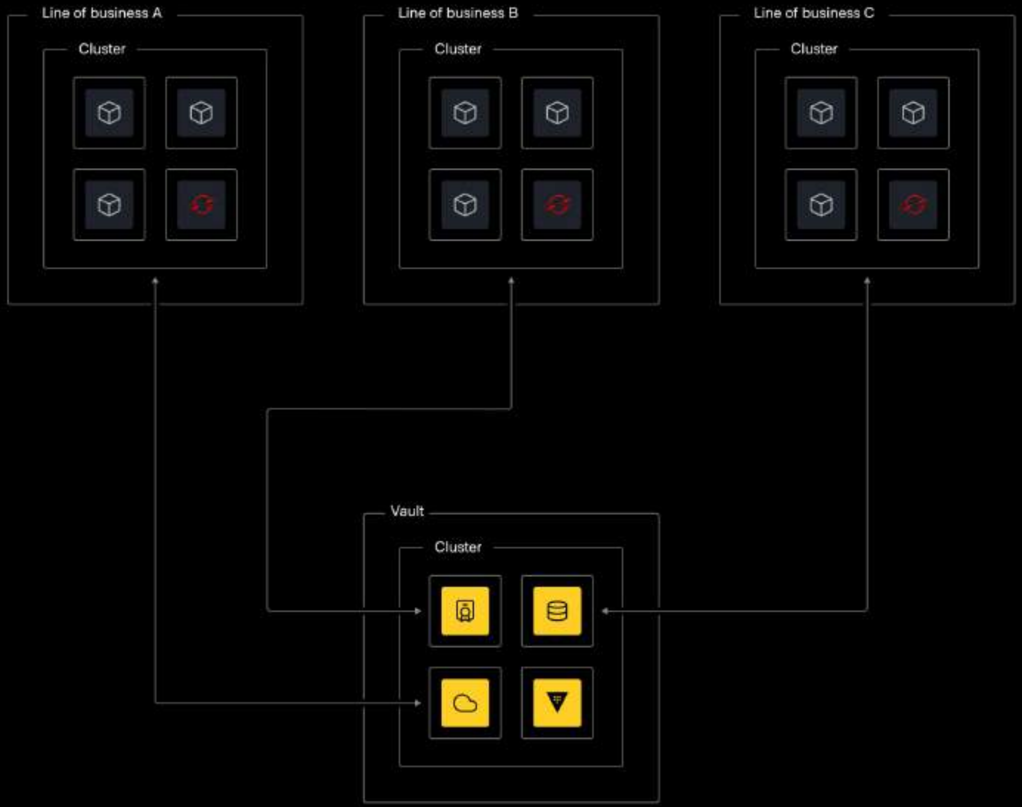
- Provides secret data to Pods using ephemeral volumes
- Depends on the CSI Secrets driver
- Depends on Vault being up during Pod scaling

OpenShift secrets management with Vault

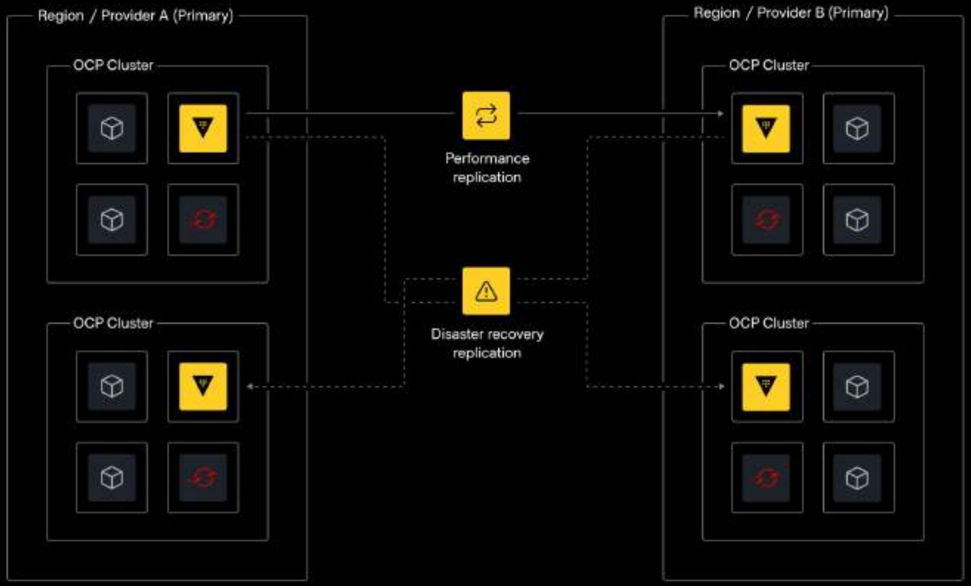
- Centralized management of secret estate
- Developers can focus on their applications
- Standardized deployment of all secrets



Secure multi-tenancy with Vault namespaces

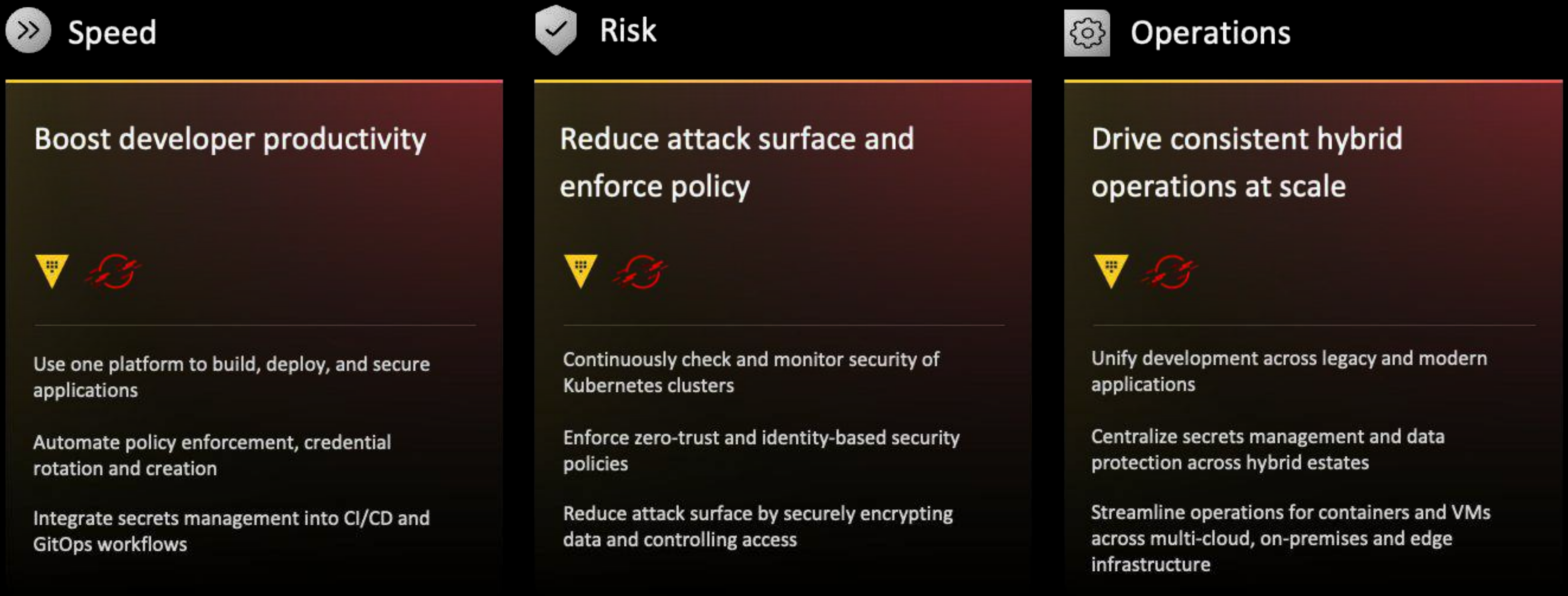


Replication patterns with OpenShift and Vault


Region / Provider A (Primary)
OCP Cluster
OCP Cluster
Performance replication
Disaster recovery replication
Region / Provider B (Primary)
OCP Cluster
OCP Cluster

Red Hat and HashiCorp

Key takeaways



Speed

Boost developer productivity

- Use one platform to build, deploy, and secure applications
- Automate policy enforcement, credential rotation and creation
- Integrate secrets management into CI/CD and GitOps workflows

Risk

Reduce attack surface and enforce policy

- Continuously check and monitor security of Kubernetes clusters
- Enforce zero-trust and identity-based security policies
- Reduce attack surface by securely encrypting data and controlling access

Operations

Drive consistent hybrid operations at scale

- Unify development across legacy and modern applications
- Centralize secrets management and data protection across hybrid estates
- Streamline operations for containers and VMs across multi-cloud, on-premises and edge infrastructure

References

- Vault Documentation
- Vault Secrets Operator
- Validated Designs
  - Vault Solution Design
  - Vault Operating Guide
    - For Adoption
    - For Standardization
    - For Scale
- Vault Validated Patterns
- Red Hat - The state of Kubernetes security report

HashiCorp
an IBM Company

Thank you