HashiCorp Vault

HCP Vault: Operationalizing for Production

June 2022

Agenda

- Automate HCP Control Plane
- Automate Vault Configuration
- Audit Log
- Telemetry
- Next Steps
- Q & A

HCP Vault Path to Production

Weeks 1-2

Vault Deployed

- Kick Off Webinar
- Architectural Deep Dive Workshop
- Office Hours

Weeks 4-5

Operational Considerations

- Codify Vault Config, Telemetry, Monitoring,
- 1:1 Success Planning

Weeks 6-7

Identity & Access Management

- Namespaces, ACL Policies, Auth Methods
- 1:1 Success Planning

Weeks 8-9

Secrets Management

- How to consume Vault into applications Workshop
- Office Hours
- 1:1 Success Planning

Weeks 10

Ready for Production!

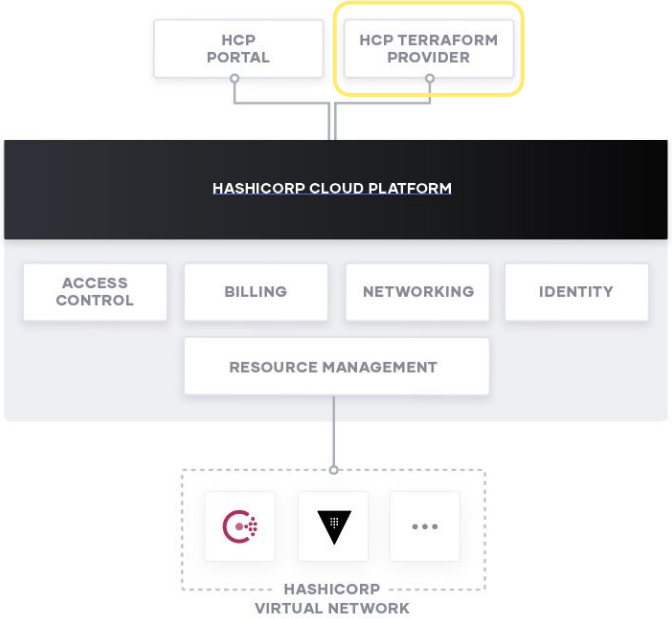
- Success Plan Reviews
- Exit Ramp & Operational Readiness Checklist

Automate HCP Control Plane

HashiCorp Cloud Platform

Overview

HashiCorp Cloud Platform supports management of the platform through web interface or it can be integrated with your automation processes by leveraging Terraform, HCP provider, and Vault provider.



Service Principals

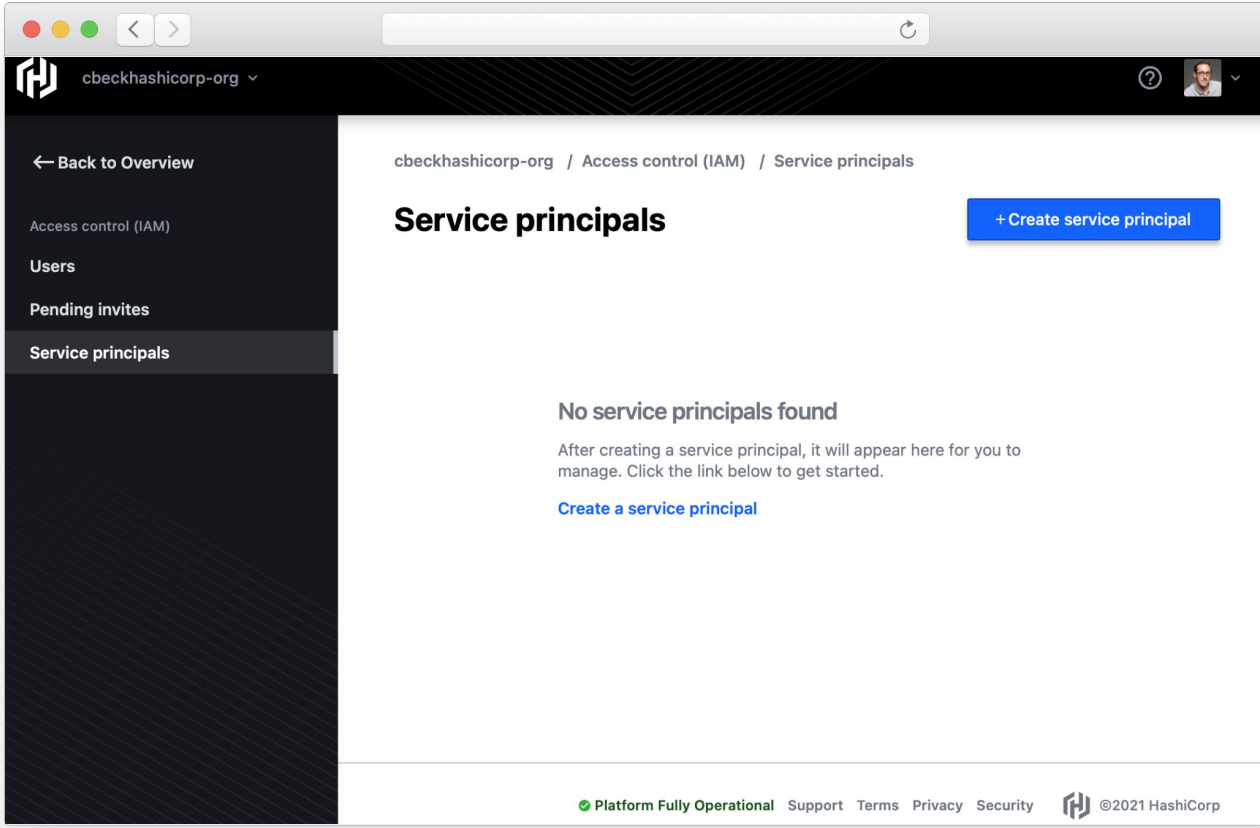
Access Controls

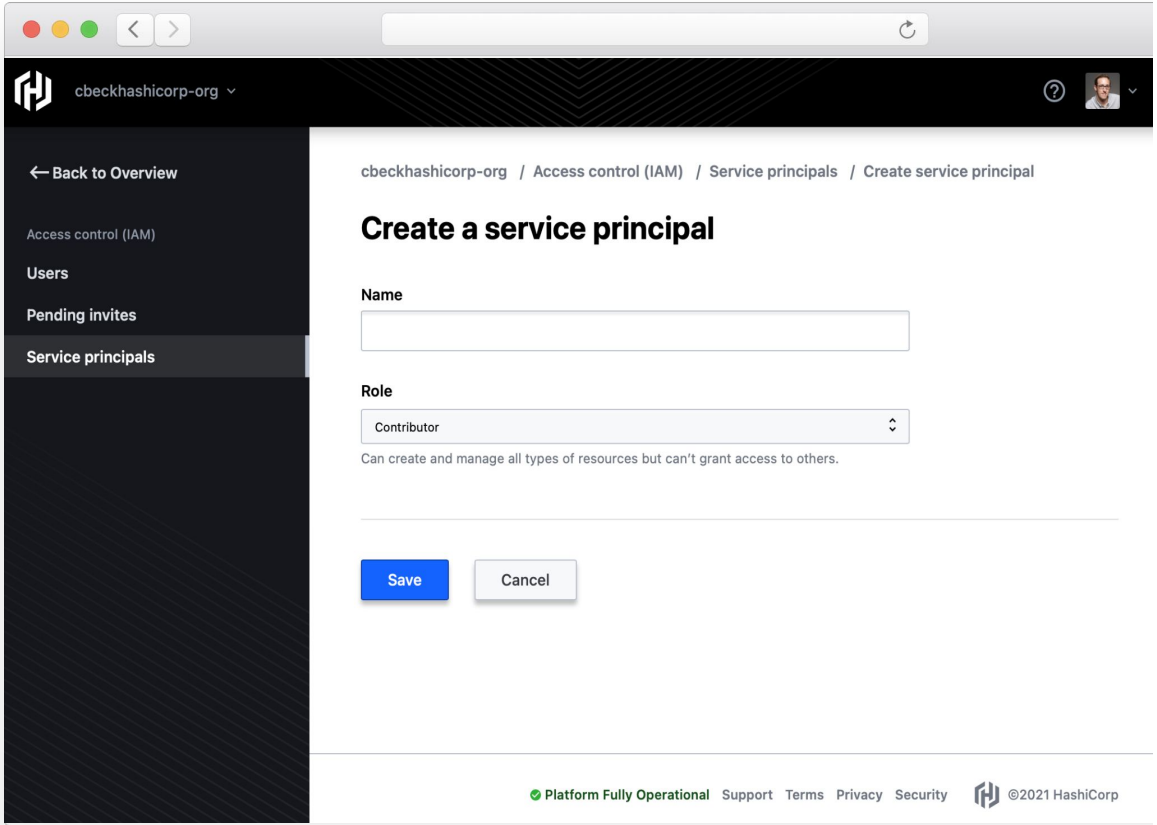
HashiCorp Cloud Platform allows you to grant access to both user and machines. Users will typically have access managed via their user principal that is tied to their identity. Non-human clients or machine users will need to be granted access using service principals.

RBAC

Both user principals and service principles can be assigned one of three roles (viewer, contributor, and admin) depending on the type of operations the user or service will need to perform.

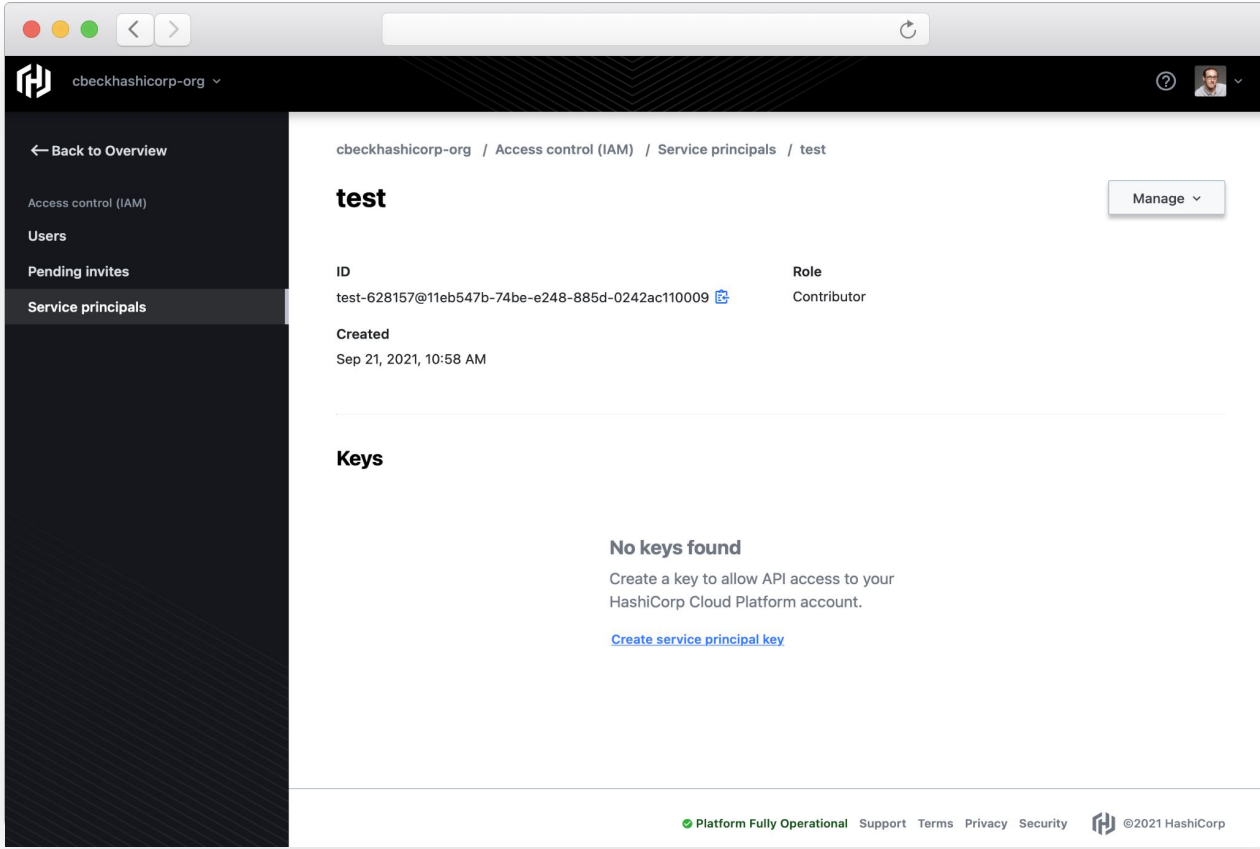
Creating Service Principals

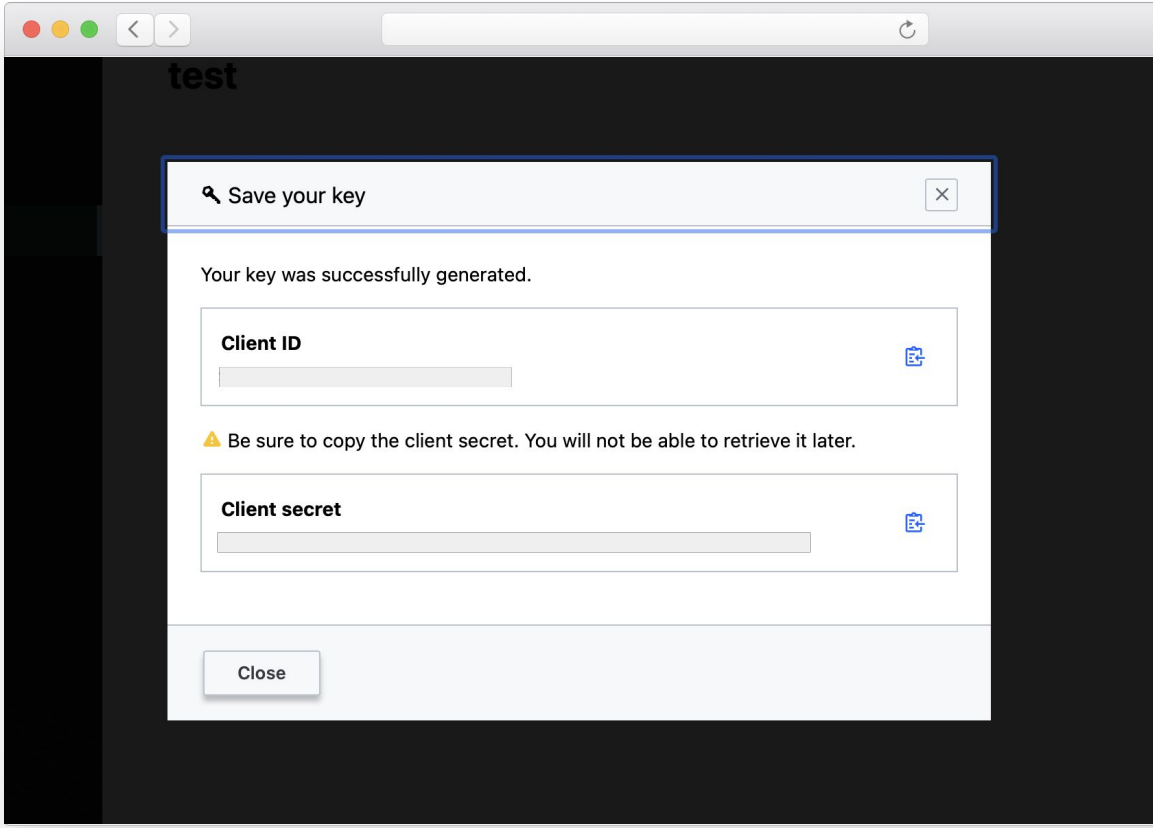




Service Principal Role Selection

Creating Service Principal Key





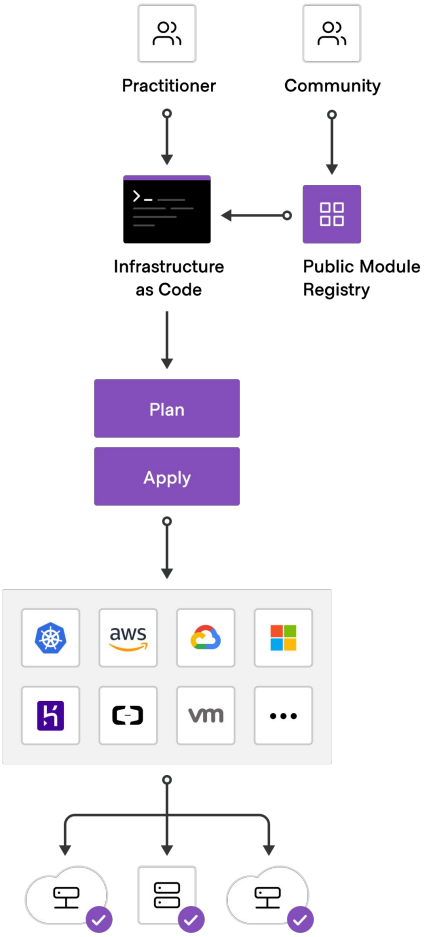
Service Principal Key Secret

Client secret cannot be retrieved later.

Terraform Overview

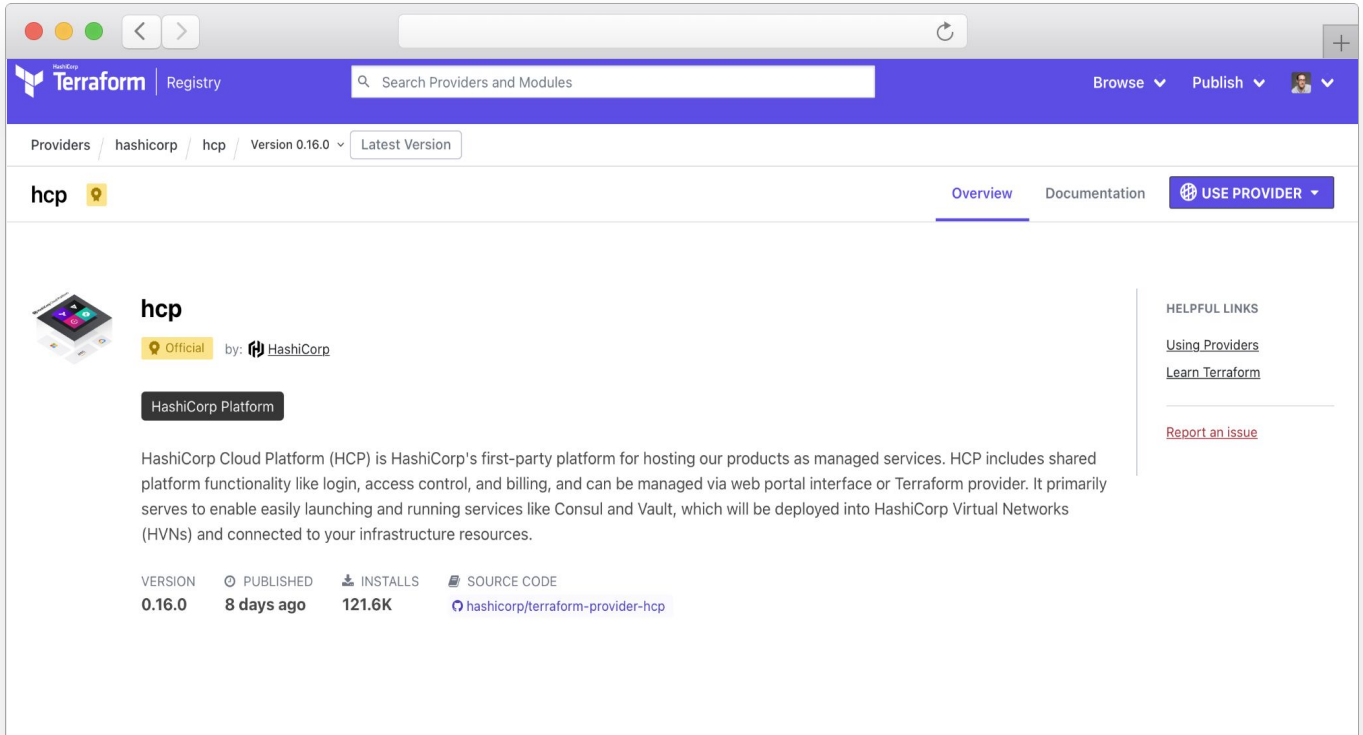
Cloud Infrastructure Automation

Terraform enables cloud infrastructure automation by codifying your infrastructure as code. Infrastructure and services from any provider can be provisioned in a codified, secure, and automated fashion.



HCP Provider

Provision and management of control plane resources in HCP



Module

[Code on GitHub]

main | 1 branch | 0 tags | Go to file | Add file | Code

cbeckhashicorp Update readme — fbfd629 4 hours ago — 37 commits

.gitignore	update gitignore	20 days ago
awsvpc.tf	update	19 days ago
hcp.tf	update	19 days ago
instances.tf	subnet	20 days ago
outputs.tf	update	19 days ago
provider.tf	cloud	20 days ago
readme.md	Update readme	4 hours ago
variables.tf	variable	20 days ago
vault.tf	update	20 days ago

readme.md

COBRA - HCP Vault Onboarding Train

Week 5 - Operationalizing HCP Vault

This repository contains example code for configuration of the HCP control plane and deployment of an HCP Vault cluster.

THe code includes deployment and configuration of the required AWS components including VPC, Transit

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

HCL 100.0%

Automate Vault Configuration

Vault Provider

Provision namespaces, policies, secrets engines, and auth methods

Providers / hashicorp / vault / Version 2.24.0 Latest Version

vault

Overview Documentation USE PROVIDER

vault

Official by: HashiCorp

HashiCorp Platform

Allows Terraform to read from, write to, and configure Hashicorp Vault.

VERSION	PUBLISHED	INSTALLS	SOURCE CODE
2.24.0	8 days ago	34.5M	hashicorp/terraform-provider-vault

How to use this provider

To install this provider, copy and paste this code into your Terraform configuration. Then, run `terraform init`.

Terraform 0.13+

```
terraform {
  required_providers {
    vault = {
      source = "hashicorp/vault"
      version = "2.24.0"
    }
  }
}

provider "vault" {
  # Configuration options
```

Access HCP Vault using Terraform

CODE EDITOR

```
data "hcp_vault_cluster" "dev" {
      cluster_id = var.cluster_id
}

resource "hcp_vault_cluster_admin_token" "token" {
      cluster_id = var.cluster_id
}

provider "vault" {
      address    = data.hcp_vault_cluster.dev.vault.private_endpoint_url
      token      = hcp_vault_cluster_admin_token.token.token
      namespace  = “admin”
}
```

CODE EDITOR

```
resource "vault_namespace" "infosec" {
  path = "infosec"
}

provider vault {
  alias     = "infosec"
  namespace = vault_namespace.infosec.path
}

resource "vault_policy" "example" {
  provider = vault.infosec
  …
}
```

Namespace and Provider Alias

Create Policy

Create auth method for OIDC provider

CODE EDITOR

```
data "vault_policy_document" "dev_user_policy" {
    rule {
        path         = "secret/data/development/*"
        capabilities = ["create", "read", "update",
"delete", "list"]
    }
}

resource "vault_policy" "devusers" {
    name   = "dev-policy"
    policy = "${data.vault_policy_document.hcl}"
}
```

Enable User Authentication Method

Create auth method for OIDC provider

CODE EDITOR

```
resource "vault_jwt_auth_backend" "oidcauth" {
    description         = "Auth0 OIDC"
    path                = "oidc"
    type                = "oidc"
    oidc_discovery_url  = "https://myco.auth0.com/"
    oidc_client_id      = "1234567890"
    oidc_client_secret  = "secret123456"
    bound_issuer        = "https://myco.auth0.com/"
    tune {
        listing_visibility = "unauth"
    }
}
```

CODE EDITOR

```
resource "vault_jwt_auth_backend_role" "example" {
  backend         = vault_jwt_auth_backend.oidc.path
  role_name       = "test-role"
  token_policies  = ["default", "dev", "prod"]

  user_claim            = "https://vault/user"
  role_type             = "oidc"
  allowed_redirect_uris =
["http://localhost:8200/ui/vault/auth/oidc/oidc/callbac
k"]
}
```

Create Auth Role

Role will define the user claim to authenticate a user and which policy assignments they have in Vault.

Enable Secrets Engines

CODE EDITOR

```
resource "vault_mount" "kvv2-infosec" {
  path                          = "infosec"
  type                          = "kv-v2"
}

resource "vault_mount" "pki-dev" {
  path                          = "pki-dev"
  type                          = "pki"
  default_lease_ttl_seconds     = 3600
  max_lease_ttl_seconds         = 86400
}
```

Best Practices

Protect State

Terraform, by default, stores state in the working directory where Terraform CLI is executed. Remote State should be used and encrypted. Access to state should be limited by following practice of least privilege.

Manage as Code

Treat Terraform configuration files as code. Store in a VCS like Github and practice least privilege for access and who can commit changes. Integrate into CI process and ensure code is tested in dev before pushing to production.

Sensitive Values

Do not put any secrets in code. Pass any secrets, such as credentials or Vault token by using environment variables. Sensitive values may appear in state if not handled correctly.

Audit Log

Audit Log

Overview

HCP Vault includes auditing capabilities for all production tier clusters. The logs are written locally and stored in an encrypted S3 bucket.

Audit log retention period varies based on the cluster tier as each tier has different storage capabilities.

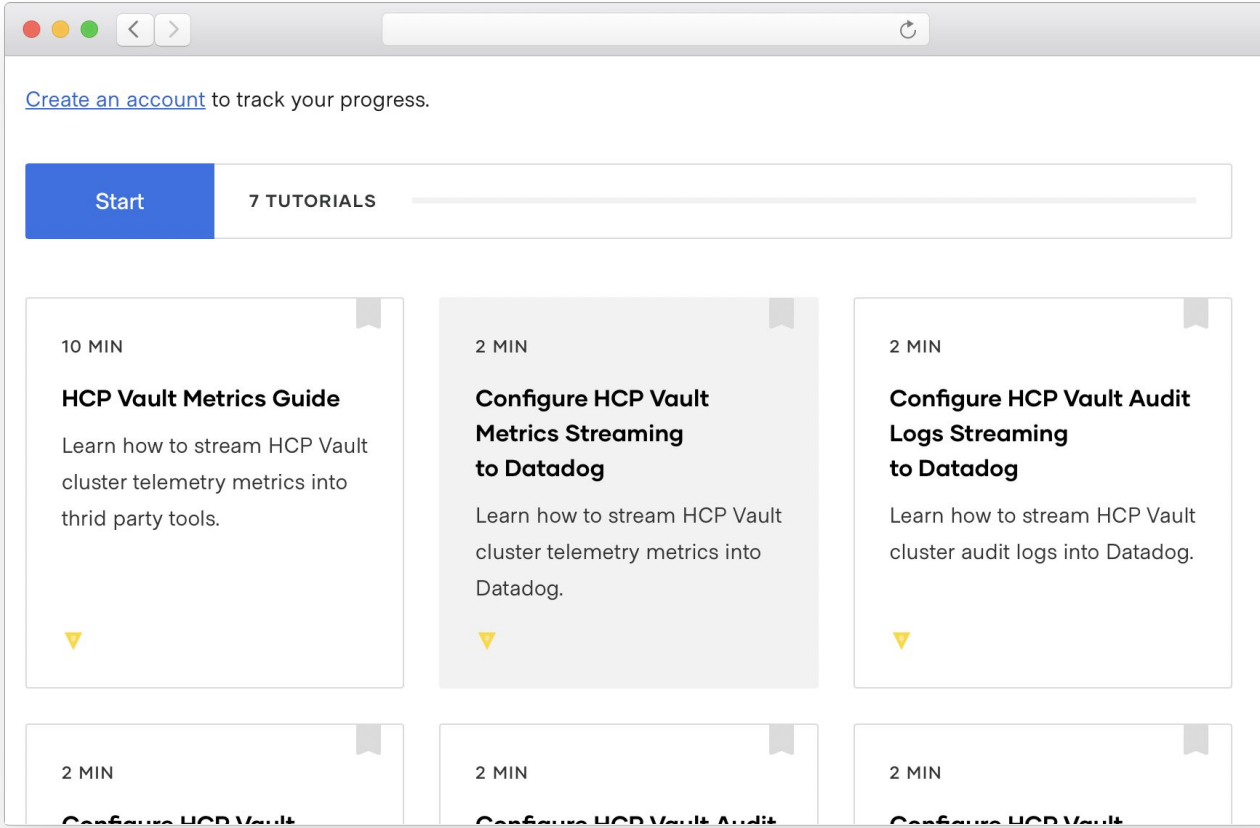
Streaming

Audit logs can be streamed from any production tier cluster to supported third party logging providers.

Currently, we support audit log streaming to Datadog, Grafana Cloud, and Splunk.

Setup Guides

learn.hashicorp.com



Audit Log Access

Audit Logs appear under Vault configuration on the cluster page

Cluster

Configure your HashiCorp Virtual Network, **hvn**, with a peering connection or transit gateway attachment to access your cluster.

View network

This cluster's IP address configuration has been set to public. It is not recommended to use this configuration in production.

Learn more about integration & scaling

Ready to learn more about using Vault and security best practices? Start with our Learn and Docs sites.

- Running Commands with the Vault CLI
- Managing access & policies
- Enabling authentication methods
- Enabling secrets engines
- Integrating with your applications
- Optimizing clients for Vault Cloud
- Vault documentation

Usage

Current clients	0
	0 entity tokens
	0 non-entity tokens

Vault configuration

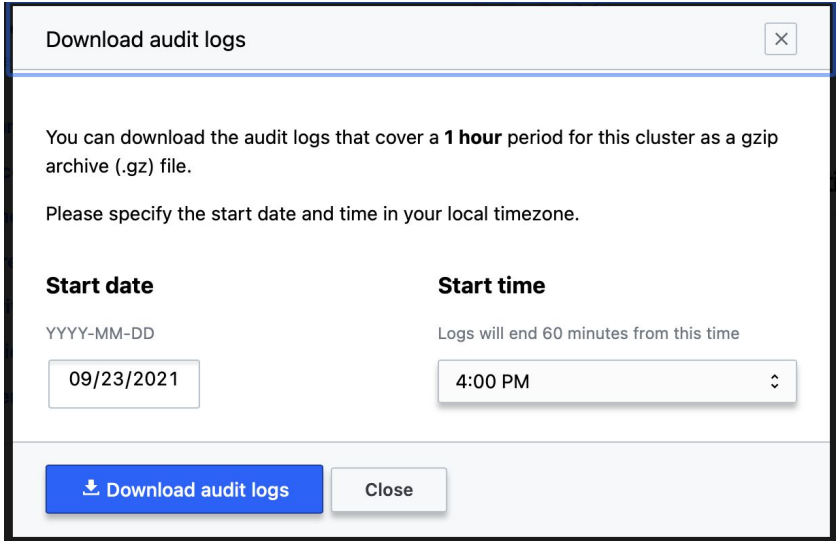
State	Running
Version	v1.8.1
Namespace	admin
Cluster URLs	Private
	Public
Configuration	Default
Generate admin token	+ Generate token
Audit logs	Access audit logs

In case of emergency

Download Audit Logs

Audit logs can be downloaded in 1 hour increments. Once the audit log has been downloaded you can import that into your monitoring solutions for evaluation.

API access is being evaluated as a future capability. Please watch our blogs and release notes to be notified as capabilities are added.



Telemetry

Vault Telemetry

Overview

HCP Vault supports telemetry monitoring to better understand the metrics and usage of your HCP Vault implementation. Metrics can be streamed to Datadog, Grafana Cloud, and Splunk.

Considerations

Metrics streaming is currently supported with the three providers listed above. If you are using an additional provider that is not currently supported, contact us with your provider details so we can investigate support in future releases.

Metrics streaming is not supported with development tier clusters.

Monitoring Patterns

Organizations that have successfully adopted Vault at scale typically classify Vault as a tier 0 application as it is typically a dependency for their most critical applications. Below are the three patterns that should be adopted for monitoring the health of Vault.

1. Time-series telemetry data
2. Log Analytics
3. Active Health Checks

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.

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

Key System Metrics

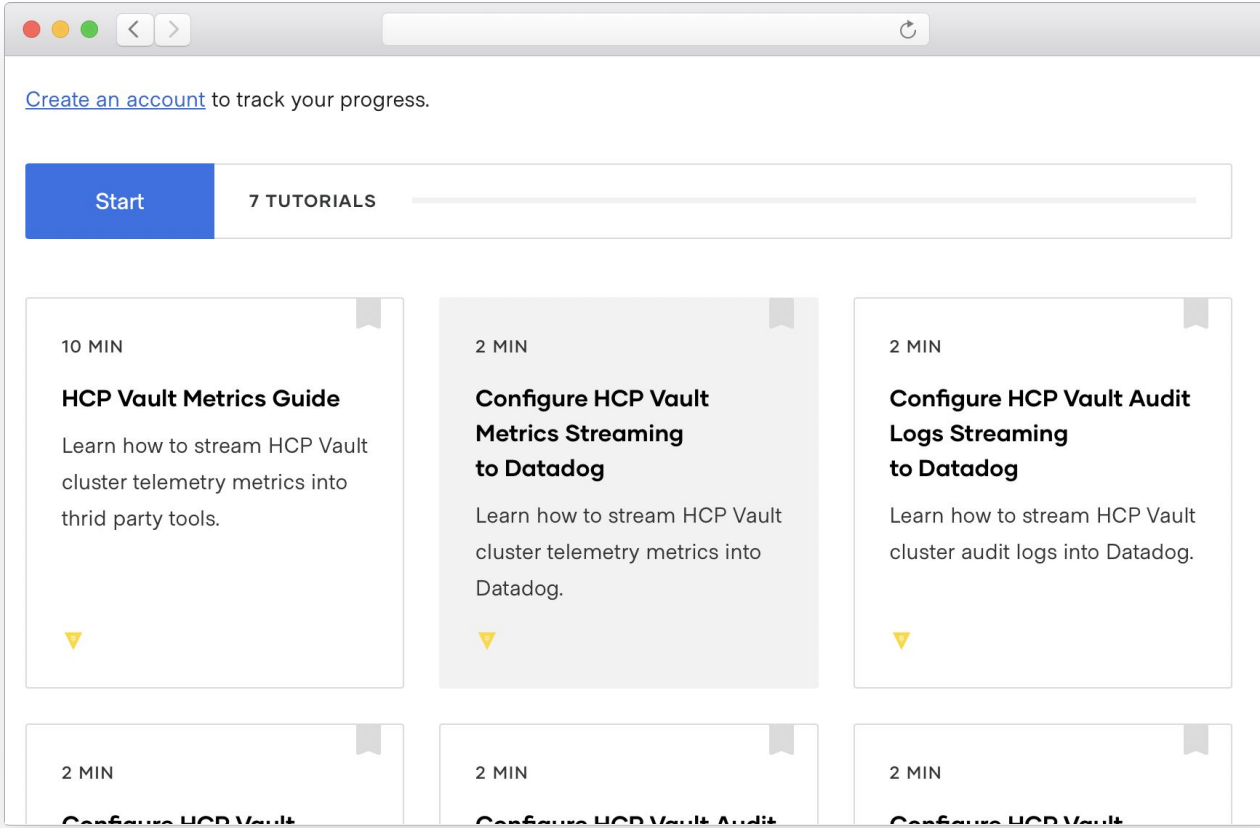
Metric	Description	What to look for?
vault.core.unsealed	Status of Vault seal 1 unsealed. 0 sealed	Unexpected changes to 0
host_cpu_seconds_total	Total CPU time	Heavy
Host_cpu_seconds_total (idle mode)	Time CPU in idle state	Look for heavy CPU usage or unexpected periods of idle, may indicate incorrect sizing.
host_cpu_seconds_total	Total CPU time	
host_memory_total_bytes	Physical RAM available to server	Look for high memory usage or under utilized physical RAM to ensure correct system sizing.
host_memory_available_bytes	Unused physical RAM on the server	

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.)

Setup Guides

learn.hashicorp.com



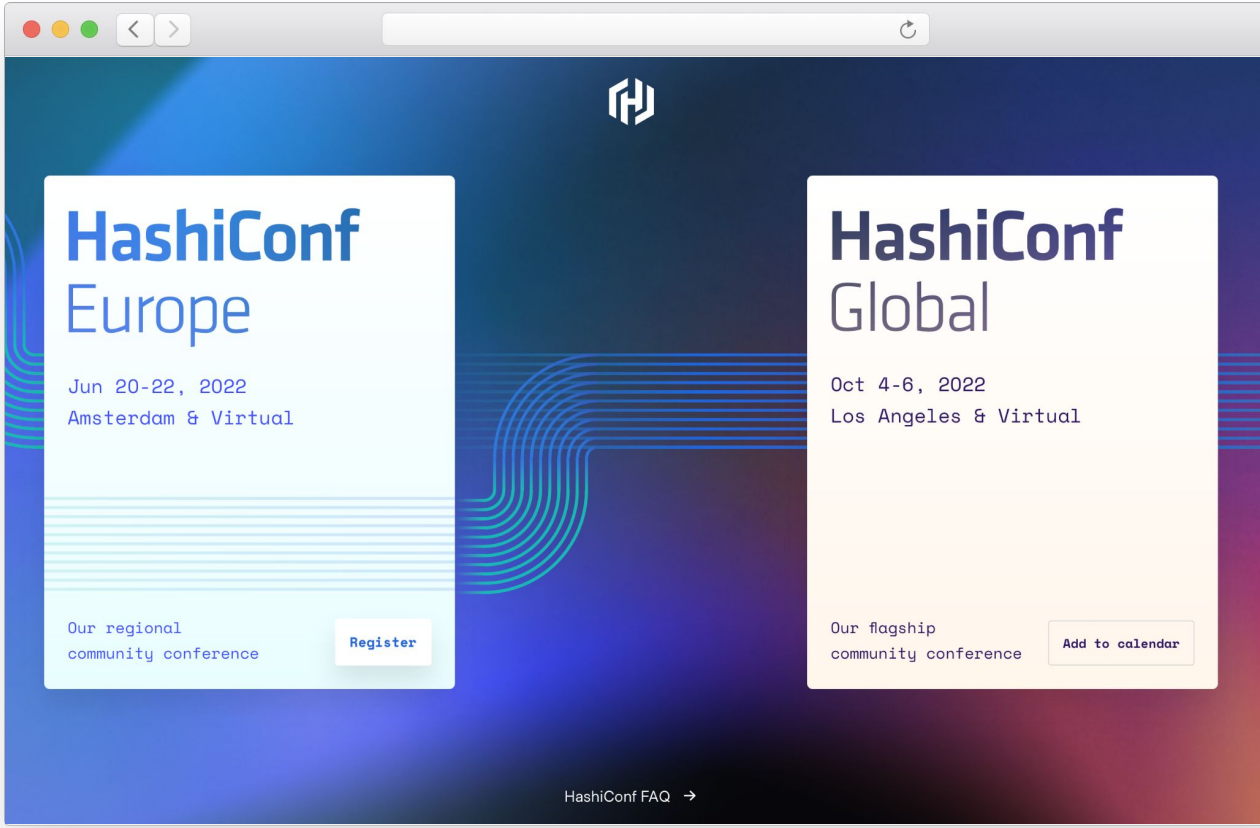
Next Steps

Next Steps

- Upcoming Schedule:
  - Week 6 - Namespaces, Authentication, and Policies Webinar - Learn how to implement identity and access management in HCP Vault
  - Week 7 - Community Office Hours - Bring your questions to this week!
  - Week 8 - Consuming HCP Vault webinar - Learn how to consume secrets from Vault in your apps and services
  - Week 9 - HCP Vault roadmap overview and Q&A with James Bayer, EVP of Secure Product & Engineering
  - Week 10 - HCP Vault train closing session

HashiConf

https://hashiconf.com



Learn

Step-by-step guides to implement features in HCP and HCP Vault

HashiCorp Learn | Browse tutorials | Search / | Sign in

HCP Vault Week 1

- Week 1 - Welcome to HCP Vault
- HCP Vault Introduction
- Create a Vault Cluster on HCP
- Multi-tenancy with Namespaces
- Your First Secret
- Create Vault Policies
- Manage Authentication Methods
- Vault Operation Tasks
- Week 1 Wrap-up

HCP Vault Onboarding Week 1

1 HR 9 MIN | 9 TUTORIALS

Get off to a strong start with HCP Vault in week 1 of your onboarding journey.

15 MIN

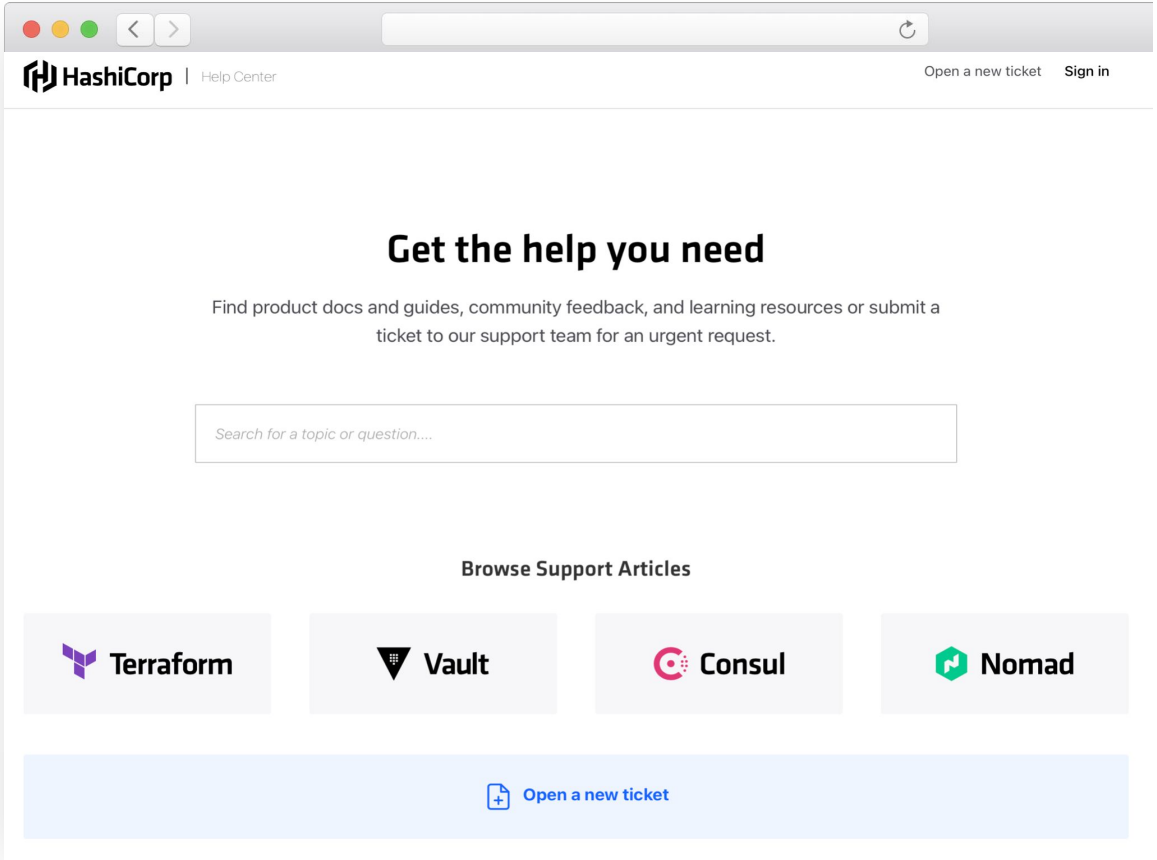
Week 1 - Welcome to HCP Vault

Welcome to week 1 with HashiCorp Cloud Platform (HCP) Vault. Watch a quickstart demo and learn more about the internals of Vault.

5 MIN

HCP Vault Introduction

HashiCorp Cloud Platform (HCP) Vault enables you to deploy a managed Vault service on AWS.



Support

https://support.hashicorp.com

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 new ticket for your issue at Hashicorp Support.

Q & A

Thank You

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