# HashiCorp Vault

Secrets management

## CONCEPTS

The CLI is a thin wrapper over the HTTP API: everything lives at a path, and read/write/list/delete work on any of them.

**Auth method** userpass, token, oidc...
↓ issues
**Token** carries policies
↓ checked by
**Policy** path capabilities
↓ allows
**Secrets engine** kv, transit, database...

The server starts sealed. Unseal it with enough key shares before any request works; dev mode skips this.

Dev mode is in-memory and insecure. Never use it for real secrets.

## CONNECT

**export VAULT_ADDR=**<*url*>
Set the server address

**export VAULT_TOKEN=**<*token*>
Authenticate via environment

**vault status**
Show seal and HA state

**-format=json**
Machine-readable output (also yaml)

**-namespace=**<*ns*>
Run inside a namespace (Enterprise)

**-ca-cert=**<*file*>
Trust a private CA

## DEV SERVER

**vault server -dev**
Run in-memory, unsealed

**vault server -dev -dev-root-token-id=**<*id*>
Dev server with a known root token

**vault server -config=**<*file*>
Run with a config file

## LOGIN AND TOKENS

**vault login**
Log in with a token (prompts)

## LOGIN AND TOKENS (cont.)

**vault login -method=**<*type*> <*key*>=<*value*>
Log in with another auth method

**vault token lookup**
Show the current token

**vault token create -policy=**<*name*>
Create a token with a policy

**vault token create -ttl=**<*dur*> **-use-limit=**<*n*>
Short-lived, limited-use token

**vault token renew**
Extend the current token

**vault token revoke** <*token*>
Revoke a token and its children

## KV SECRETS

**vault kv put -mount=secret** <*path*> <*key*>=<*value*>
Write a secret

**vault kv get -mount=secret** <*path*>
Read a secret

**vault kv get -mount=secret -field=**<*key*> <*path*>
Print one field only

**vault kv list -mount=secret**
List paths under a mount

**vault kv get -mount=secret -version=**<*n*> <*path*>
Read an older version

**vault kv patch -mount=secret** <*path*> <*key*>=<*value*>
Change keys, keep the rest

**vault kv put -mount=secret** <*path*> **@**<*file*>**.json**
Write from a JSON file

**vault kv put -mount=secret** <*path*> <*key*>**=-**
Read a value from stdin

**vault kv delete -mount=secret** <*path*>
Soft-delete the latest version

**vault kv undelete -mount=secret -versions=**<*n*> <*path*>
Restore a deleted version

**vault kv destroy -mount=secret -versions=**<*n*> <*path*>
Permanently remove versions

## KV SECRETS (cont.)

**vault kv metadata get -mount=secret** <*path*>
Show versions and settings

## RAW API PATHS

**vault read** <*path*>
GET any path

**vault read -field=**<*key*> <*path*>
Print one field of a read

**vault write** <*path*> <*key*>=<*value*>
POST/PUT data to a path

**vault write -force** <*path*>
Write with no data

## ENGINES AND AUTH METHODS

**vault secrets list**
Show enabled secrets engines

**vault secrets enable -path=**<*path*> <*type*>
Enable an engine

**vault auth list**
Show enabled auth methods

**vault auth enable** <*type*>
Enable an auth method

## POLICIES

**vault policy list**
List all policies

**vault policy read** <*name*>
Show a policy's rules

**vault policy write** <*name*> <*file*>**.hcl**
Upload a policy from a file

**Read-only policy**

```
path "secret/data/app/*" {
  capabilities = ["read", "list"]
}
```

KV v2 policies need /data/ (and /metadata/) in the path, even though -mount hides it in the CLI.

## OPERATE AND SEAL

**vault operator init**
Initialize a new server

**vault operator unseal**
Enter one unseal key (prompts)