HashiCorp Vault Commands Cheat Sheet

Quick reference for the Vault CLI, API, and day-to-day operations.

By Sam Gabrail · TeKanAid · 2026

1. Environment Setup

```
# Talk to a Vault server
export VAULT_ADDR='http://127.0.0.1:8200'
export VAULT_TOKEN='hvs.xxxxx'

# Skip TLS verification (dev only)
export VAULT_SKIP_VERIFY=true

# Use a CA bundle
export VAULT_CACERT=/etc/vault/ca.pem

# Namespace (Enterprise)
export VAULT_NAMESPACE='admin'
```

2. Server Lifecycle

```
# Dev server (in-memory, single unseal key, root token printed)
vault server -dev

# Production server with config file
vault server -config=/etc/vault.d/vault.hcl

# Status
vault status

# Version
vault version
```

3. Initialize and Unseal

```
# Initialize (Shamir, 5 keys, threshold 3)
vault operator init -key-shares=5 -key-threshold=3

# Initialize with auto-unseal (KMS) and recovery keys
vault operator init -recovery-shares=5 -recovery-threshold=3

# Unseal (run threshold times with different keys)
vault operator unseal <unseal-key>

# Seal
vault operator seal

# Rotate the encryption key
vault operator rotate

# Rekey (change unseal/recovery key shares)
vault operator rekey -init -key-shares=5 -key-threshold=3
```

4. Authentication

```
# Login with token
vault login <token>

# Login with userpass
vault login -method=userpass username=sam

# Login with AppRole
vault write auth/approle/login \
  role_id=<role_id> \
  secret_id=<secret_id>

# Login with OIDC (browser flow)
vault login -method=oidc role=reader

# Logout / revoke current token
vault token revoke -self
```

Token Operations

```
vault token create -policy=default -ttl=1h
vault token create -orphan -policy=admin
vault token lookup
vault token lookup -accessor <accessor>
vault token renew
vault token revoke <token>
vault token revoke -mode=orphan <token>
```

---

5. Auth Methods

```
# List enabled auth methods
vault auth list

# Enable an auth method
vault auth enable userpass
vault auth enable -path=k8s kubernetes
vault auth enable approle
vault auth enable jwt
vault auth enable aws

# Disable an auth method
vault auth disable userpass

# Tune
vault auth tune -default-lease-ttl=1h userpass/
```

AppRole

```
vault write auth/approle/role/my-role \
  token_policies="my-policy" \
  token_ttl=1h token_max_ttl=4h

vault read auth/approle/role/my-role/role-id
vault write -f auth/approle/role/my-role/secret-id
```

Kubernetes

```
vault write auth/kubernetes/config \
  kubernetes_host="https://kubernetes.default.svc" \
  kubernetes_ca_cert=@ca.crt \
  token_reviewer_jwt=@token

vault write auth/kubernetes/role/my-app \
  bound_service_account_names=my-sa \
  bound_service_account_namespaces=default \
  policies=my-app ttl=1h
```

Userpass

```
vault write auth/userpass/users/sam \
  password=changeme \
  policies=admin
```

---

6. Secrets Engines

```
# List enabled engines
vault secrets list -detailed

# Enable
vault secrets enable -path=secret kv-v2
vault secrets enable database
vault secrets enable pki
vault secrets enable transit
vault secrets enable aws

# Disable (destroys data)
vault secrets disable secret/

# Tune (max lease, default lease, audit log redaction)
vault secrets tune -max-lease-ttl=24h pki/
```

KV v2

```
# Write
vault kv put secret/myapp/config username=admin password=pw

# Read
vault kv get secret/myapp/config
vault kv get -field=password secret/myapp/config
vault kv get -format=json secret/myapp/config

# List
vault kv list secret/myapp

# Versions
vault kv get -version=2 secret/myapp/config
vault kv undelete -versions=3 secret/myapp/config
vault kv destroy -versions=3 secret/myapp/config
vault kv metadata get secret/myapp/config

# Patch (merge update)
vault kv patch secret/myapp/config new_key=value

# Delete latest version (soft delete)
vault kv delete secret/myapp/config

# Delete metadata + all versions (hard delete)
vault kv metadata delete secret/myapp/config
```

Transit (encryption-as-a-service)

```
vault write -f transit/keys/my-key
vault write transit/encrypt/my-key plaintext=$(echo "hello" | base64)
vault write transit/decrypt/my-key ciphertext=vault:v1:...
vault write -f transit/keys/my-key/rotate
vault write transit/rewrap/my-key ciphertext=vault:v1:...
```

Database (dynamic creds)

```
vault write database/config/postgres \
  plugin_name=postgresql-database-plugin \
  allowed_roles="readonly" \
  connection_url="postgresql://{{username}}:{{password}}@db:5432/postgres" \
  username="vault" password="vaultpw"

vault write database/roles/readonly \
  db_name=postgres \
  creation_statements="CREATE ROLE \"{{name}}\" WITH LOGIN PASSWORD '{{password}}' VALID UNTIL
'{{expiration}}'; GRANT SELECT ON ALL TABLES IN SCHEMA public TO \"{{name}}\";" \
  default_ttl=1h max_ttl=24h

vault read database/creds/readonly
```

PKI

```
vault secrets enable pki
vault secrets tune -max-lease-ttl=87600h pki/

vault write pki/root/generate/internal \
  common_name="example.com" ttl=87600h

vault write pki/config/urls \
  issuing_certificates="$VAULT_ADDR/v1/pki/ca" \
  crl_distribution_points="$VAULT_ADDR/v1/pki/crl"

vault write pki/roles/example-dot-com \
  allowed_domains="example.com" \
  allow_subdomains=true max_ttl=72h

vault write pki/issue/example-dot-com \
  common_name="web.example.com" ttl=24h
```

AWS

```
vault write aws/config/root \
  access_key=AKIA... secret_key=... region=us-east-1

vault write aws/roles/my-role \
  credential_type=iam_user \
  policy_document=@policy.json

vault read aws/creds/my-role
```

7. Policies

```
# Write a policy from a file
vault policy write admin admin-policy.hcl

# Read a policy
vault policy read admin

# List policies
vault policy list

# Delete a policy
vault policy delete admin

# Format a policy
vault policy fmt my-policy.hcl
```

Policy Syntax

```
# admin-policy.hcl
path "secret/data/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}

path "sys/policies/acl/*" {
  capabilities = ["read", "list"]
}

path "auth/token/create" {
  capabilities = ["create", "update"]
}
```

Capabilities: `create`, `read`, `update`, `delete`, `list`, `sudo`, `deny`, `patch`.

8. Leases

```
vault list sys/leases/lookup/database/creds/readonly
vault lease lookup <lease_id>
vault lease renew <lease_id>
vault lease revoke <lease_id>
vault lease revoke -prefix database/creds/readonly
vault lease revoke -force -prefix pki/
```

9. Audit Devices

```
vault audit list
vault audit enable file file_path=/var/log/vault_audit.log
vault audit enable syslog tag="vault" facility="AUTH"
vault audit disable file/
```

10. Raft Storage Operations

```
# Cluster
vault operator raft list-peers
vault operator raft join https://active-node:8200
vault operator raft remove-peer <node-id>

# Snapshots
vault operator raft snapshot save backup.snap
vault operator raft snapshot restore backup.snap
vault operator raft snapshot restore -force backup.snap

# Autopilot
vault operator raft autopilot get-config
vault operator raft autopilot state
```

11. Namespaces (Enterprise)

```
vault namespace list
vault namespace create finance
vault namespace lookup finance
vault namespace delete finance

# Operate inside a namespace
VAULT_NAMESPACE=finance vault kv put secret/key value=hi
```

12. Useful API Calls (curl)

```
# Read a secret
curl -s -H "X-Vault-Token: $VAULT_TOKEN" \
  $VAULT_ADDR/v1/secret/data/myapp/config | jq

# Health
curl -s $VAULT_ADDR/v1/sys/health | jq

# Seal status
curl -s $VAULT_ADDR/v1/sys/seal-status | jq

# Login with AppRole
curl -s -X POST -d '{"role_id":"...","secret_id":"..."}' \
  $VAULT_ADDR/v1/auth/approle/login | jq

# OpenAPI spec (pro tip)
curl -s -H "X-Vault-Token: $VAULT_TOKEN" \
  $VAULT_ADDR/v1/sys/internal/specs/openapi > vault-openapi.json
```

13. Agent and Templating

```
# Run Vault Agent with a config file
vault agent -config=/etc/vault/agent.hcl

# Render a template once
vault agent -config=agent.hcl -exit-after-auth
```

`agent.hcl` essentials: `auto_auth`, `sink`, `template`, `cache`, `listener`.

14. Debugging

```
# Bundle metrics, pprof, logs
vault debug -duration=2m -output=/tmp/vault-debug.tar.gz

# Live monitor
vault monitor -log-level=debug

# Metrics
curl -s -H "X-Vault-Token: $VAULT_TOKEN" \
  $VAULT_ADDR/v1/sys/metrics?format=prometheus
```

15. Common Recipes

```
# Rotate a static DB password
vault write -force database/rotate-root/postgres

# Wrap a secret_id (response wrapping)
vault write -wrap-ttl=120s -f auth/approle/role/my-role/secret-id

# Unwrap
VAULT_TOKEN=<wrapping_token> vault unwrap

# Lookup wrapping token
vault write sys/wrapping/lookup token=<wrapping_token>

# Identity entity / alias
vault write identity/entity name="sam" policies="admin"
vault write identity/entity-alias \
  name="sam" \
  canonical_id=<entity_id> \
  mount_accessor=<userpass_accessor>
```

Capability Cheat Card

Need	Command
See what a token can do	`vault token capabilities <token> <path>`
Find the active node	`vault status` (look for `HA Mode`)
Force-revoke all leases under a path	`vault lease revoke -prefix -force <path>`
Test policy without applying	`vault policy fmt my.hcl`
Find a mount's accessor	`vault auth list` / `vault secrets list` (`-detailed`)
Replay encrypted data after key rotation	`vault write transit/rewrap/<key> ciphertext=...`

Want the full Vault learning path? Take the Vault Associate 003 course at tekanaid.com/course/vault-associate-003.