PRACTICAL: HIGH-PERFORMANCE COMPUTING SYSTEM ADMINISTRATION

Encryption Tools

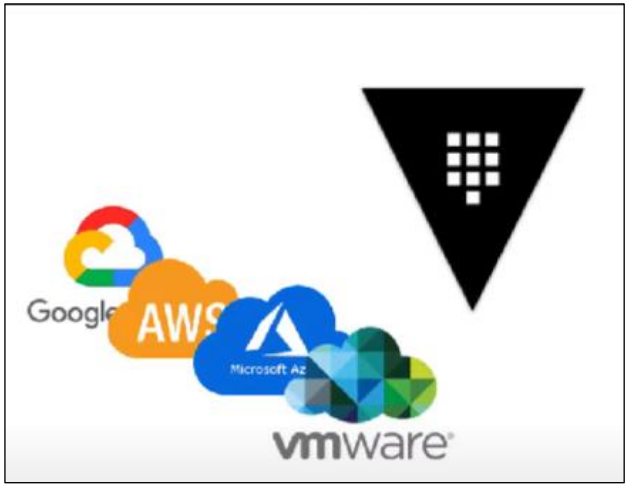
Sonal Lakhotia

Table Of Contents

Table Of Contents

What is HashiCorp Vault?

- Identity-based secrets and encryption management system
- Secret - encryption keys, passwords, and certificates
- It is a cloud agnostic
- API driven
- Used to generate dynamic credentials with short TTL



Why Vault?

- Most enterprises have credentials sprawled across their organizations
- Passwords, API keys, and credentials stored in
  - Plain text
  - App source code
  - Config files, and other locations
- Difficult to determine who has access and authorization to what
- Potential for malicious attacks - internal and external attackers.

How does Vault work?

- **Authenticate** – determines the client
- **Validation** - client validation against trusted third-party such as Github, LDAP, AppRole
- **Authorize** - A client is matched against the Vault security policy.
- **Access** - Vault grants access to secrets, keys, and encryption capabilities

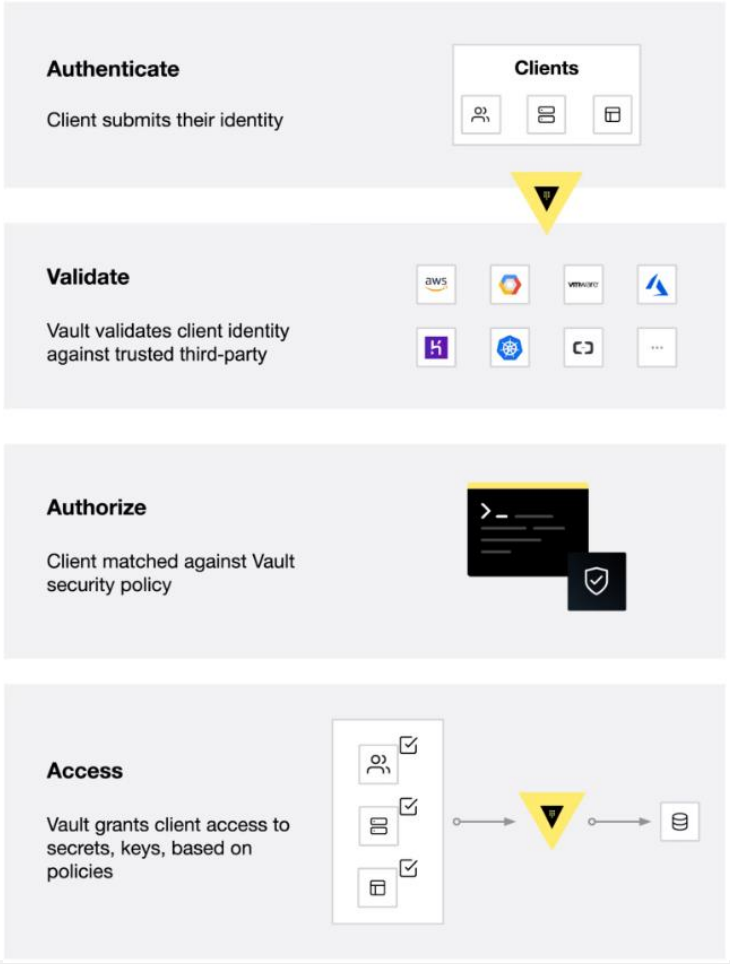


Table Of Contents

Vault Configuration

- Vault can be configured with one or more HashiCorp Configuration Language (HCL) files
- Using Command line flags
- Environment variables
- Using HCL files to configure is most common

Vault Configuration

- Global configuration options: ui and disable_mlock
- Listener configuration: address and tls_disable
- Storage configuration: path

```
disable_mlock = true
ui            = true

listener "tcp" {
  address     = "127.0.0.1:8200"
  tls_disable = "true"
}

storage "file" {
  path = "/tmp/vault-data"
}
```

Table Of Contents

Security Considerations

- Not using root account - defense against various privilege-escalation attacks
- Allow minimal write privileges – securing overwriting Vault configuration files.
- Always use TLS to provide secure communication between clients and the Vault server
- Firewall traffic - to restrict incoming and outgoing traffic to Vault

Security Considerations

- Enable audit logging
- Restrict Storage Access
- Use Short TTLs - credentials issued from Vault e.g. tokens should be short-lived
- Use Correct Filesystem Permissions - ensure appropriate permissions are applied to files prior to starting Vault
- Upgrade Frequently

Table Of Contents

Use Case Requirements

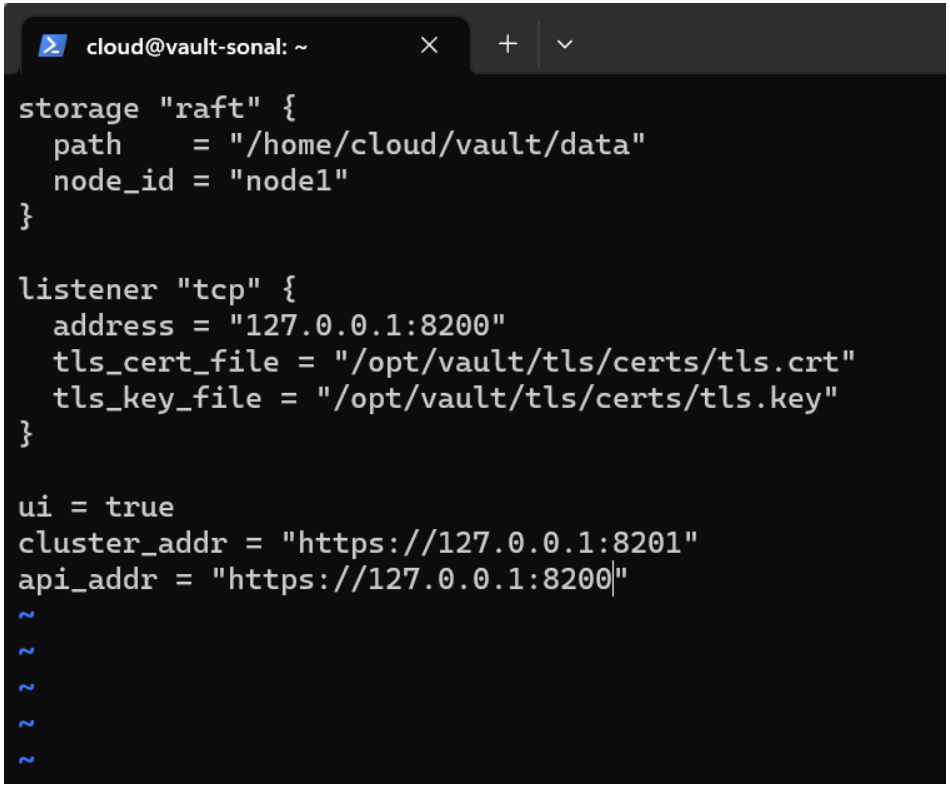
Tokens are the core method for authentication within Vault

- We use service tokens for most cases as they are renewable and revocable within their TTL
- Single-use tokens: We can set a use limit while creating tokens
- Root users can generate periodic tokens
- Limited TTL: Tokens have a TTL of 32 days but this can be changed in Vault‘s config file
- Storage in RAM: Storage of secrets in /tmpfs

Table Of Contents

Vault Configuration

- Installing and Configuring Vault with TLS – TLS is usually disabled in dev mode



Vault Configuration

- api_addr - address to advertise to route client requests
- cluster_addr - Indicates the address and port to be used for communication between the Vault nodes in a cluster
- ui - gives access to web UI
- Vault should always use TLS for security - It requires a certificate file and key file on each Vault host

Vault Configuration

```
root@vault-sonal:~# export VAULT_ADDR='https://127.0.0.1:8200'
root@vault-sonal:~# vault status
Key                Value
---                -----
Seal Type          shamir
Initialized        true
Sealed             true
Total Shares       5
Threshold          3
Unseal Progress    0/3
Unseal Nonce       n/a
Version            1.12.2
Build Date         2022-11-23T12:53:46Z
Storage Type       raft
HA Enabled         true
root@vault-sonal:~#
```

```
root@vault-sonal:~# vault login
Token (will be hidden):
Success! You are now authenticated. The token information displayed below
is already stored in the token helper. You do NOT need to run "vault login"
again. Future Vault requests will automatically use this token.

Key                  Value
---                  -----
token                hvs.AQIXHeuuDv2kTlwf1EDQJBI5
token_accessor       wuq5kMV1N1nAum855e0V763z
token_duration       ∞
token_renewable      false
token_policies       ["root"]
identity_policies    []
policies             ["root"]
root@vault-sonal:~#
```

Further Configurations

- I would proceed to create single-use tokens for users
- Tokens with limited TTL