Enterprise HashiCorp Vault Automation

Podman Deployment & Intelligent File Sync Solutions

Powered by solutions.com.sa

Infrastructure Automation Division

August 26, 2025

Abstract

This document summarizes a single-node HashiCorp Vault deployment automated with Podman plus a helper script (`sync_on_change_clean.sh`) that watches a local file, securely syncing changes to a remote host. The watcher now supports layered configuration precedence and per-variable source reporting.

Brand Alignment

This automation solution reflects **solutions.com.sa**'s commitment to enterprise-grade infrastructure automation. Our approach emphasizes:

- **Security First**: TLS-enabled deployments with certificate management
- **Operational Excellence**: Automated deployment with comprehensive health monitoring
- **Intelligent Automation**: Smart file synchronization with configurable precedence
- **Enterprise Ready**: Production-focused with secure defaults and validation

For additional infrastructure solutions and consulting services, visit **www.solutions.com.sa**.

1 Overview

- Installs / verifies Podman (and optionally Vault CLI).
- Prepares directory structure under `$HOME/vault-server`.
- Generates (or reuses) a self-signed TLS certificate.
- Optionally trusts the certificate system-wide.
- Applies secure filesystem permissions (with permissive fallback).
- Writes `config.hcl` using file storage backend + TLS listener.
- Optionally validates the configuration.
- Manages firewall openings (unless disabled).
- Starts or replaces the Vault container and polls health.

2 Quick Start (Non-Interactive)

```
TRUST_CERT=1 INSTALL_CLI=1
    ./prod/install_vault_container_prod-moduler-version-clean-working.sh
```

If the certificate is not trusted:

```
export VAULT_ADDR=https://127.0.0.1:8200
export VAULT_SKIP_VERIFY=1 # Dev only; prefer --cacert
```

3 Initialize & Unseal

```
vault operator init
vault operator unseal
vault status
```

4 Core Environment Variables

Variable	Purpose	Default
VAULT_VERSION	Vault image tag	`latest`
VAULT_PORT	API listen port	8200
VAULT_CLUSTER_PORT	Cluster port	8201
VAULT_API_ADDR	Advertised API	`https://127.0.0.1:8200`
INSTALL_CLI	Install Vault CLI (1=yes)	0

Variable	Purpose	Default
TRUST_CERT / TRUST_VAULT_CERT	Trust self-signed cert	0
PERMISSIVE_STORAGE	Force wide-open perms	0
FIREWALL_DISABLE	Skip firewall changes	0
CHECK_VAULT_CONFIG	Run `-check-config`	1

5 Directory Layout

```
$HOME/vault-server/
  data/
    certs/
      public.crt
      private.key
    storage/
  config/
    config.hcl
```

6 Container Mounts

```
/data   -> data + certs
/config -> configuration
```

7 Health & Status

```
curl --cacert $HOME/vault-server/data/certs/public.crt https://127.0.0.1:8200/v1/sys/health
# Dev only:
curl -k https://127.0.0.1:8200/v1/sys/health
podman logs vault | head
podman exec vault vault status -tls-skip-verify
```

Return codes:

- 501 = uninitialized
- 503 = sealed

8 Updating Vault

```
VAULT_VERSION=1.20.2 ./prod/install_vault_container_prod-moduler-version-clean-working.sh
```

Omit `VAULT_VERSION` to keep `latest`.

9 Clean Removal

Script:

```
./clean_vault_script.sh # interactive
./clean_vault_script.sh -f # forced
CLEAN_IMAGE=1 CLEAN_CLI=1 ./clean_vault_script.sh -f
```

Manual:

```
podman rm -f vault
rm -rf $HOME/vault-server
```

10 Security Notes

- Replace self-signed cert in production.
- File storage backend is single-node; use Raft/external for HA.
- Avoid permissive permissions outside dev.
- Protect unseal keys and root token.
- Pin explicit Vault versions.

11 Troubleshooting

Symptom	Action
Startup timeout	`podman logs vault` (check TLS, perms, mlock)
Certificate errors	Use `TRUST_CERT=1` or `--cacert`
Port in use	`ss -tulnp | grep ':8200'` then free or override
501 health	Initialize (`vault operator init`)
503 health	Unseal (`vault operator unseal`)

12 Watcher: sync_on_change_clean.sh

Purpose: monitor a single local file, detect content changes (SHA-256), scp it to a remote host, and set execute permission remotely.

Usage

```
./sync_on_change_clean.sh <file> [-c <config-file>] [-h]
```

Examples

```
./sync_on_change_clean.sh prod/install_vault_container_prod-moduler-version-clean-working.sh
./sync_on_change_clean.sh
    prod/install_vault_container_prod-moduler-version-clean-working.sh -c
    ./sync_on_change.conf
```

Configuration Precedence (first found wins)

1. `-c <config-file>` (explicit; must exist; no fallback)
2. `<script_dir>/sync_on_change.conf`
3. `$PWD/.sync_on_change.conf`
4. `<script_dir>/.sync_on_change.conf`
5. `<watched_file_dir>/.sync_on_change.conf`

Overridable Variables

- `remote_user`
- `remote_host`
- `remote_path`
- `interval`
- `max_failures`

Startup Reporting

The script prints each variable along with its source (default vs the config file path).

Sample Configuration File

```
# sync_on_change.conf
remote_user="username"
remote_host="172.00.00.00"
remote_path="~/vault-scripts/"
interval=2
max_failures=15
```

13 Minimal Workflow

```
Run install script -> vault operator init -> vault operator unseal
Set VAULT_ADDR -> use Vault
```

14 Production Reminders

Harden TLS, use HA backend, secure unseal keys, pin versions, monitor logs.