Dynamic Secrets Engines

Ned Bellavance

HashiCorp Certified Instructor

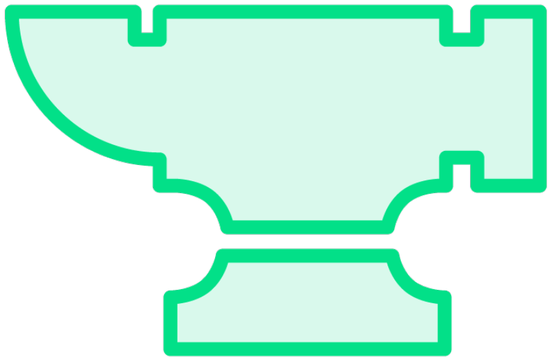
@nedinthecloud | nedinthecloud.com

Secret Types

Static secrets

Dynamic secrets

Dynamic Secrets Engines



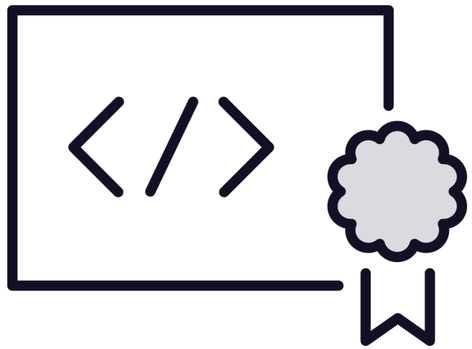
Create secrets on-demand

Manage secret lifecycle

Lease controls secret validity

Secret revoked when lease or token expires

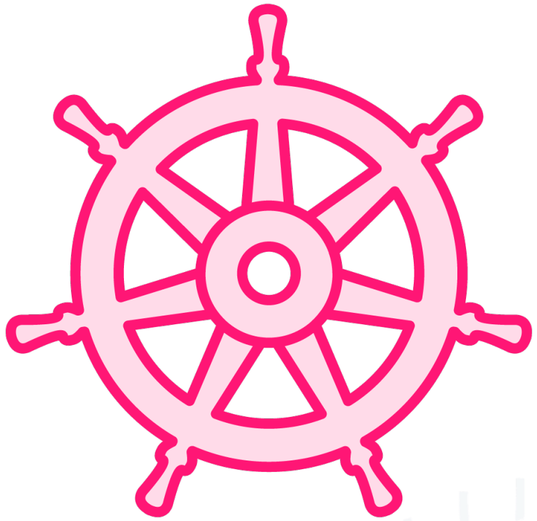
Dynamic Secrets Engine Examples



PKI

Database

Cloud



Kubernetes

Dynamic Secrets Engine Setup

1. Gather requirements and configuration info
2. Enable secrets engine at desired path
3. Configure secrets engine with gathered info
4. Create roles for clients
5. Create policies to grant access to roles

Roster Scenario

Requirements

- Postgres database access
- Receive credentials on-demand
- Native application integration
- Admin and reader roles

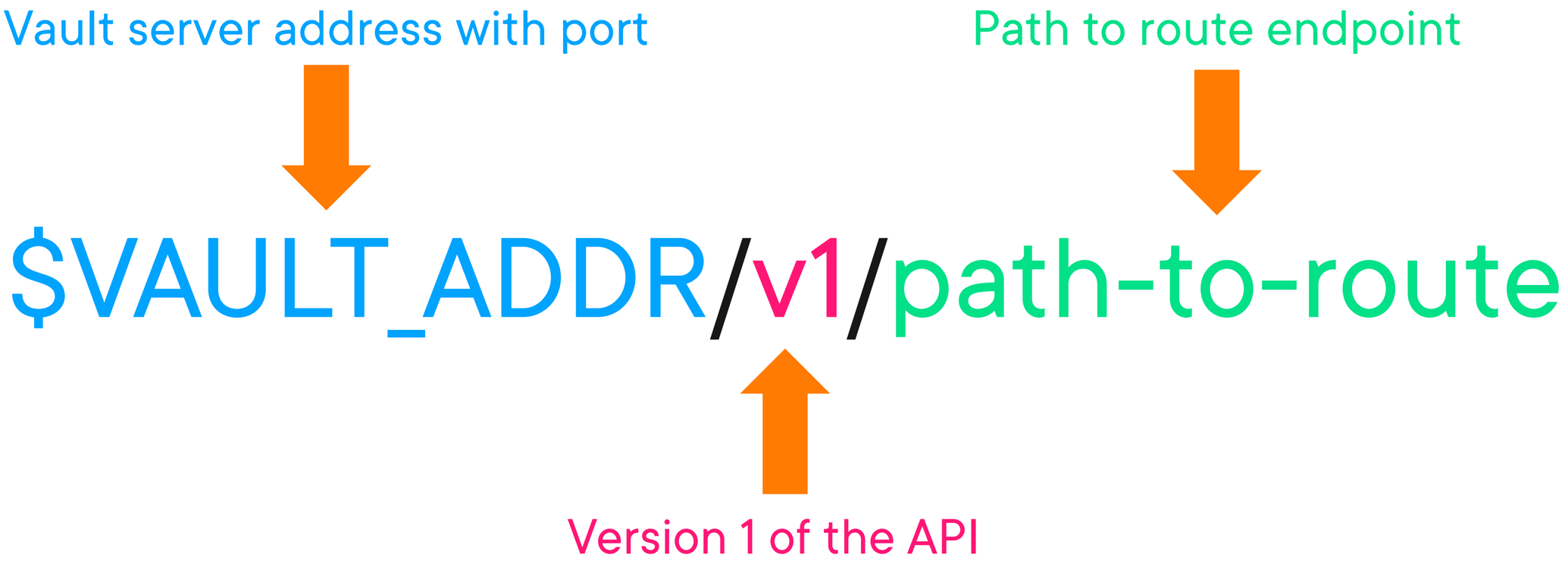
Solution

- Database secrets engine with Postgres plugin
- Demonstrate API

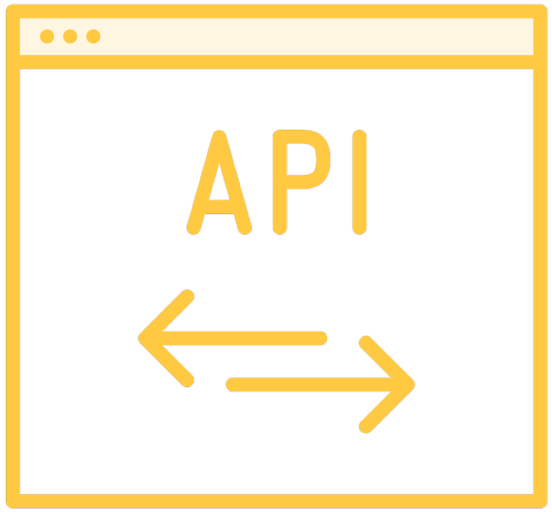
Using the Vault API

Everything in Vault is treated as a path

Vault API Path Construction



Vault API for Secrets Engines



Mount and tune

- /sys/mounts

Configure and interact

- /engine_path

HTTP verbs

- GET, POST, PUT, PATCH, DELETE, LIST

HTTP headers

- X-Vault-Token
- Bearer: <token>

Helpful Resources

API Explorer

-output-curl-string

API Docs