

The Secret Problem

David King
Sr. HPC Engineer



**National Center for
Supercomputing Applications**

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

The Secret Problem

- There are secrets (certificates, krb5keytab, passwords) in configuration management that can be difficult to manage and store securely
- Finding a way to securely store and access secrets without exposing secrets directly
- Storing Secrets in Private Git Repo
 - Always a Bad Idea
 - Security by Obfuscation
 - There is no way to limit secrets to various groups without removing access to the repositories to less privileged groups
- No good way to store secrets, Just “OK” ways.



Options for Solutions

- Storing Secrets in Git
 - Still requires specific key management or specific to a platform
 - Encrypt Secrets within Git
 - Git-Secret, gcrypt or Eyaml hiera for Puppet
- Storing Secrets in Configuration Files
 - Stored separate from Git repo
 - Deployed Separately from Puppet
- A Vault Solution
 - Hashicorp Vault
 - Azure or Google Cloud
 - Lastpass



What We Chose and Why

- Vault by Hashicorp with Consul as encrypted storage.
- On Premises
- Authorization for compute is done using Puppet Server Certificate Authority and through OIDC Single Sign On
 - Many other authentication methods
- Provides flexibility to create policies that control administrative access
- A single Vault instance provides secret storage for multiple Puppet Control Repos at NCSA
- Puppet has integration with numerous Puppet Forge projects that integrate with Vault




Other Vault Features

- API driven
- Auditable client interaction
- Lots of different plugins for different applications
 - MultiFactor
- Lots of Authentication methods
- Dynamic Secrets
- Web GUI



Web GUI



Secrets

Access

Tools

Status

< delta

< common

common

JSON

Delete

Copy

Version 10

Create new version

Key	Value
createhost.keytab	<div><div></div><div></div><div></div></div>
duoapihost	<div><div></div><div></div><div></div></div>
duoikey	<div><div></div><div></div><div></div></div>
duoskey	<div><div></div><div></div><div></div></div>
root	<div><div></div><div></div><div></div></div>
root-private	<div><div></div><div></div><div></div></div>
root-public	<div><div></div><div></div><div></div></div>

© 2021 HashiCorp

[Vault 1.8.4](#)

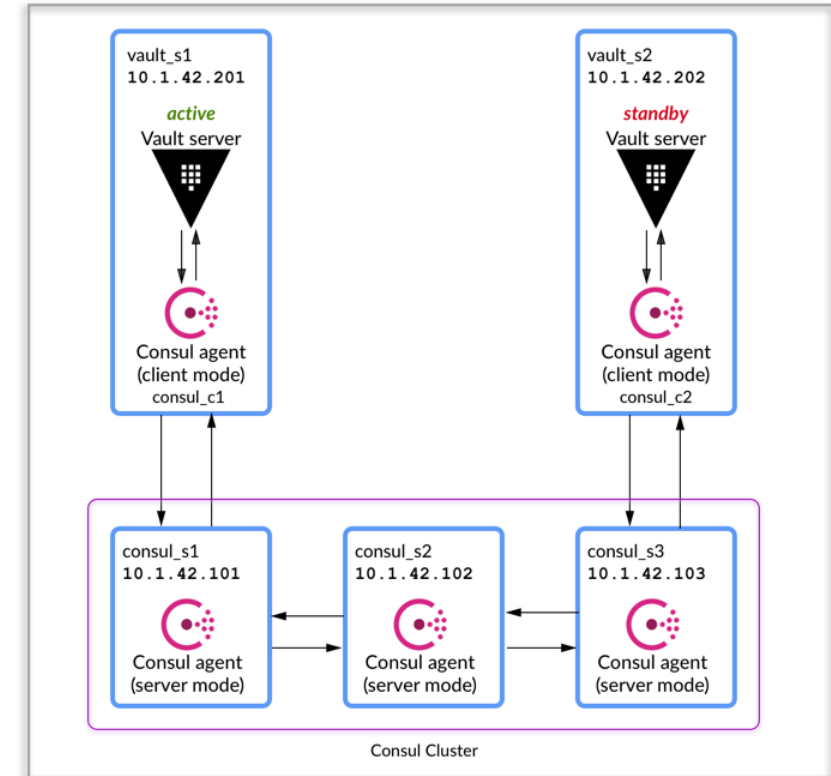
[Upgrade to Vault Enterprise](#)

[Documentation](#)



Our Infrastructure

- 3 Consul Servers running on different VM infrastructures
 - Encrypted Key Pair Storage
 - High availability
 - Encrypted Storage Key Pair
 - Hashicorp Supported
- 2 Vault Servers
 - Round Robin
 - Active/Passive configuration
 - High Availability possible with load balancer
 - Future Task



Integration with Puppet

- Puppet module vault_secrets from Puppet Forge southalc
 - https://forge.puppet.com/modules/southalc/vault_secrets
- Created a common profile for secret lookups
 - https://github.com/ncsa/puppet-profile_secrets
- Lookups with in Hieradata or from within a module
- Lookups on the compute node with a Deferred Function



Outcome

- Secrets are easier to manage
 - Changing of passwords happens without
- Security is improved with no storing of secrets in GIT
- Administration is more granular
- No longer rely on using Lastpass as the definitive source of data



Future Plans

- High Availability for Vault Frontends
 - Using Consul or other Load Balancer
- Integration with Lastpass
 - Backup passwords to Lastpass
- Possibly providing Vault as a service to other NCSA groups
- Leverage Consul Infrastructure for other uses



References

- <https://withblue.ink/2021/05/07/storing-secrets-and-passwords-in-git-is-bad.html>
- <https://git-secret.io/>
- <https://github.com/spwhitton/git-remote-gcrypt>
- https://git-annex.branchable.com/tips/fully_encrypted_git_repositories_with_gcrypt/
- <https://dzone.com/articles/storing-encrypted-credentials-in-git>
- <https://www.vaultproject.io/>
- <https://puppet.com/blog/my-journey-securing-sensitive-data-puppet-code/>
- <https://learn.hashicorp.com/tutorials/vault/ha-with-consul>

