HashiCorp Vault Plugin

Plugin Information
View HashiCorp Vault on the plugin site for more information.

This plugin adds a build wrapper to set environment variables from a HashiCorp Vault secret. Secrets are generally masked in the build log, so you can't accidentally print them.

Migration Guide

Upgrade from 1.x to 2.0

The *BuildWrapper* did not change, so no changes to your Jenkinsfile should be necessary. However, you need to reconfigure Vault in your Jenkins instance based on the instructions above. There is no way to smoothly upgrade this, because this is a major rewrite and handling of configuration completely changed.

Vault Authentication Backends

This plugin allows authenticating against Vault using the AppRole authentication backend. Hashicorp recommends using AppRole for Servers / automated workflows (like Jenkins) and using Tokens (default mechanism, Github Token, ...) for every developer's machine. Furthermore, this plugin allows using a Github personal access token, or a Vault Token - either configured directly in Jenkins or read from an arbitrary file on the Jenkins Machine.

How does AppRole work?

In short: you register an approle auth backend using a self-chosen name (e.g. Jenkins). This approle is identified by a `role-id` and secured with a `secret_id`. If you have both of those values you can ask Vault for a token that can be used to access vault.
When registering the approle backend you can set a couple of different parameters:

- How long should the `secret_id` live (can be indefinite)
- How often can one use a token that is obtained via this backend
- Which IP addresses can obtain a token using `role-id` and `secret-id`?
- many more

This is just a short introduction, please refer to HashiCorp itself to get detailed information.

What about other backends?

Hashicorp explicitly recommends the AppRole Backend for machine-to-machine authentication. Token based auth is mainly supported for backwards compatibility.
Another backend that might make much sense is the AWS EC2 backend, but we do not support it yet. Feel free to contribute!

Implementing additional authentication backends is actually quite easy:

Simply provide a class implementing *VaultTokenCredential* that contains a *Descriptor* extending *BaseStandardCredentialsDescriptor*.
The *Descriptor* needs to be annotated with *@Extension*. Your credential needs to know how to authenticate with Vault and provide an authenticated Vault session.

See VaultAppRoleCredential.java for an example.

Plugin Usage

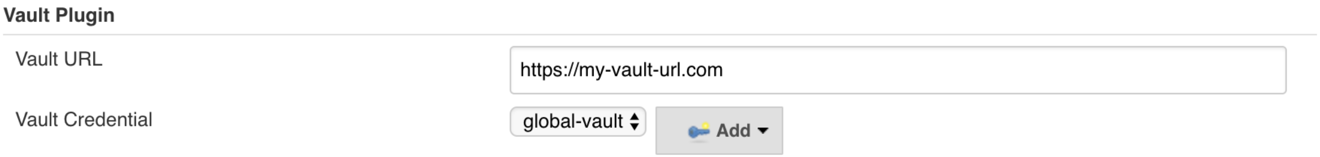
Configuration

You can configure the plugin on three different levels:

- Global: in your global config
- Folder-Level: on the folder your job is running in
- Job-Level: either on your freestyle project job or directly in the Jenkinsfile

The lower the level the higher its priority, meaning: if you configure a URL in your global settings, but override it in your particular job, this URL will be used for communicating with Vault.

In your configuration (may it be global, folder or job) you see the following screen:



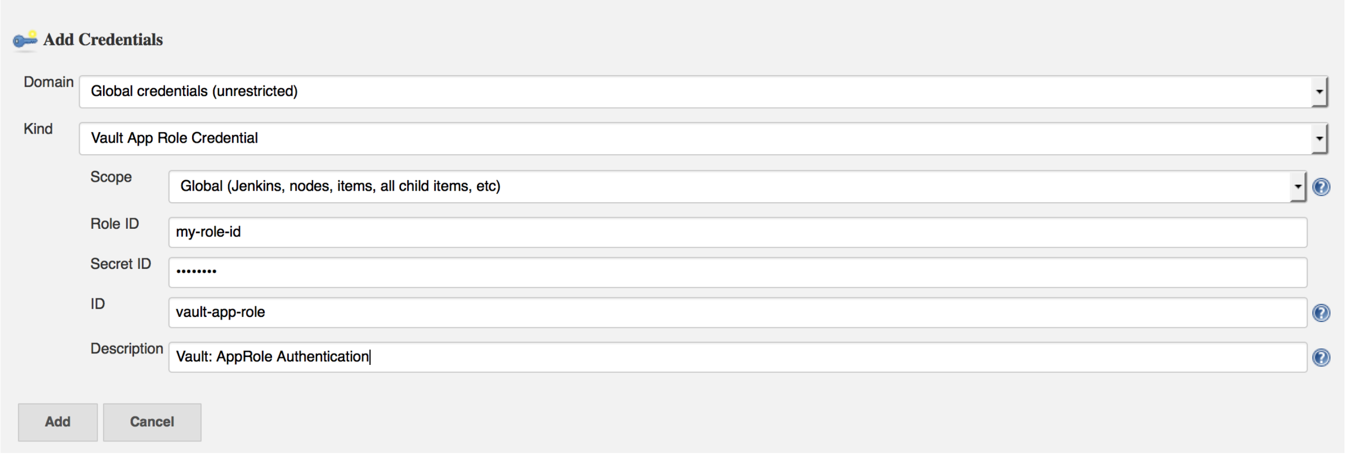
Vault Plugin

Vault URL: https://my-vault-url.com

Vault Credential: global-vault

The credential you provide determines what authentication backend will be used.
Currently, there are three different Credential Types you can use:

Vault App Role Credential



Add Credentials

Domain: Global credentials (unrestricted)

Kind: Vault App Role Credential

Scope: Global (Jenkins, nodes, items, all child items, etc)

Role ID: my-role-id

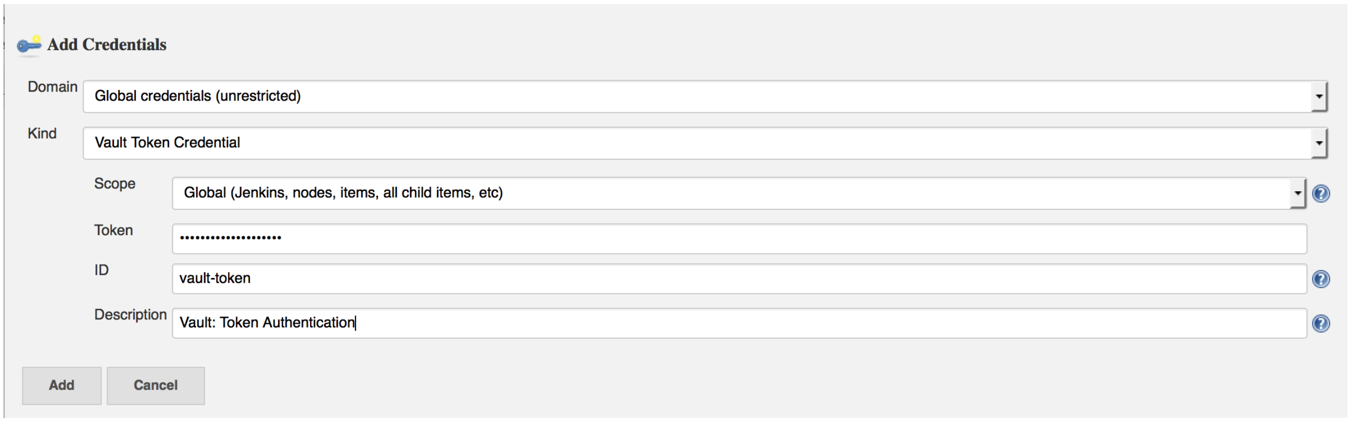
Secret ID: ••••••••

ID: vault-app-role

Description: Vault: AppRole Authentication

You enter your *role-id* and *secret-id* there. The description helps to find your credential later, the id is not mandatory (a UUID is generated by default), but it helps to set it if you want to use your credential inside the Jenkinsfile.

Vault Token Credential



Add Credentials

Domain: Global credentials (unrestricted)

Kind: Vault Token Credential

Scope: Global (Jenkins, nodes, items, all child items, etc)

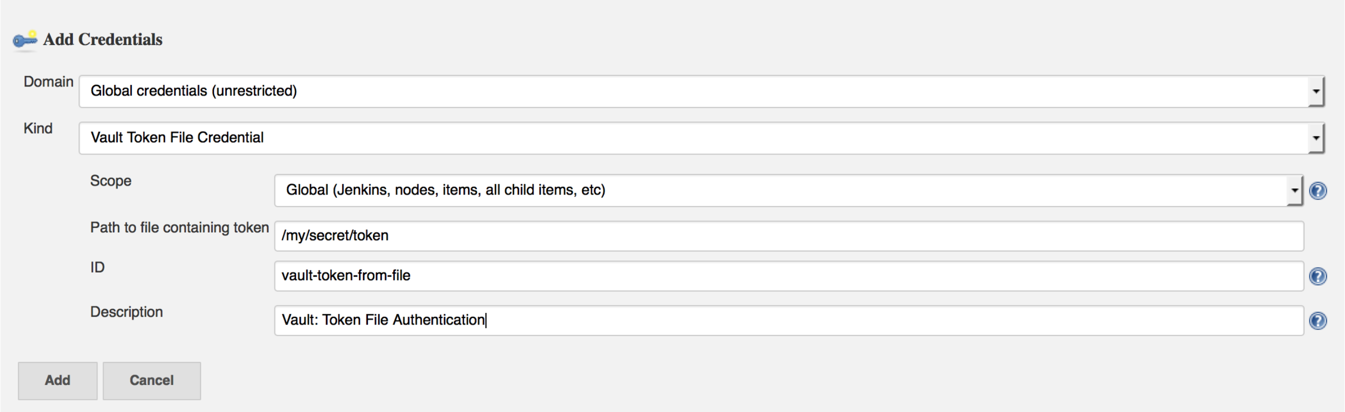
Token: ••••••••••••••••••••

ID: vault-token

Description: Vault: Token Authentication

Directly specify a token to be used when authenticating with vault.

Vault Token File Credential



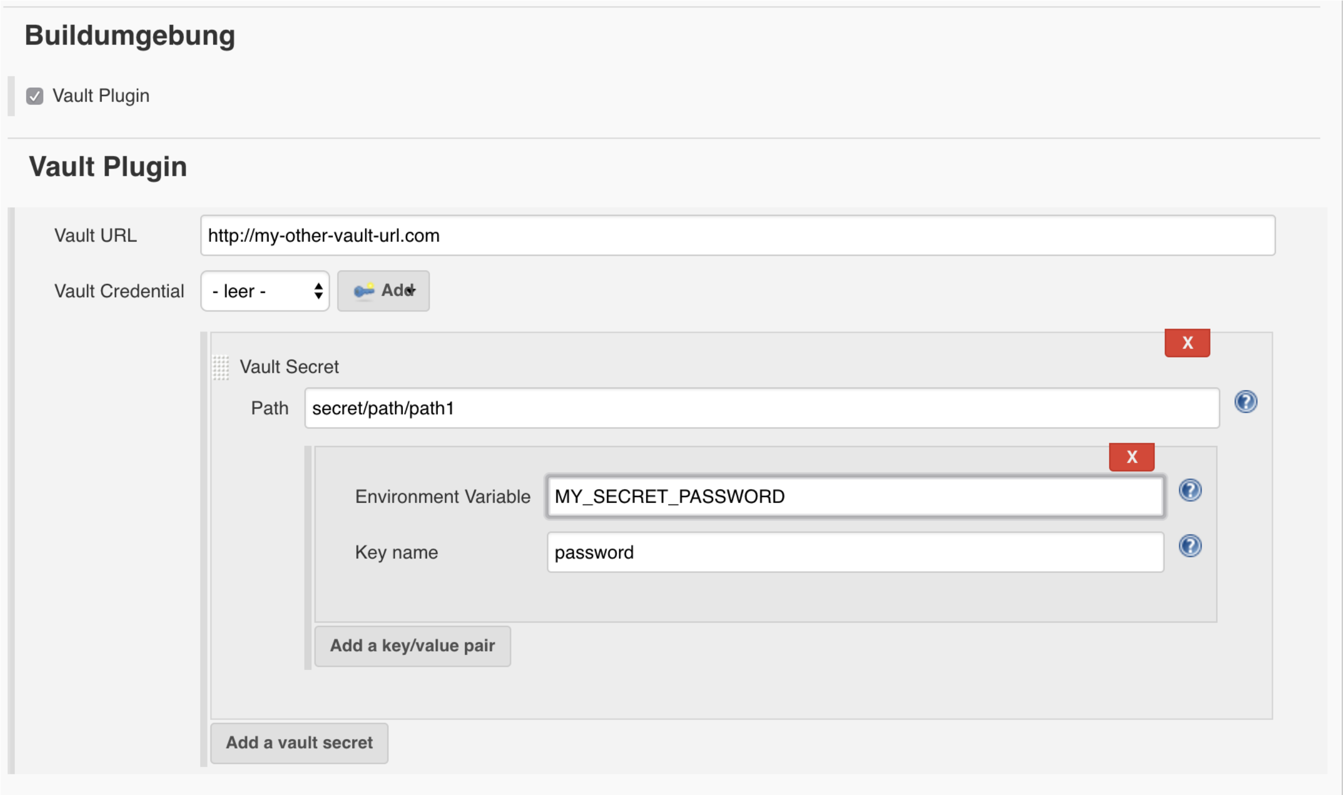
Add Credentials

Domain	Global credentials (unrestricted)
Kind	Vault Token File Credential
Scope	Global (Jenkins, nodes, items, all child items, etc)
Path to file containing token	/my/secret/token
ID	vault-token-from-file
Description	Vault: Token File Authentication

Basically the same as the Vault Token Credential, just that the token is read from a file on your Jenkins Machine.
You can use this in combination with a script that periodically refreshes your token.

Usage in FreeStyle Jobs

If you still use free style jobs (hint: you should consider migrating to Jenkinsfile), you can configure both configuration and the secrets you need on the job level.



Buildumgebung

☑ Vault Plugin

Vault Plugin

Vault URL	http://my-other-vault-url.com
Vault Credential	- leer -

Vault Secret

Path	secret/path/path1
Environment Variable	MY_SECRET_PASSWORD
Key name	password

The secrets are available as environment variables then.

Usage via Jenkinsfile

With version 2.3.0, we added a "withVault" symbol and made "envVar" optional as shown in the second secretValue with "another_test" will use the vaultKey as the envVar.

Jenkinsfile

```
node {
    // define the secrets and the env variables
    def secrets = [
        [path: 'secret/testing', secretValues: [
            [envVar: 'testing', vaultKey: 'value_one'],
            [envVar: 'testing_again', vaultKey: 'value_two']]],
        [path: 'secret/another_test', secretValues: [
            [vaultKey: 'another_test']]]
    ]

    // optional configuration, if you do not provide this the next higher configuration
    // (e.g. folder or global) will be used
    def configuration = [vaultUrl: 'http://my-very-other-vault-url.com',
                         vaultCredentialId: 'my-vault-cred-id']
    // inside this block your credentials will be available as env variables
    withVault([configuration: configuration, vaultSecrets: secrets]) {
        sh 'echo $testing'
        sh 'echo $testing_again'
        sh 'echo $another_test'
    }
}
```

Before version 2.2.0 and below:

Jenkinsfile

```
node {
    // define the secrets and the env variables
    def secrets = [
        [$class: 'VaultSecret', path: 'secret/testing', secretValues: [
            [$class: 'VaultSecretValue', envVar: 'testing', vaultKey: 'value_one'],
            [$class: 'VaultSecretValue', envVar: 'testing_again', vaultKey: 'value_two']]],
        [$class: 'VaultSecret', path: 'secret/another_test', secretValues: [
            [$class: 'VaultSecretValue', envVar: 'another_test', vaultKey: 'another_test']]]
    ]

    // optional configuration, if you do not provide this the next higher configuration
    // (e.g. folder or global) will be used
    def configuration = [$class: 'VaultConfiguration',
                         vaultUrl: 'http://my-very-other-vault-url.com',
                         vaultCredentialId: 'my-vault-cred-id']
    // inside this block your credentials will be available as env variables
    wrap([$class: 'VaultBuildWrapper', configuration: configuration, vaultSecrets: secrets]) {
        sh 'echo $testing'
        sh 'echo $testing_again'
        sh 'echo $another_test'
    }
}
```

In the future we might migrate to a BuildStep instead of a BuildWrapper.

CHANGELOG

Change log moved to Github releases

- **2016/08/02** - Initial release - 1.0
- **2016/08/11** - Bugfix release - 1.1
    - Refactor to allow getting multiple vault keys in a single API call (JENKINS-37151)
- **2016/08/15** - Re-release due to failed maven release - 1.2
- **2017/03/03** - Feature Release - 1.3
    - Vault Plugin should mask credentials in build log (JENKINS-39383)
- **2017/04/10** - Feature Release - 1.4
    - Support reading Vault Token from file on disk (JENKINS-37713)

    - Using credentials plugin for authentication token (JENKINS-38646)
- **2017/04/27** - Major Release - 2.0.0
    - Breaking change release (AppRole auth backend, Folder ability, improved configuration, ...)
- **2017/05/19** - Bugfix Release - 2.0.1
    - Build fails if plugin is enabled for a job without secrets specified (JENKINS-441630)
- **2017/05/22** - Feature Release - 2.1.0
    - Vault Key Not Saved In Vault Error Messaging (JENKINS-38647)
    - Add support github token auth (JENKINS-38939)
- **2018/05/01 -** Bugfix Release - 2.1.1
    - MaskingConsoleLogFilter should filter out null secrets (JENKINS-38647)
    - Avoid NPE Crash
    - Switch to SimpleBuildWrapper for pipeline compatibility (JENKINS-48049)
    - Dynamic secrets should be revoked after build wrapper completes (JENKINS-46794)

TODO

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
🟥	JENKINS-60440	Invalid git username /password on Jenkins agent when using Vault Username-Password Credential with '@' in username	Unassigned	Gordon Li	⇈	OPEN	Unresolved	Dec 11, 2019	Dec 14, 2019	
🟥	JENKINS-60091	HashiCorp Vault plugin using approle is not working since v3.0.0	Peter Tierno	Christophe Le Guern	⊘	OPEN	Unresolved	Nov 07, 2019	Nov 08, 2019	
➕	JENKINS-59902	Additional credential types for HashiCorp Vault plugin	Joseph Petersen	Tomas Zvala	⇊	OPEN	Unresolved	Oct 23, 2019	Oct 30, 2019	
⬆	JENKINS-59847	Hashicorp Vault plugin - CASC - approle path not configurable	Joseph Petersen	Emmanuel Cornette	⇊	IN PROGRESS	Unresolved	Oct 18, 2019	Oct 30, 2019	
🟥	JENKINS-59836	HashiCorp Vault plugin configuration cannot be enabled for Freestyle projects	Peter Tierno	Kalana Samaraweera	⇈	OPEN	Unresolved	Oct 18, 2019	Jan 15, 2020	
➕	JENKINS-59085	Ability use vault plugin in combination with other plugins that require credentials	Peter Tierno	Bjørn Åge Tungesvik	⇊	OPEN	Unresolved	Aug 26, 2019	Dec 14, 2019	
➕	JENKINS-52895	Ability to read all key /value pairs at /secret/: path	Peter Tierno	Torsten Reinhard	⇊	OPEN	Unresolved	Aug 06, 2018	Aug 06, 2018	
🟥	JENKINS-52889	new lines in secret is substituted by spaces when reading via Vault plugin	Peter Tierno	Hee Won Kim	⇊	OPEN	Unresolved	Aug 05, 2018	Jul 29, 2019	
🟥	JENKINS-48892	Vault plugin exits on specifying a secret that is a large base64 encoded value	Peter Tierno	Matt Snoby	⇈	REOPENED	Unresolved	Jan 10, 2018	Jun 13, 2018	
☑	JENKINS-45685	Declarative Pipeline Example	Peter Tierno	Nikolay Tsutsarin	⇈	IN PROGRESS	Unresolved	Jul 20, 2017	Oct 18, 2019	
⬆	JENKINS-39374	Add ability to get SCM authentication tokens from Vault	Richard Vodden	Petrik van der Velde	⇊	OPEN	Unresolved	Oct 31, 2016	Dec 14, 2019	
🟥	JENKINS-15912	Shell script files output is not displayed	Rakesh Singh	Markus	⇈	REOPENED	Unresolved	Nov 23, 2012	Feb 09, 2018	

12 issues