Venafi DevOps Integrations

Venafi Terraform Provider

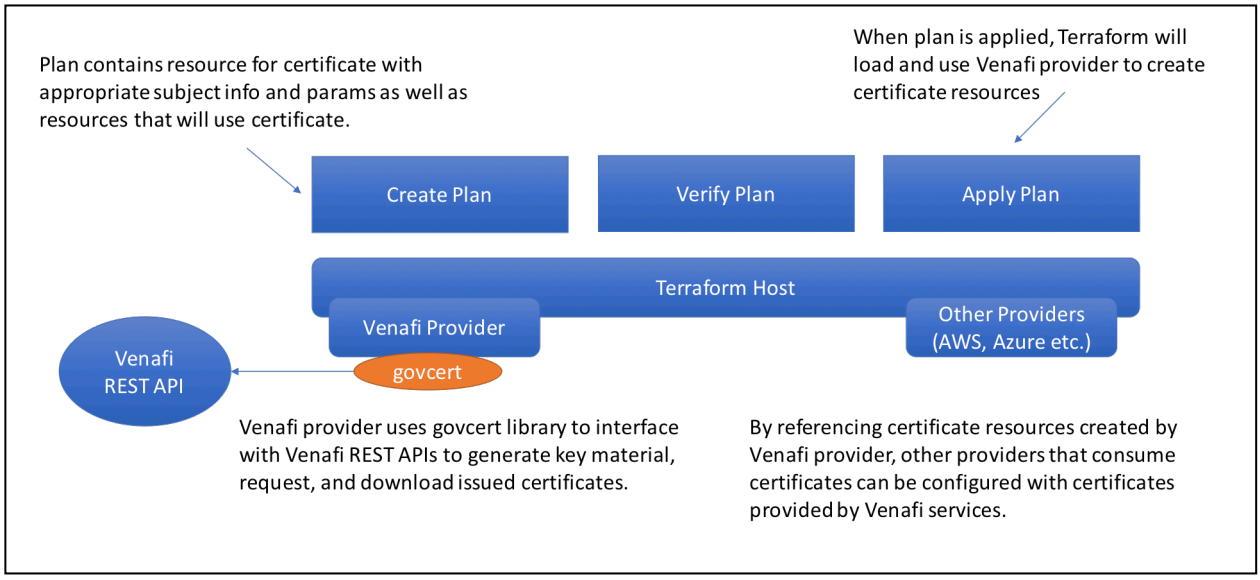
Version	Date	Description	Author
V1	March 20, 2017	Initial Version	Walter Goulet
V2	April 11, 2017	Updates with troubleshooting	Walter Goulet
V3	June 22, 2017	Trust Protection Platform additions	Ryan Treat
V4	June 23, 2017	Update URLs	Walter Goulet

Introduction

The Venafi Terraform Provider allows Terraform users to seamlessly incorporate certificate management into Terraform plans when orchestrating services.

Overview

The Venafi Terraform Provider is a custom provider that wraps the Venafi GoVCert library. This library is a lightweight library that performs key generation and interfaces with REST APIs provided by various Venafi products. The provider can be referenced via Terraform plans as any other provider. The following diagram



outlines the high-level solution architecture.

Registering with Venafi

To register for an account with Venafi, use the 'curl' utility to submit a user registration request to Venafi. A confirmation email will be sent to the address in the request. When the request is confirmed, the API key in the email can be provided in your variables file for certificate enrollment.

```
curl -H 'Content-Type:application/json' -d
'{"username":"ctest4wg@venafi.com","userAccountType":"API"}'
https://api.venafi.cloud/v1/useraccounts
```

Installing the Provider

The Venafi Terraform provider is a third-party provider written in Go. The provider must be compiled and placed on your Terraform host in your path so it can be located by Terraform.

Ensure that you have Go 1.7 and Terraform 0.8.8 or higher installed on your system, as well as git and make installed. The provider has been tested on Centos 7 but should work on any modern Linux or Mac OS.

Build GoVCert

The GoVCert library is shipped separately from the Venafi Terraform provider. Clone the GoVCert library on your system and build it as follows:

```
go get -u github.com/jteeuwen/go-bindata/...
go get github.com/Venafi/govcert/...
```

Note: Some build errors will be shown; these can be safely ignored as the built binaries are included in the repo

Install the Terraform provider

Download and build the Terraform provider as follows:

```
go get github.com/Venafi/terraform-provider-venafi/...
go build
```

Copy the resulting 'terraform-provider-venafi' binary to the same location as the 'terraform' binary itself.

Using the Provider

NOTE: Currently the Venafi Terraform provider does not encrypt private key material that is generated by terraform plans. Be sure to configure the terraform host to restrict unauthorized access.

Create provider

To use the Terraform provider create a `venafi.tf` file. In here you must first create the provider. This would be done by creating a provider block.

(For Venafi Cloud)

```
provider "venafi" {
    url = “<ALT_VENAFI_CLOUD_URL>”  // optional, defaults to production URL
    api_key = "<API_KEY>"
    zone    = "<ZONE>"  // optional, defaults to 'default'
}
```

(For Venafi Trust Protection Platform)

```
provider "venafi" {
    url = "<TPP_WEBSDK_URL>"
    tpp_username = "<TPP_WEBSDK_USERNAME>"
    tpp_password = "<TPP_WEBSDK_PASSWORD>"
    zone = "<TPP_POLICY_TREE_FOLDER>"  // optional, defaults to “\VED\Policy\Default”
}
```

Creating a Certificate Signing Request

NOTE: The Venafi Cloud service by default issues certificates intended for test and development purposes. In the default configuration, certificates that are issued will be valid only for the following test domains:

[subdomain].example.com
[subdomain].example.org
[subdomain].example.net
[subdomain].invalid
[subdomain].local
[subdomain].localhost
[subdomain].test

Where [subdomain] is the subdomain of the registered Venafi Cloud user’s email address. For example, if the user registers with email address jdoe@mydemocorp.com, certificate requests submitted to the Venafi Cloud service must match one of the following patterns:

**.mydemocorp.example.com*
**.mydemocorp.example.org*

**.mydemocorp.example.net*
**.mydemocorp.invalid*
**.mydemocorp.local*
**.mydemocorp.localhost*
**.mydemocorp.test*

Contact Venafi at support@venafi.com if support for certificates for use in production environments is required.

Certificate signing requests can be created using the `venafi_csr` resource. This resource only has 1 required field.

- `common_name` (string)

The following optional fields can also be set

field	type
`organizational_unit`	string array
`organization_name`	string
`country`	string
`state`	string
`locality`	string
`key_password`	string
`san_dns`	string array
`san_email`	string array
`san_ip`	string array

After creation this resource will expose 2 further fields:

field	type
private_key_pem	string
csr_pem	string

The following example would output the created private key and CSR:

```
provider "venafi" {
    api_key = "XXXX"
    zone = "Default"
}

resource "venafi_csr" "webserver" {
    common_name = "web.organization.localhost"
    san_dns = ["blog.organization.localhost", "contact.organization.localhost"]
    organizational_unit = ["appdev", "webdev"]
    state = "London"
    country = "UK"
}

output "csr" {
    value = "${venafi_csr.webserver.csr_pem}"
}

output "csr_private_key" {
    value = "${venafi_csr.webserver.private_key_pem}"
}
```

Creating a Certificate / Private Key pair

Certificates can be created using the `venafi_certificate` resource. This resource only has 1 required field.

- `common_name` (string)

The following optional fields can also be set:

field	type
`algorithm`	string [RSA or ECDSA]
`rsa_bits`	integer (Used when `algorithm=RSA`)
`ecdsa_curve`	string (Used when `algorithm=ECDSA`)
`san_dns`	string array
`san_email`	string array
`san_ip`	string array
`key_password`	string

After creation, this resource will expose 3 further fields:

field	type
`private_key_pem`	string
`chain`	string
`certificate`	string

The following example would output the created private key, certificate, and chain:

```
provider "venafi" {
    api_key = "XXXX" // your Venafi Cloud API key
    zone    = "Default" //optional. Defaults to 'Default'
```

```
}

resource "venafi_certificate" "webserver" {
    common_name = "web.organization.localhost"
    algorithm = "RSA"
    rsa_bits = "2048"
    san_dns = [
        "web01.organization.localhost",
        "web02.organization.localhost"
    ]
    key_password = "secretpassword"
}

output "cert_certificate" {
    value = "${venafi_certificate.webserver.certificate}"
}

output "cert_chain" {
    value = "${venafi_certificate.webserver.chain}"
}

output "cert_private_key" {
    value = "${venafi_certificate.webserver.private_key_pem}"
}
```

Troubleshooting

If the 'terraform apply' command fails with an exit code 1, the error is most likely due to incorrect authentication parameters (using an incorrect API key for Venafi Cloud, or incorrect username, password, or URL for Venafi Trust Protection Platform). Verify your authentication parameters are correct, then try again.

Known Issues

The current version of the Terraform provider does not output the request ID for submitted certificates. As a result, if there is an issue with the certificate request, the Terraform plan will not display an obvious error to the user. The easiest resolution to this is to output all certificate requests submitted to Venafi Cloud using another REST client and locate the certificate request that was submitted (based on the subject name supplied in the Terraform plan). The certificate request response will contain information that can be used to help troubleshoot the error.

```
curl -H 'Content-Type: application/json' -H 'tppl-api-key: XXXXX
https://api.staging.venafi.io/v1/certificaterequest
```