TFE Technical Enablement

TFE Automated Install

HashiCorp

Automated TFE Installation

- The installation of Terraform Enterprise can be automated for both online and airgapped installs.

- There are two parts to automating the install:

  - Configuring Replicated: the platform which runs Terraform Enterprise
  - Configuring Terraform Enterprise itself

Automated TFE Installation Prequisites

Before starting the install process, you must:

- prepare an application settings file, which defines the settings for the Terraform Enterprise application
- prepare /etc/replicated.conf, which defines the settings for the Replicated installer
- copy your license file to the instance
- download the .airgap bundle to the instance (Airgapped mode only)

Automated TFE Installation Prequisites

- You may also need to provide additional flags in order to avoid being prompted for those values when running the installer
  - Example: the instance's public and private IP addresses
  - If unspecified, this could result in either a failure of the installer or a unbounded delay while waiting for input

Full List of Pre-requisites

Remember when we first did the installation in the Lab? There was a full list of pre-requisites to validate prior to the installation.

Pre-Install List	Description
Choose Operational Mode	Decide how Terraform Enterprise should store its data. This is affected by your choice of deployment method.
Credentials	Ensure you have a Terraform Enterprise license and a TLS certificate for Terraform Enterprise to use.
Data Storage	Depending on your operational mode, prepare data storage services or a block storage device.
Linux Instance	Prepare a running Linux instance for Terraform Enterprise.
Network Requirements	Terraform Enterprise is a networked application. Its Linux instance(s) needs to allow several kinds of incoming and outgoing network traffic.
SSL Certificate	Both for the host itself and integrations (VCS, provisioning endpoint, etc)
Pre-define Application Configuration	prepare an application settings file, which defines the settings for the Terraform Enterprise application.
Set Config Files	prepare /etc/replicated.conf, which defines the settings for the Replicated installer.
License File	copy your license file to the instance.
Download Software	download the .airgap bundle to the instance (Airgapped mode only).

Application settings

This file contains the values you would normally provide in the settings screen

This could be as simple as choosing the Demo installation type, or as complex as specifying the PostgreSQL connection string and S3 bucket credentials and parameters.

You need to create this file first since it is referenced in the ImportSettingsFrom property in /etc/replicated.conf

Application settings

The settings file is JSON formatted. All values must be strings. The example below is suitable for a Demo installation:

```
{
    "hostname": {
        "value": "terraform.example.com"
    },
    "installation_type": {
        "value": "poc"
    },
    "capacity_concurrency": {
        "value": "5"
}}
```

How to Pre-Configure

By using a combination of documentation and a test and repair model (trying an install, like in the lab!) you can get all the variables needed for the installation.

Once you configure all the settings the way you want them, you can SSH in and export the settings in JSON format for use in an automated install.

```
tfe$ replicatedctl app-config export > settings.json
```

Config Example

Once you run the export command you'll see the configuration below as a sample;

```
tfe$ replicatedctl app-config export > settings.json
```

```
cat settings.json
{
    "aws_access_key_id": {},
    "aws_instance_profile": {},
    "aws_secret_access_key": {},
}
```

Full Config List

For a full list of variables you can review this documentation

It is expected the user will have completed a manual installation first and will already be familiar with the nature of these parameters from the settings screen.

Next Steps

So as a process of automating the installation, we've gathered all the pre-requisites, we've attempted one install, finally we need to configure Replicated for an automated install.

Installer Settings - Online

The following is an example /etc/replicated.conf suitable for an automated online install using a certificate trusted by a public or private CA. ImportSettingsFrom must be the full path to the application settings file. You also need to provide the full path to your license file in LicenseFileLocation.

```
{
    "DaemonAuthenticationType":     "password",
    "DaemonAuthenticationPassword": "your-password-here",
    "TlsBootstrapType":             "server-path",
    "TlsBootstrapHostname":         "server.company.com",
    "TlsBootstrapCert":             "/etc/server.crt",
    "TlsBootstrapKey":              "/etc/server.key",
    "BypassPreflightChecks":        true,
    "ImportSettingsFrom":           "/path/to/settings.json",
    "LicenseFileLocation":          "/path/to/license.rli"
}
```

Invoking the Online Installation Script

Once /etc/replicated.conf has been created, you can retrieve and execute the install script as root:

```
curl -o install.sh https://install.terraform.io/ptfe/stable
bash ./install.sh \
    no-proxy \
    private-address=1.2.3.4 \
    public-address=5.6.7.8
```

Note the private-address and public-address flags provided to the installer. If the instance will not have a separate public address, you may provide the private address for both values.

Installer Settings - Offline

The following is an example /etc/replicated.conf suitable for an automated airgapped install, which builds on the online example above. Note the addition of LicenseBootstrapAirgapPackagePath, which is a path to the .airgap bundle on the instance.

```
{
    "DaemonAuthenticationType":          "password",
    "DaemonAuthenticationPassword":      "your-password-here",
    "TlsBootstrapType":                  "server-path",
    "TlsBootstrapHostname":              "server.company.com",
    "TlsBootstrapCert":                  "/etc/server.crt",
    "TlsBootstrapKey":                   "/etc/server.key",
    "BypassPreflightChecks":             true,
    "ImportSettingsFrom":                "/path/to/settings.json",
    "LicenseFileLocation":               "/path/to/license.rli",
    "LicenseBootstrapAirgapPackagePath": "/path/to/bundle.airgap"
}
```

Invoke the Offline Installation Script

Invoking the airgap installation script Following on from the manual airgapped install steps, you must also have the installer bootstrapper already on the instance. For illustrative purposes, it is assumed the installer bootstrapper has been unarchived in `/tmp`.

Once `/etc/replicated.conf` has been created, you can now execute the install script as root:

```
cd /tmp
./install.sh \
    airgap \
    no-proxy \
    private-address=1.2.3.4 \
    public-address=5.6.7.8
```

Waiting for Terraform Enterprise

Once the installer finishes, you may poll the `/_health_check` endpoint until a 200 is returned by the application, indicating that it is fully started:

```
while ! curl -ksfS --connect-timeout 5 https://tfe.example.com/_health_c
    sleep 5
done
```

Lab: Automated Install

Complete the entire track

https://play.instruqt.com/hashicorp-chip-te/tracks/tfe-module-install

HashiCorp

Chapter Summary

- Terraform Enterprise supports both a manual and automated installation
- To pre-valdiate the installation its best to do a manual installation and use those values
- Review all the pre-reqs for both the manual install and the automated install before getting started
- Double check the reference links next to see all the configuration variables

Reference links

- How to do Automated Install
- Replicated Configuration
- Variables