TFE Technical Enablement

E1 Training Course

HashiCorp

Table of Contents

TFE Technical Enablement

TFE Architecture Overview

HashiCorp

TFE Components

1. **Application Layer**
    - TFE Core - Rails Application for web front end and background workers
    - TFE Services & Terraform workers - Go services that function in isolated execution environments.
2. **Coordination Layer**
    - Redis - Caching and coordination between web and background workers in the application layer
    - Rabbit MQ - TF worker coordination
3. **Storage Layer**
    - PostgreSQL DB - Stores application data for workspace/user settings
    - Blob Storage - Stores TF state files, logs, etc.
    - Vault - Encrypts sensitive data
    - Config Data - Replicated admin console configuration
4. **Network Service** - NGINX

Operational Mode Decision

TFE Operational Mode determines how data is persisted.

The operational mode is selected at install time and cannot be changed once Terraform Enterprise is running.

- **Production - External Services**
  - Store the majority of the stateful data in an external DB and external object storage. Best for users with access to managed services.
- **Production - Mounted Disk**
  - Store data in a separate directory on the host which stores data on an external disk. Best for users with experience mounting performant block storage.
- **Demo**
  - All data is stored on the instance. Suitable for testing and validation only.

Architecture Components

- **Virtual Compute**
  - Replicated Container, System Configuration and Application Configuration
- **External PostgreSQL Database**
  - Responsible for the secure storage of Application Configuration data, workspace settings and Vault data
- **S3 Compatible Storage/Blob Storage**
  - Terraform artifacts including plan results, state and logs
- **External Vault Instance** (if used)
  - Transit keys for encryption of PostgreSQL data

Virtual Compute Requirements

TFE Instance Requirements

- Recommended Compute Requirements
  - 2 CPU, 4-8 core, 16-32 GB RAM, 50GB Storage
- Supported Operating Systems
  - Most Linux Operating Systems. List

PostgreSQL Requirements

- Recommended Compute Requirements
  - 2 CPU, 4-8 core, 16-32 GB RAM, 50GB Storage
- PostgreSQL Version - 9.4, 9.5, 9.6, 10.x, 11.x
- Lower system requirements can be used for non-production and testing

Object Storage

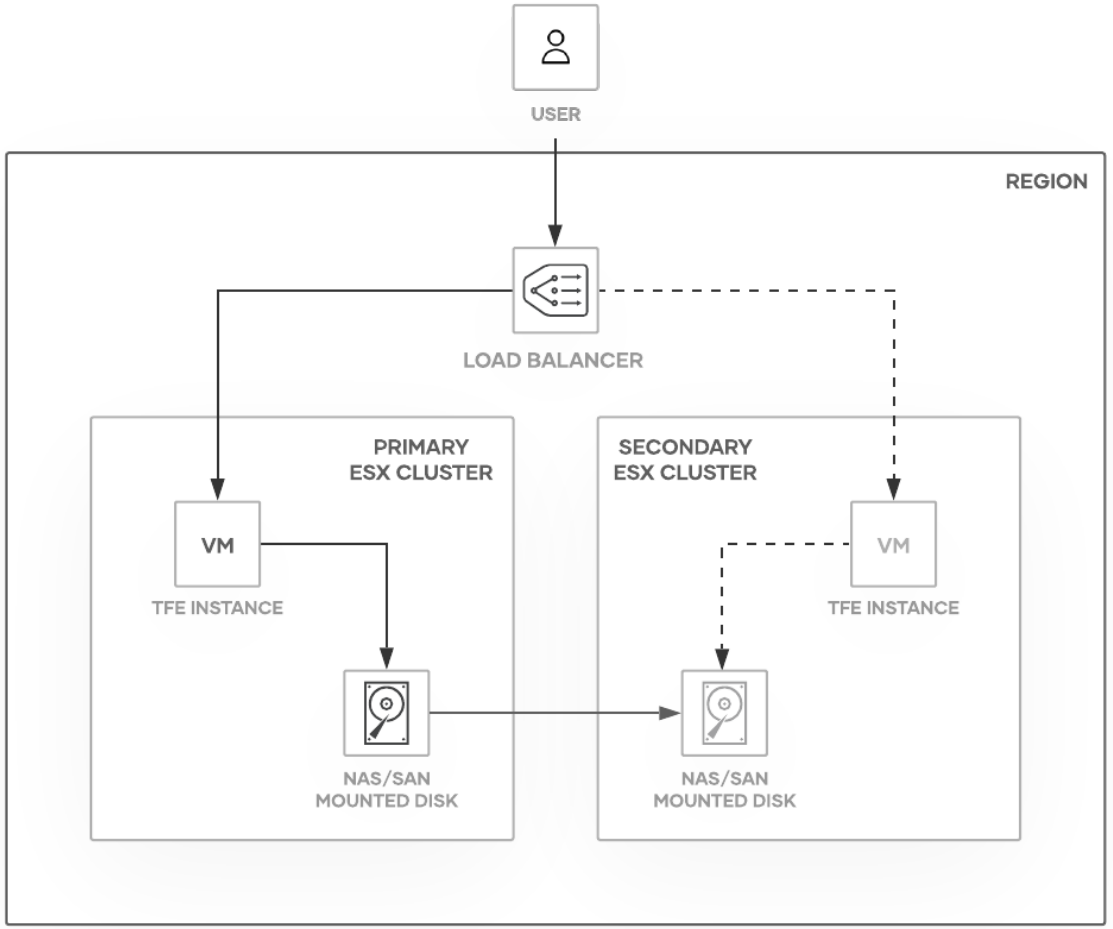
- An S3 Standard bucket, or compatible storage
  - AWS S3
  - Google Cloud Storage
  - Azure Blob Storage
  - Minio or Ceph
- Be aware of high availability capabilities for each cloud

Terraform Enterprise Reference Architecture

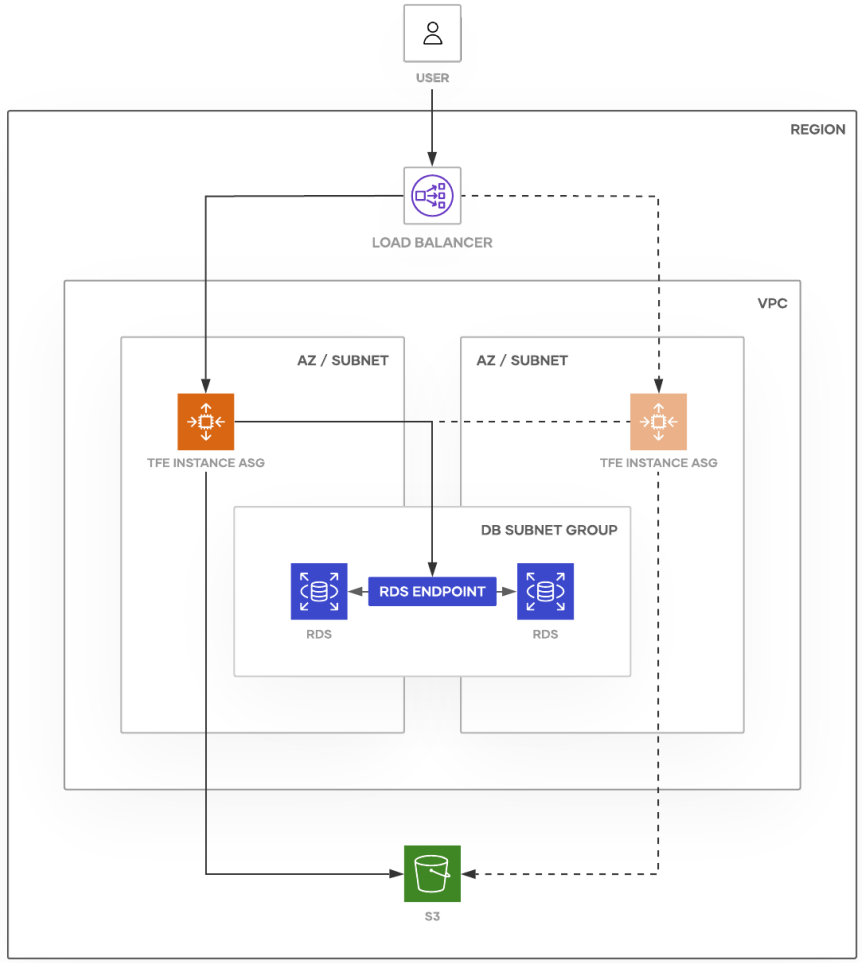
Depending on where you choose to deploy Terraform Enterprise, there are different services available to maximize the resiliency of the deployment.

For example, most major cloud service providers offer a resilient RDS offering, removing the need to manage a complex database cluster/failover architecture.

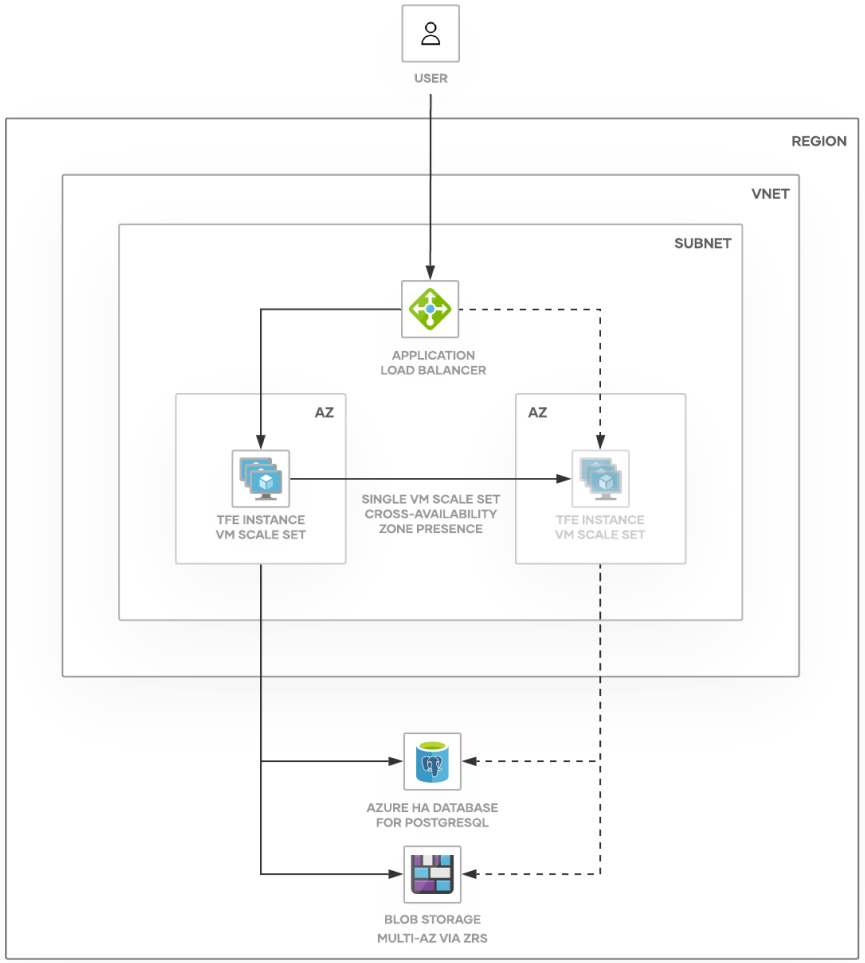
TFE VMware Reference Architecture



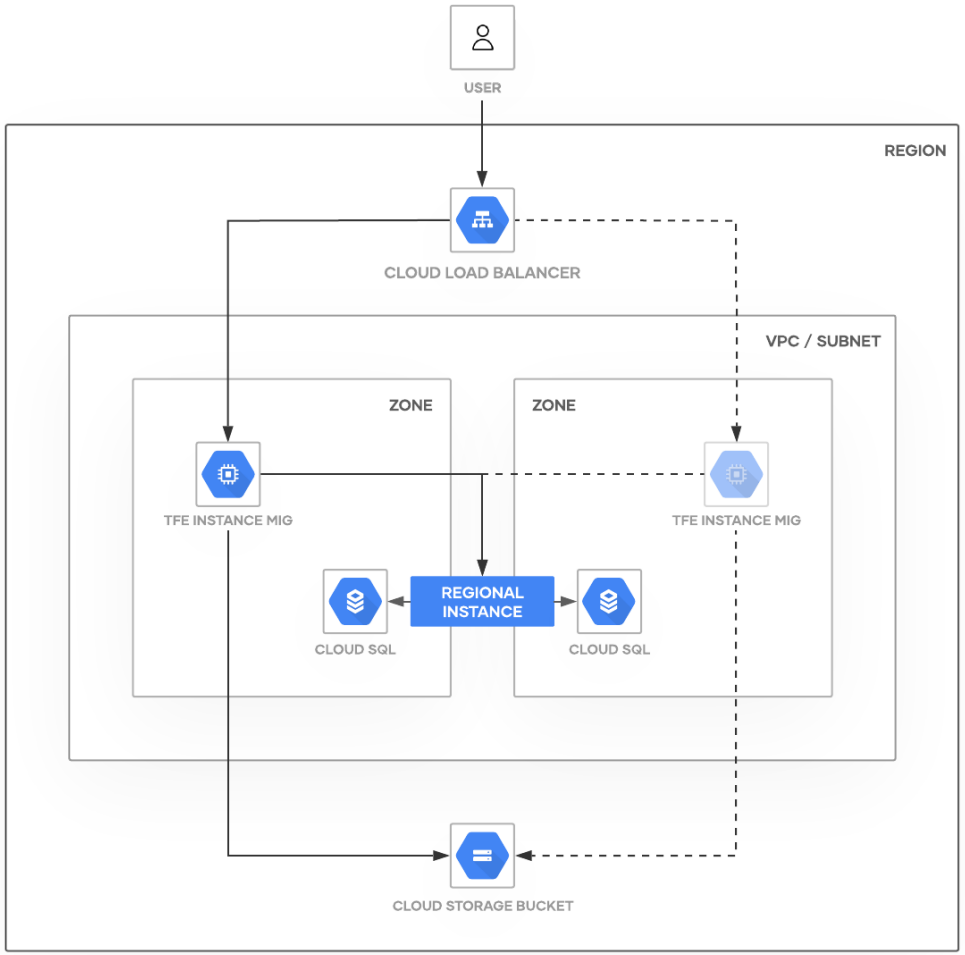
TFE AWS Reference Architecture



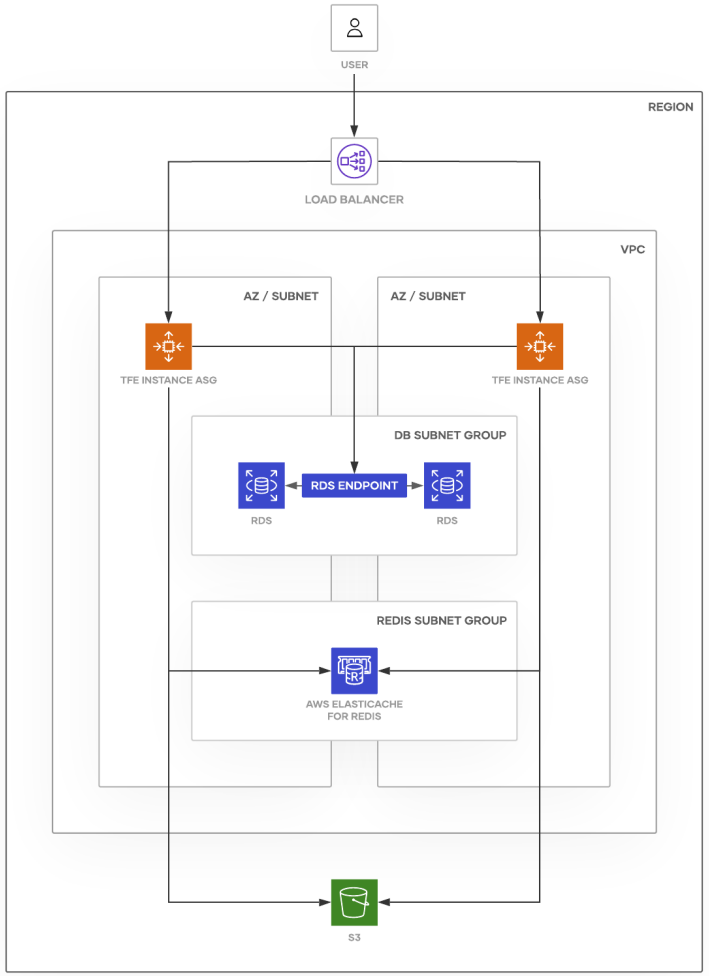
TFE Azure Reference Architecture



TFE GCP Reference Architecture



TFE AWS Active/Active Reference Architecture



External Vault Server Requirements

- Terraform Enterprise configures and manages an internal vault instance.
- Organizations with exceptional data encryption and record keeping requirements can configure an external Vault ~~server~~ cluster.
- A production Terraform instance will be dependent on a highly available (production grade) Vault cluster.

Data Storage and Security Considerations

Object	Storage	Encrypted
Ingressed VCS Data	Blob Storage	Vault Transit Encryption
Terraform Plan Result	Blob Storage	Vault Transit Encryption
Terraform State	Blob Storage	Vault Transit Encryption
Terraform Logs	Blob Storage	Vault Transit Encryption
Terraform Environment Variables	PostgreSQL	Vault Transit Encryption
Organization Settings	PostgreSQL	No
Account Password	PostgreSQL	bcrypt
2FA Recovery Codes	PostgreSQL	Vault Transit Encryption
SSH Keys	PostgreSQL	Vault Transit Encryption

Data Storage and Security Considerations

Object	Storage	Encrypted
User/Team/Organization Tokens	PostgreSQL	HMAC SHA512
OAuth Client ID + Secret	PostgreSQL	Vault Transit Encryption
OAuth User Tokens	PostgreSQL	Vault Transit Encryption
Twilio Account Configuration	PostgreSQL	Vault Transit Encryption
SMTP Configuration	PostgreSQL	Vault Transit Encryption
SAML Configuration	PostgreSQL	Vault Transit Encryption
Vault Unseal Key	PostgreSQL	ChaCha20+Poly1305

Terraform Enterprise Architecture

Fault Tolerance

HashiCorp

Failure Tolerance by Application Tier

Each component of Terraform Enterprise has its own Fault Tolerance.

- **Terraform Enterprise Application Servers**
  - Through deployment of two virtual machines in different ESX clusters/zones, the Terraform Enterprise Reference Architecture is designed to provide improved availability and reliability.
  - Combined with a standard load balancer configuration for health checking against the instance the application servers are very robust

Failure Tolerance by Application Tier

- **PostgreSQL Database**
  - Using a PostgreSQL cluster will provide fault tolerance at the database layer.
- **Object Storage**
  - Each cloud implements high availability of their object storage differently
  - Be sure to review your specific cloud (or VMWare configuration) and ensure you are using a supported highly available configuration
- **Vault Servers**
  - HashiCorp Vault supports a highly available cluster model. Please review the documentation

Reference links

- Infranstructure Components
- Architecture Reference Diagrams

TFE Installation Key Concerns

HashiCorp

Replicated

Replicated provides SaaS vendors a container-based platform for easily deploying their cloud-native applications inside customers' environments. It provides license management, online and airgap installation, and a control panel for your docker-based application.

Installation Checklist

When preparing for an installation of Terraform Enterprise collect the following details:

- Install type: Airgapped/Offline or Online?
- Identify if there is a Proxy?
- TLS Configuration
  - TLS is mandatory for a lot of the communication of TFE as well as the host itself
  - Properly configuring the CA chain of trust is critical
- Custom TFE Worker Image
  - Many customers choose to preload their own worker image
  - Creation pipeline, storage, installation
- Operational Mode

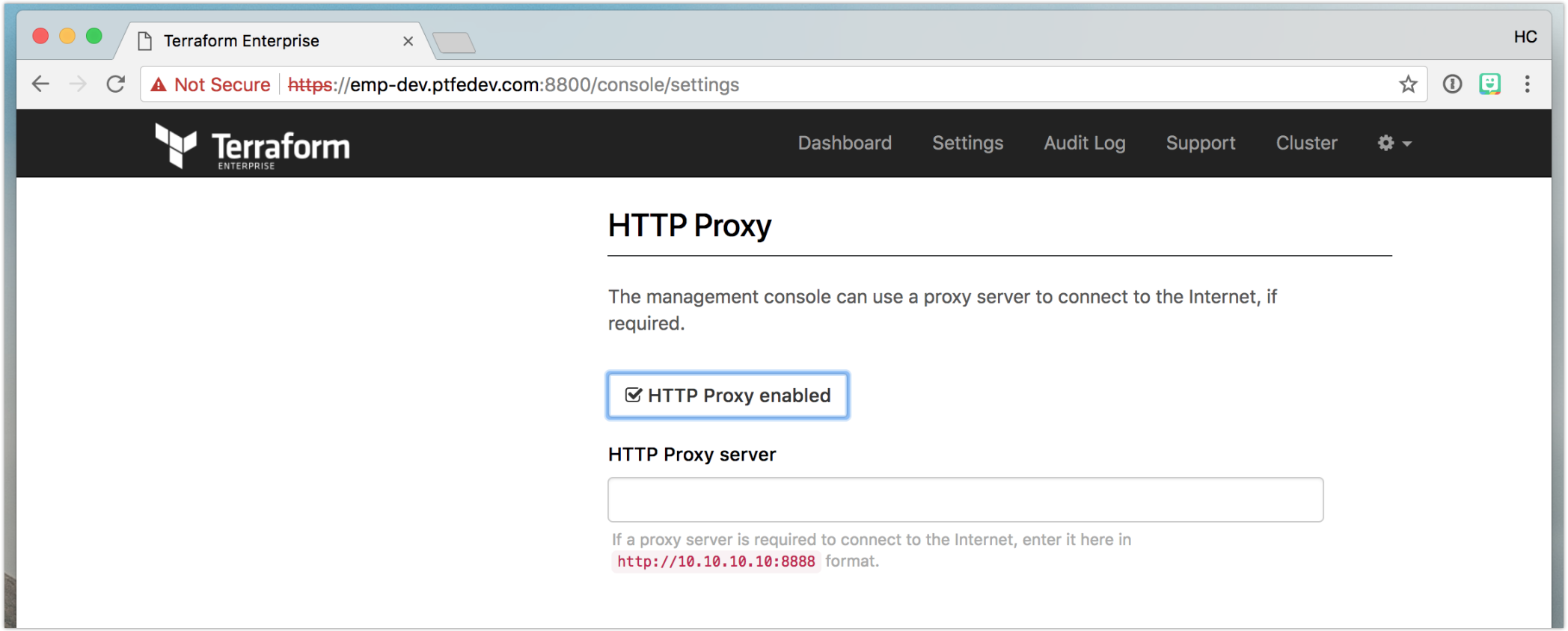
Network Requirements for Terraform Enterprise

The Terraform Enterprise server needs to allow several kinds of incoming connections. It also needs to access several external services.

- Source - User/Client/VCS
  - 443: To access the Terraform Enterprise application via HTTPS
- Source - Administrators
  - 22: To access the instance via SSH from your computer. SSH access to the instance is required for administration and debugging.
  - 8800: To access the installer dashboard.
- Source - TFE Server(s)
  - 9870-9880 (inclusive): For internal communication on the host and its subnet; not publicly accessible.
  - 23000-23100 (inclusive): For internal communication on the host and its subnet; not publicly accessible.
- A few others for internal services. Full List

Proxies with TFE

Both the GUI and the Installer script accept input for Proxy configuration

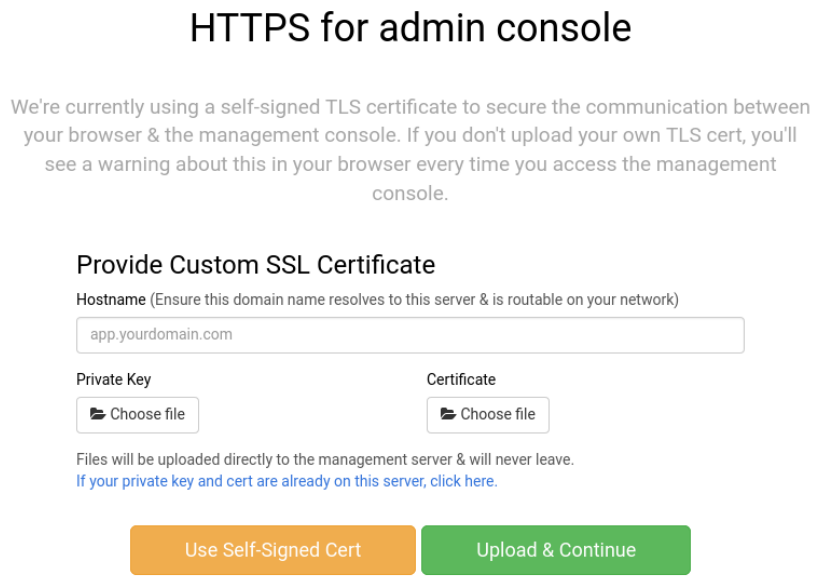


Proxy Special Notes

- The proxy configuration can be updated post install via the Replicated Console
- If you plan on running the installer again or want the configuration validation to pass, proxy config for Replicated components needs to be edited on the host. TFE Docs
- Docker's proxy config will be also need to be updated on the host

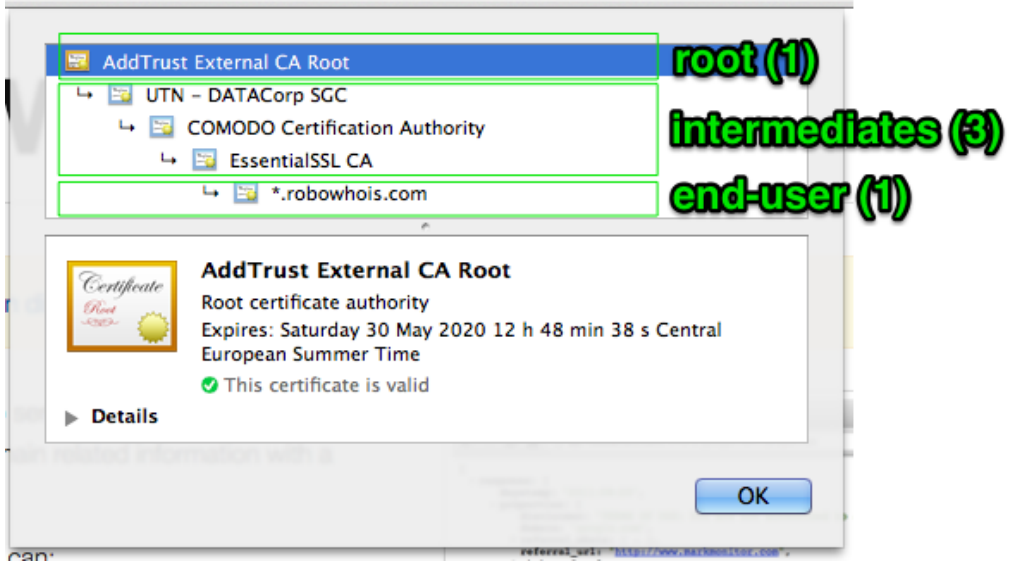
TLS Configuration

Both the GUI and the Installer script accept input for TLS Configuration. Note There are two sections for TLS configuration; the "TLS Key & Cert" section and the "SSL/TLS Configuration" section



Quick Certificate Refresher

- TLS certificates are issued by Certificate Authorities (CAs) to establish identity and setup secure communication
- There are two types of CAs: root CAs and intermediate CAs. Combined they represent a Certificate Chain.
- In order for a TLS certificate to be trusted, that certificate must have been issued by a CA that is trusted by the device that is connecting.



TLS Configuration

- Terraform Enterprise requires a TLS certificate and a private key in order to operate.
- Certificate must match Terraform Enterprise's hostname, either by being issued for the FQDN or being a wildcard certificate.
- Certificate can be signed by a public or private CA, but it must be trusted by all of the services that Terraform Enterprise is expected to interface with:
  - VCS Provider
  - Any CI Systems
  - Notification systems configured to work with Terraform Enterprise
  - Build systems working with Terraform Enterprise

TLS Key & Cert

The "TLS Key & Cert" section is where the TLS private key and certificate can be configured to allow HTTPS connections to Terraform Enterprise

3 options for specifying Key & Cert:

1. **Self signed (generated)** - TFE automatically generates
2. **Server Path** - TFE loads from specified path on host
3. **Upload Files** - User uploads

- Both the TLS certificate and private key files must be PEM-encoded.

SSL/TLS Configuration

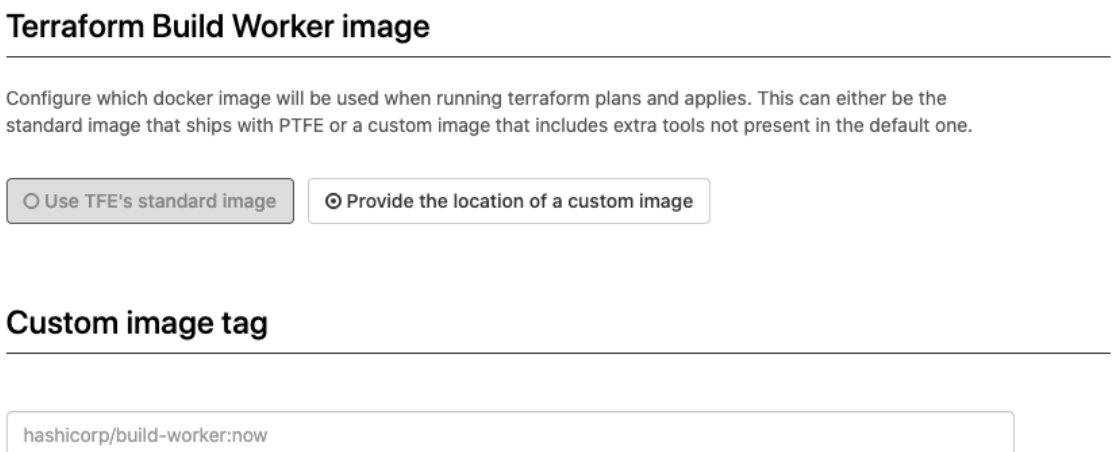
The "SSL/TLS Configuration" section is used to add custom trusted CAs so Terraform Enterprise can connect to services that use SSL/TLS certificates issued by them.

- This is used to allow Terraform Enterprise to trust Private CAs and securely connect to resources like VCS, DB etc.
- All certificates in the certificate signing chain, meaning the root certificate and any intermediate certificates, must be included
- All certificates must be PEM encoded and listed on after another.

If you see failures during VCS connection self-signed untrusted certs are your likely culprit.

Alternative Worker Image

- TFE runs terraform plan and terraform apply operations in a disposable Docker containers
- Default Docker image may not have additional tools used during Terraform runs
  - `local-exec`
  - `external` data source
- To allow use of these tools for any plan or apply, users can build their own image and configure TFE to use that instead.



Custom Worker Best Practices

- Manage the custom worker image as an artifact through a pipeline
- The base image must be `ubuntu:xenial`
- The image must exist on the Terraform Enterprise host.
- Required CA certificates must be added when building the image
- Terraform does not need be installed on the image. TFE provides at runtime.
- Alternative Worker images also offer extension points to execute scripts:
  - **Initialize Script** - Before `terraform init`
  - **Finalize Script** - After `plan` or `apply`

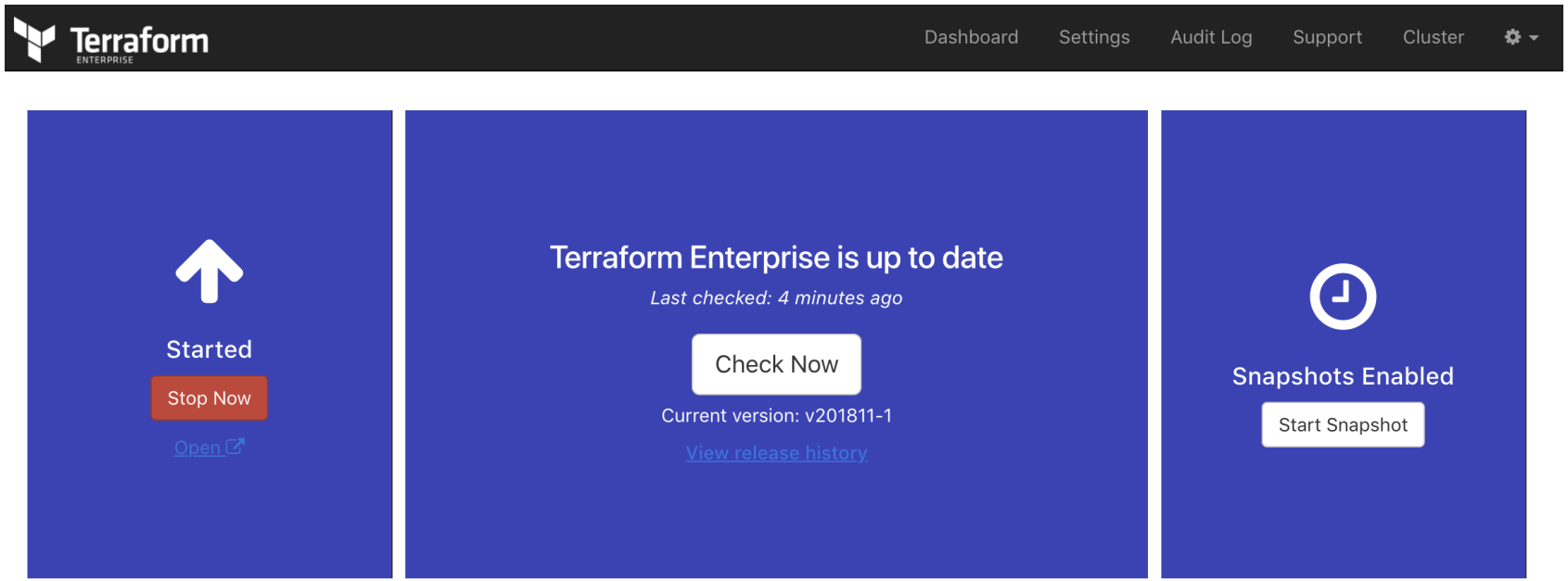
Installation Options

The installer can run in two modes, **Online** or **Airgapped**. Each of these modes has a different way of executing the installer, but the result is the same.

- After running the installer script, the remainder of the installation is done through a browser using the installer dashboard on port 8800 of the TFE instance.
- To complete the installation, you must be able to connect to that port via HTTPS.
- The installer uses an internal CA to issue bootstrap certificates.

Installation Mode - Online

- The standard way to install Terraform Enterprise
- `curl https://install.terraform.io/ptfe/stable | sudo bash`
- Installation continues in the browser!



Installation Mode - Airgapped

- Used to support offline environments, regulated environments, legal requirements
  - Connection on port 8800 is still required
- Airgapped installations require Docker to be pre-installed
- `.airgap` file with required artifacts is provided by Replicated

By default Terraform Enterprise does not include any providers and fetches them from the Public Registry as needed

- For airgapped installs these must be provided after install as `bundles`
- See Docs for details on how to operationalize this

Chapter Summary

- Terraform Enterprise has several supported architectures to support major Infrastructure platforms
  - AWS, Azure, GCP, VM*
- Reference the Installation Checklist in this chapter when you install
- Both online and offline installers are supported

Reference links

- How to Install Terraform Enterprise
- Pre-Install Checklist
- Offline and Online Installer
- Data Security Model