HashiCorp Terraform

Terraform Enterprise Architecture Deep Dive

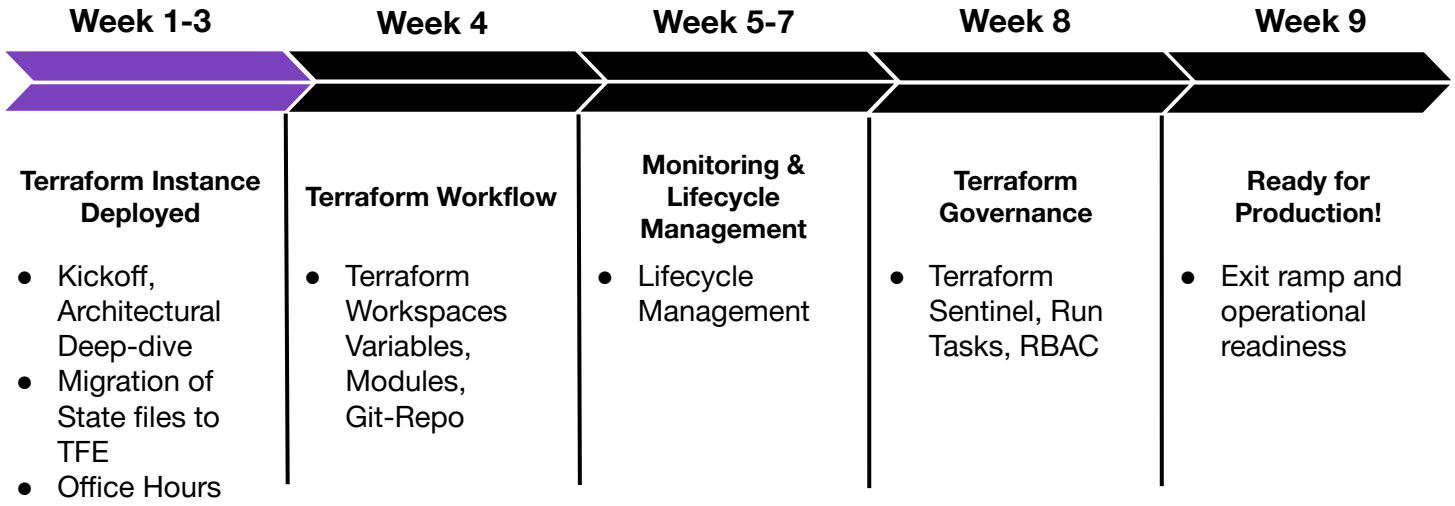
Stand Alone

April 2022

Agenda

- Architecture
- Deployment Patterns
- Configuration
- Next Steps
- Q & A

Terraform Enterprise Path to Production

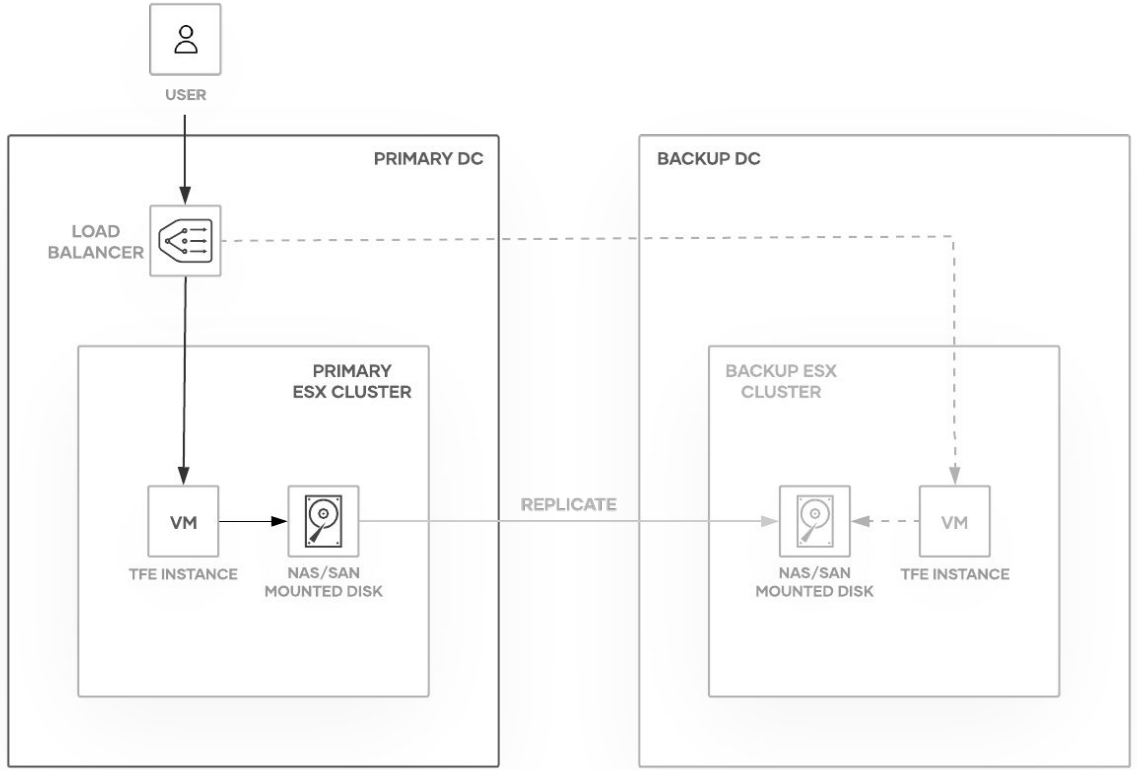


Reference Architecture

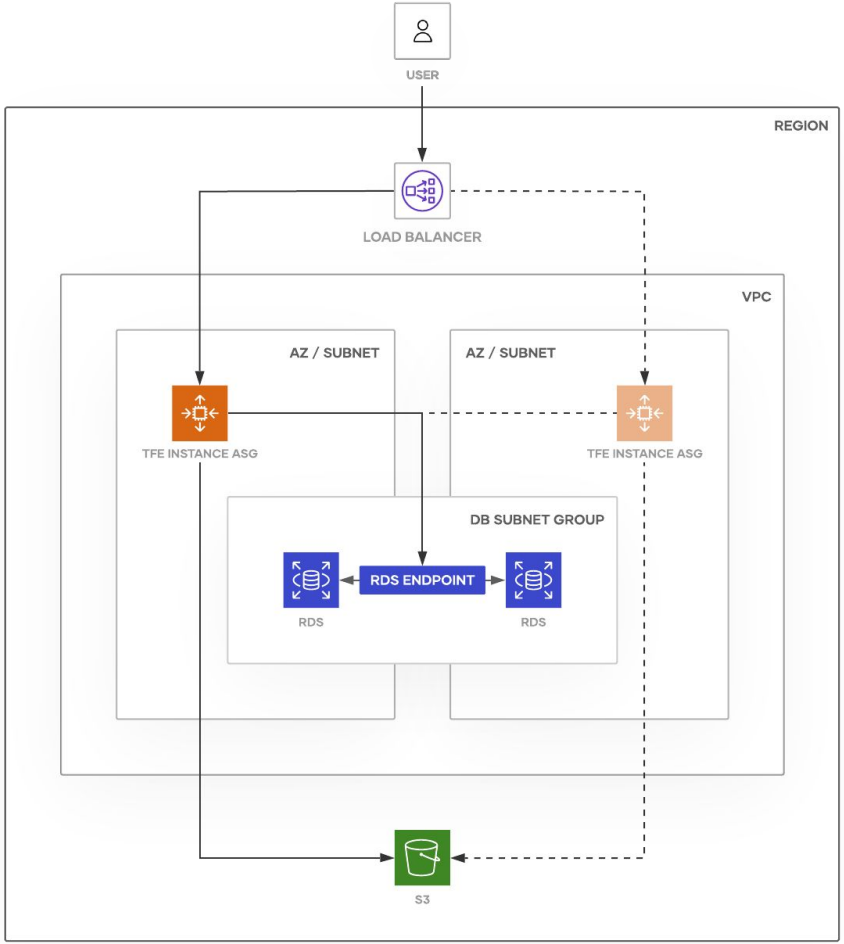
Architecture

- Terraform is an **Infrastructure as Code (IaC)** system that enables companies to define their cloud resources as **HashiCorp Configuration Language (HCL)**.
- HCL code can be stored in a Git repo to provide Version Control and **Auditing**. Git can also be used to **trigger automation**.
- **Terraform Enterprise (TFE)** is a self-managed service, unlike Terraform Cloud.
    - Includes remote runners called '**Cloud Agents**', that can be deployed both on-prem and across multiple cloud providers.
    - Uses **S3**-compatible storage, **Postgres** and **Replicated** for licensing.

VMware Standalone, Reference Architecture (Recommended)



Cloud Provider Standalone, Reference Architecture (Recommended)



Pre-Deploy Requirements

OS & Software Requirements

Operating System	
Distro	**Version**
Ubuntu	20.04, 18.04
Debian	11, 10, 9
CentOS	8.x, 7.x
Oracle Linux	8.x, 7.x
Red Hat Enterprise Linux	8.x, 7.x
Amazon Linux	2.0

Software	
Name	**Version**
Docker CE*	20.x 19.x
containerd.io	containerd 1.4.9 (2021-07-29) containerd 1.5.5 (2021-07-29) or newer
runc	runc v1.0.0-rc93 (2021-02-03) or newer
PostgreSQL	10.x, 11.x, 12.x

Note: Docker CE is not vendor supported on Red Hat Enterprise Linux and Oracle Linux
https://www.terraform.io/docs/enterprise/before-installing/rhel-requirements.html#install-requirements

VMware Recommended Sizing

TFE Server (vSphere and vRealize)					
Type	**CPU Sockets**	**CPU Cores**	**Memory**	**Storage**	**Storage (Mounted Disk Volume)**
Minimum	2	4	16 GB RAM	40GB	200GB
Scaled	2	8	32 GB RAM	40GB	200GB
PostgreSQL Database (External Services)					
Type	**CPU Sockets**	**CPU Cores**	**Memory**	**Storage**	
Minimum	2	2 cores	8 GB RAM	50GB	
Scaled	2	4-8 cores	16-32 GB RAM	50GB	

https://www.terraform.io/docs/enterprise/before-installing/reference-architecture/vmware.html

Cloud Recommended Sizing

TFE Server						
Type	**CPU**	**Memory**	**Storage**	**AWS**	**Azure**	**GCP**
Minimum	4 core	16 GB RAM	50GB	m5.xlarge	Standard_D4_v4	N1-standard-4 (200GB Storage)
Scaled	8 core	32 GB RAM	50GB	m5.2xlarge	Standard_D8_v4	N1-standard-8 (200GB Storage)
PostgreSQL Database						
Type	**CPU**	**Memory**	**Storage**	**AWS**	**Azure**	**GCP**
Minimum	4 core	16 GB RAM	50GB	db.m4.xlarge	GP_Gen5_4	Custom PostgreSQL Production
Scaled	8 core	32 GB RAM	50GB	db.m4.2xlarge	GP_Gen5_8	Custom PostgreSQL Production

https://www.terraform.io/docs/enterprise/before-installing/reference-architecture/aws.html
https://www.terraform.io/docs/enterprise/before-installing/reference-architecture/azure.html
https://www.terraform.io/docs/enterprise/before-installing/reference-architecture/gcp.html

Network Requirements (Egress)

Online Installations

- *.replicated.com
- *.quay.io
- quay-registry.s3.amazonaws.com
- *.cloudfront.net
- *.docker.com
- *.docker.io
- *.terraform.io
- releases.hashicorp.com

TFE also needs Egress access to:

- Any VCS servers/services that will be utilized
- Login/authentication servers if SAML will be configured (ADFS, Okta, etc)
- The various cloud API endpoints that will be managed with Terraform
- Any other third party services that will either be integrated with the TFE server or managed with it.

Cost Estimation APIs

When Cost Estimation is enabled, it uses the respective cloud provider's APIs to get up-to-date pricing info.

- api.pricing.us-east-1.amazonaws.com
- cloudbilling.googleapis.com
- prices.azure.com

https://www.terraform.io/docs/enterprise/before-installing/network-requirements.html

Network Requirements (Ingress)

Source - User/Client/VCS

- **80:** Terraform Enterprise application access (HTTP; redirects to HTTPS)
- **443:** Terraform Enterprise application access (HTTPS)
- VCS Web Hooks are inbound. If you intend to use VCS integration with a cloud hosted VCS, you'll need to expose TFE on the public internet.

Source - Administrators

- **22:** SSH access (administration and debugging)
- **8800:** Replicated (TFE setup dashboard, HTTPS)

https://www.terraform.io/docs/enterprise/before-installing/network-requirements.html

Network Requirements (TLS Certificates)

Trusted

Prior to installation determine the hostname and generate a certificate that will be used during installation. Using an IP to access TFE is not supported.

- If using on prem VCS, then the cert could be signed by your own root authority.
- If using SaaS VCS, then you'll need a publicly trusted cert.

Self-signed

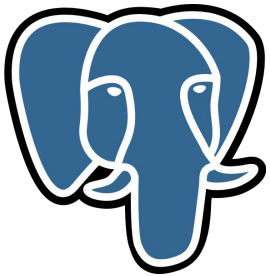
Don't do it! It will be more trouble in the long run.

Load Balancer / WAF: TLS Offload / Inspection

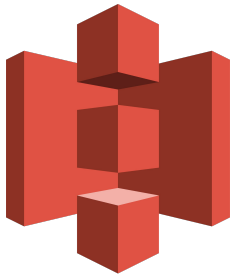
Ensure you re-encrypt the traffic and still use a trusted cert on the origin servers.

Components

External Services



Postgres



S3 Storage

Replicated

Replicated is an external vendor that HashiCorp uses to manage licensing of Terraform Enterprise, and as a scheduler to manage Docker containers.

Customers should not interact with Replicated directly unless directed by support.

https://www.terraform.io/docs/enterprise/system-overview/architecture.html

Terraform - Services

- **ptfe_nginx** - Nginx reverse proxy, facilitates access to the Terraform Enterprise services
- **ptfe_atlas** - The API and Web UI. Terraform Enterprise used to be known as Atlas
- **ptfe_build_manager** - Manages the queue of Terraform runs
- **ptfe_build_worker** - Creates workers on-demand as required by the queue. Injects variables, secrets, and Terraform configuration to a temporary container, ptfe_worker
- **ptfe_worker** - Executes a Terraform plan or apply. This container can be replaced with a custom image. This ephemeral container may be created with a randomly generated name by Docker

https://support.hashicorp.com/hc/en-us/articles/1500012954942-Terraform-Enterprise-Containers

Operational Modes

Operational Modes

- Demo Mode
- Mounted Disk **
- External Services
- Active/Active

** *Recommended Operational Mode*

Demo Mode

Not supported nor recommended for Production use

- Ephemeral
- Low Capacity
- Single Region
- Self-contained
- Easy to set up manually
- Good for Non-Production Local Testing
- Single Docker instance for Postgres, S3 Storage

https://www.terraform.io/docs/enterprise/system-overview/reliability-availability.html#operation-modes

Mounted Disk

Production supported
(Recommended for VMWare)

- Good for Production workloads
- Long-Lived
- Single Region
- Self-contained
- Easy to set up manually
- High Availability (HA) / Disaster Recovery (DR) with cold standby
- Single Docker instance for Postgres, S3 Storage

https://www.terraform.io/docs/enterprise/system-overview/reliability-availability.html#operation-modes

Standalone External Services

- Good for high Capacity Production Workloads
- Single Region
- Needs automation to set up quickly
- Uses externally running Postgres, S3 Storage
- High Availability (HA) / Disaster Recovery (DR) with cold standby

https://www.terraform.io/docs/enterprise/system-overview/reliability-availability.html#operation-modes

Active/Active

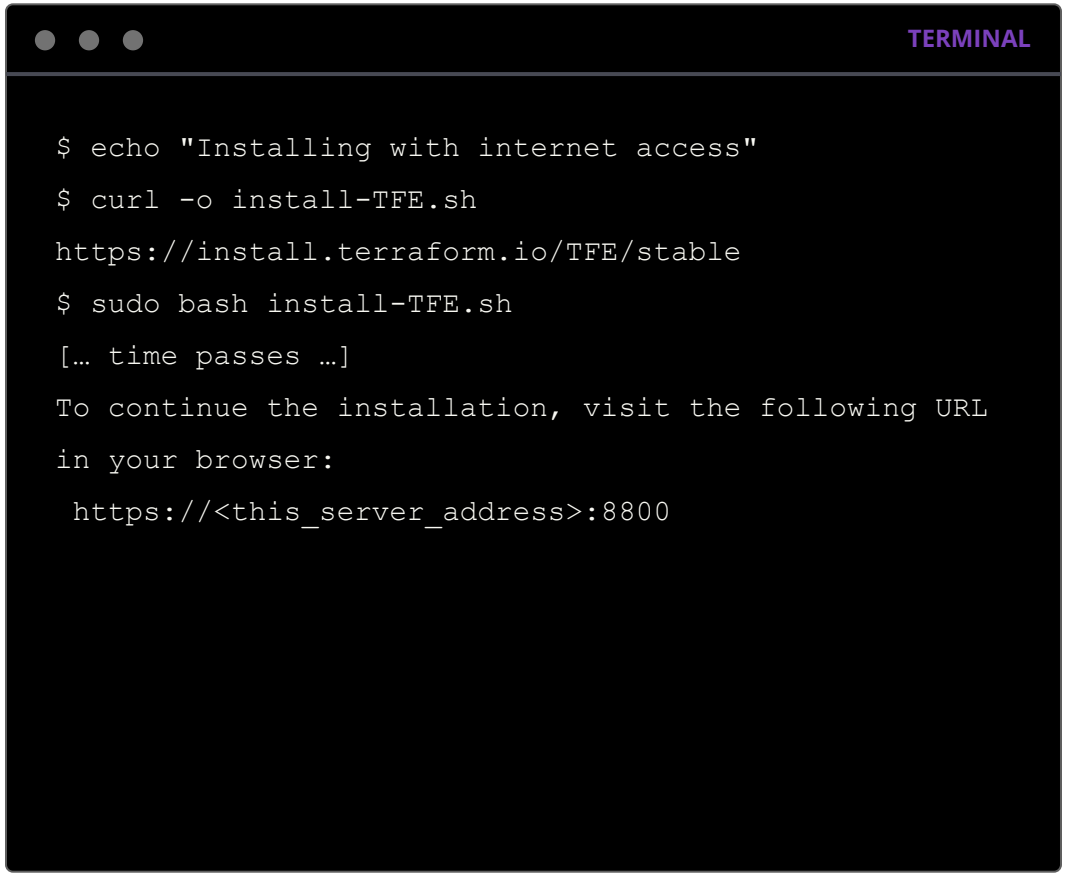
- High Availability (HA
- Disaster Recovery (DR)
- Single Region
- Needs automation to set up quickly, with twice the hardware
- Good for Production Workloads with fast MTTR, RTO, RPO
- Uses externally running Postgres, S3 Storage, and Redis

https://www.terraform.io/docs/enterprise/system-overview/reliability-availability.html#operation-modes

Deployment Patterns

Deployment Patterns

- Online
- Airgapped



Online Installation

Airgapped Installation

Running Terraform Enterprise within an Airgapped Environment includes a variety of manual configuration and management processes.

TERMINAL

```
$ echo "Installing with Airgap file"
$ apt install docker -y
$ wget
https://install.terraform.io/airgap/latest.tar.gz
$ echo "SCP the .airgap file onto the machine"
$ tar xzf replicated.tar.gz
$ sudo ./install.sh airgap
[… time passes …]
To continue the installation, visit the following URL
in your browser:
 https://<this_server_address>:8800
```

Automated Installation

Replicated Settings
The initial replicated configuration of the dashboard password, TLS certificates, license file, and application settings can be defined in /etc/replicated.conf. Replicated automatically checks this location during execution of the installer.

Application Settings
The configuration of TFE can be completed by passing a JSON formatted settings file during the installation. This is supported for both online and airgapped installations.

Initial User Creation
After installation an initial admin user must be created to use TFE. This is normally created in the UI however we can leverage the API to create this user.

Automated Installation

Replicated Settings

TERMINAL

```
{
    "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/application-settings.json",
    "LicenseFileLocation":          "/path/to/license.rli"
}
```

CODE EDITOR

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

Automated Installation

Application Settings

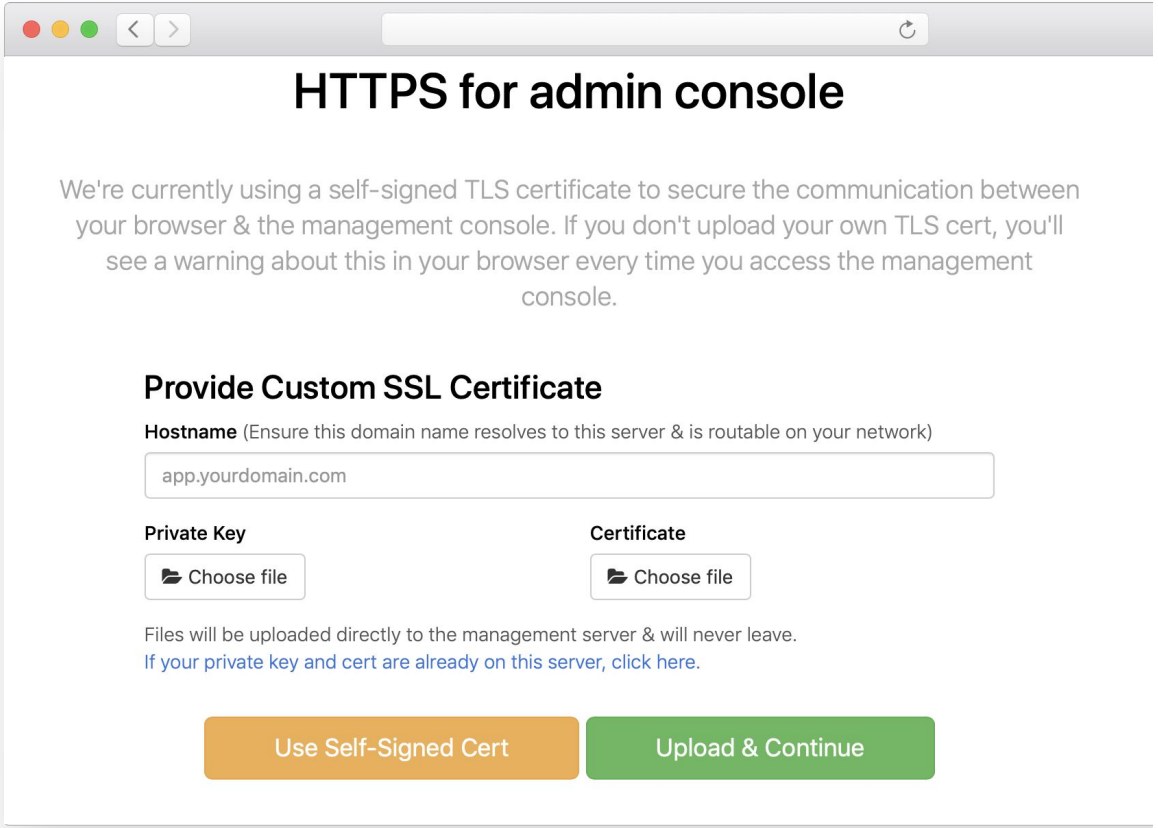
Automated Installation

Initial admin user creation

TERMINAL

```
$ replicated admin --no-tty retrieve-iact > iact.txt
$ curl payload.json
{
  "username": "admin",
  "email": "it@mycompany.com",
  "password": "thisisabadpassword"
}
$ curl --header "Content-Type: application/json" \
  --request POST --data @payload.json \
  https://TFE.company.com/admin/initial-admin-user?token=$(cat
iact.txt)
{
  "status": "created",
  "token":
"aabbccdd.v1.atlas.ddeeffgghhiijjkkllmmnnooppqqrrssttuuvvxxyyzz"
}
```

Configuration

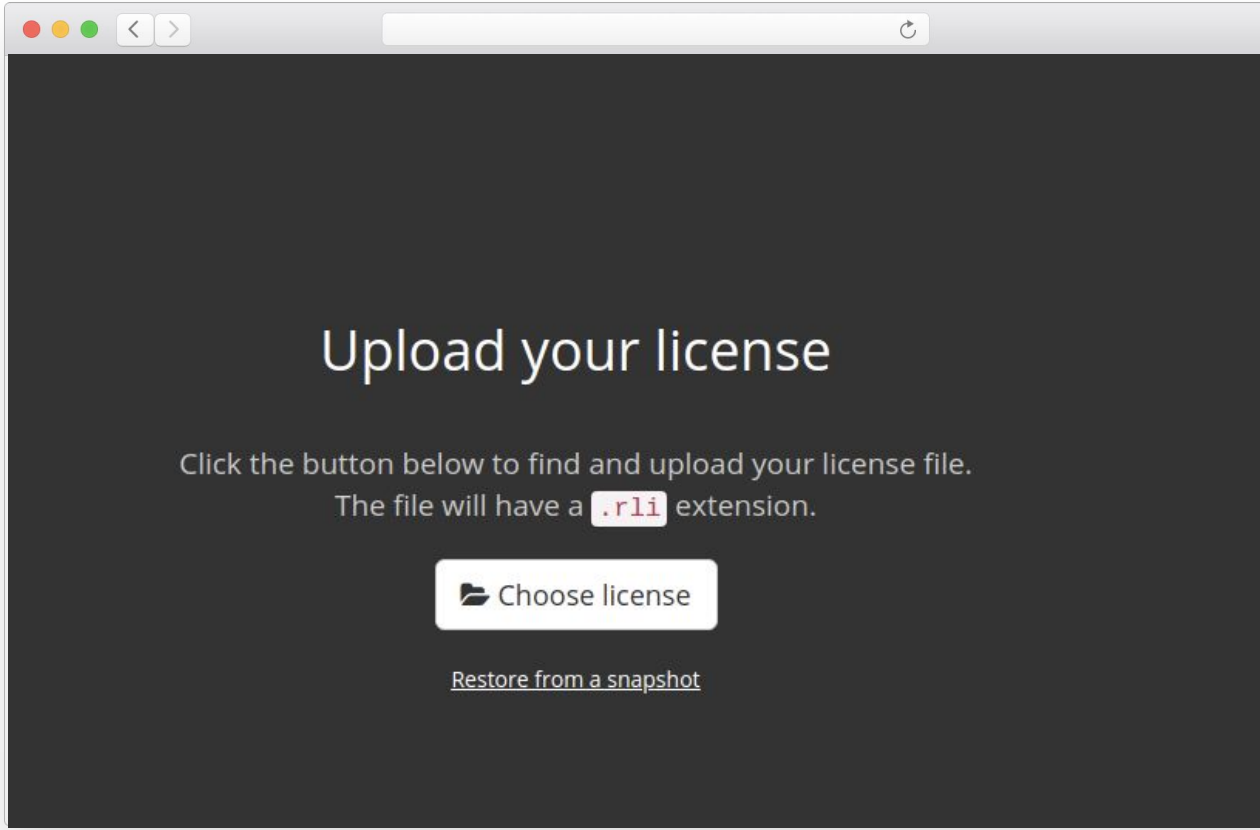


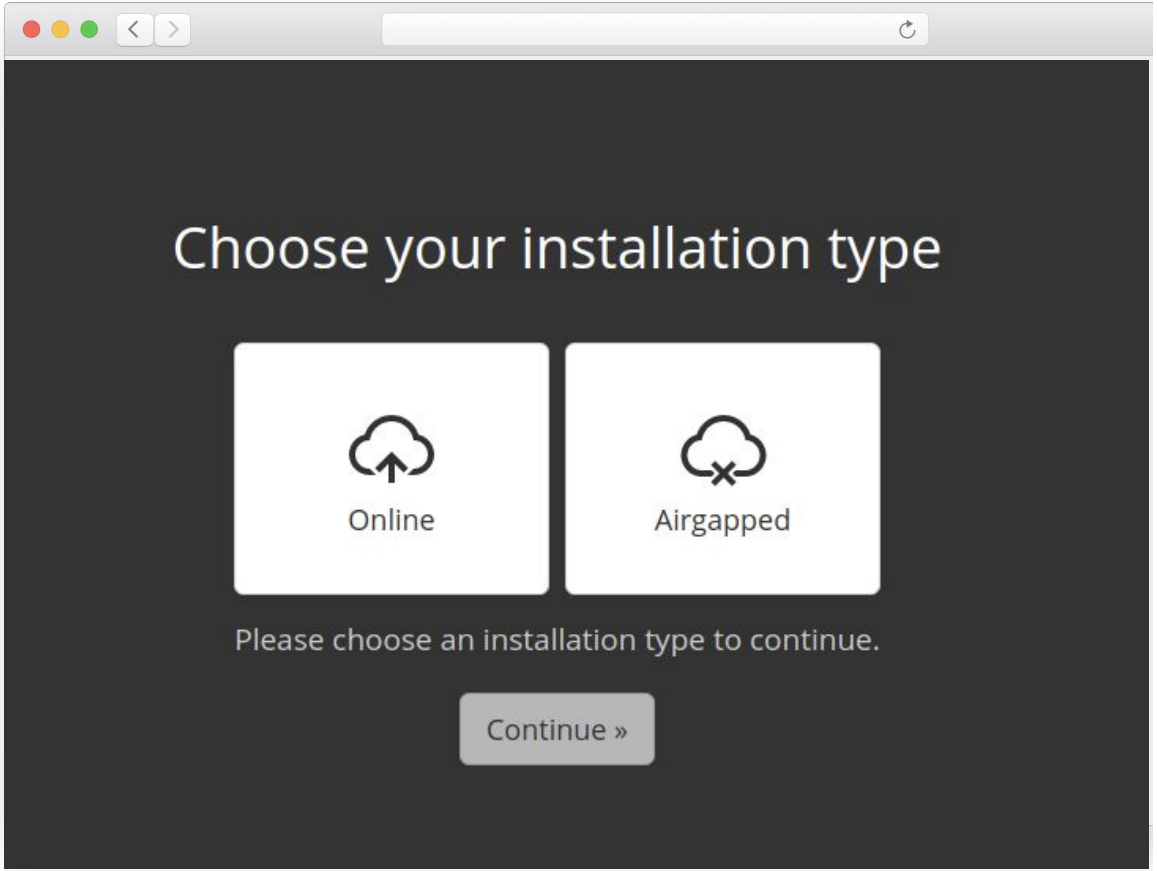
HTTPS Config

Enter hostname and upload private key and certificate

TFE Licensing

Upload .rli file provided in your welcome email



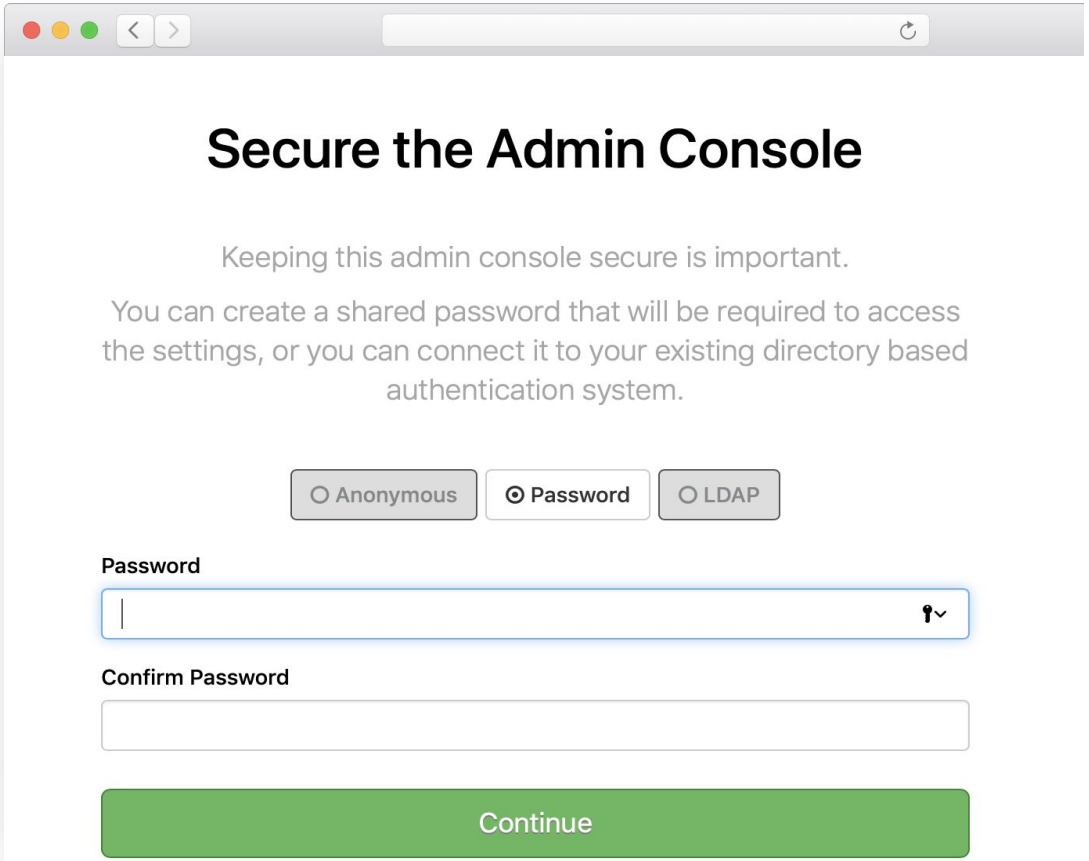


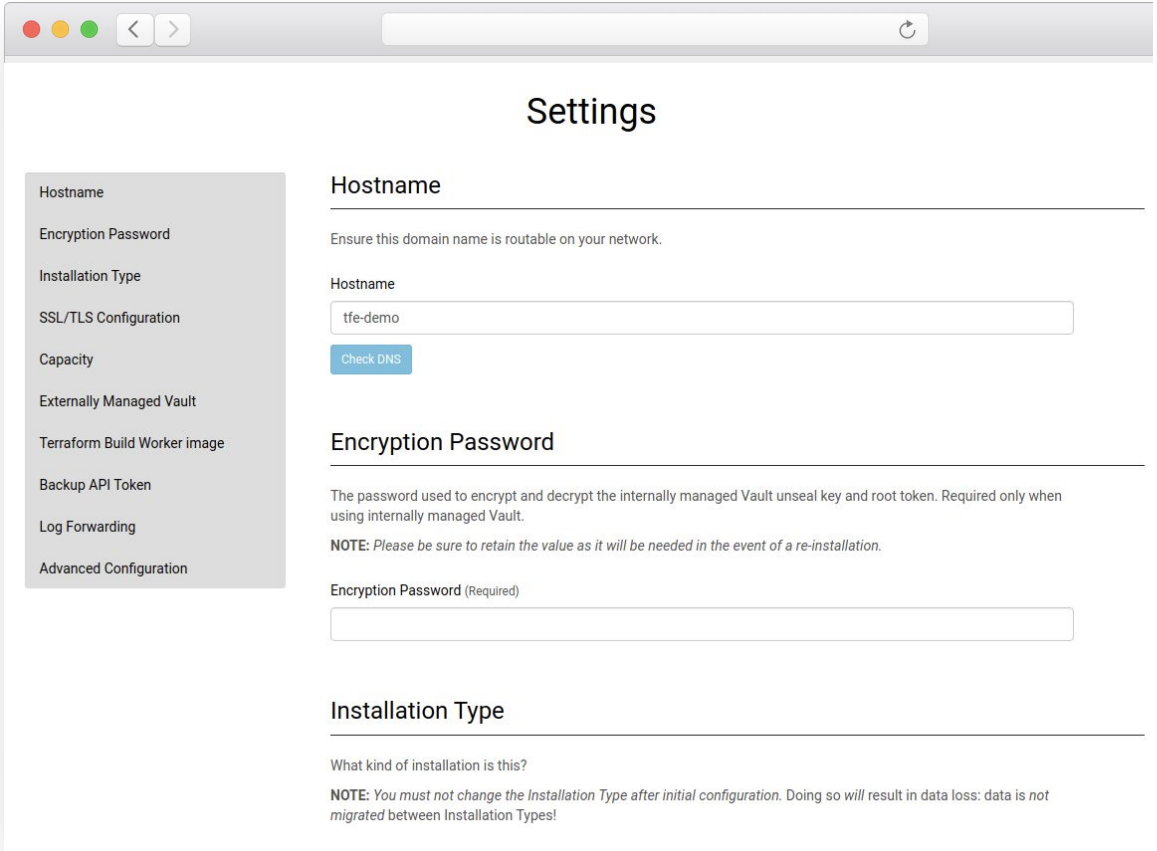
Select Installation Type

Select online or airgapped and the preflight check will begin

Admin Console Auth

Enter password or configure LDAP. Anonymous is not recommended for use in production deployment



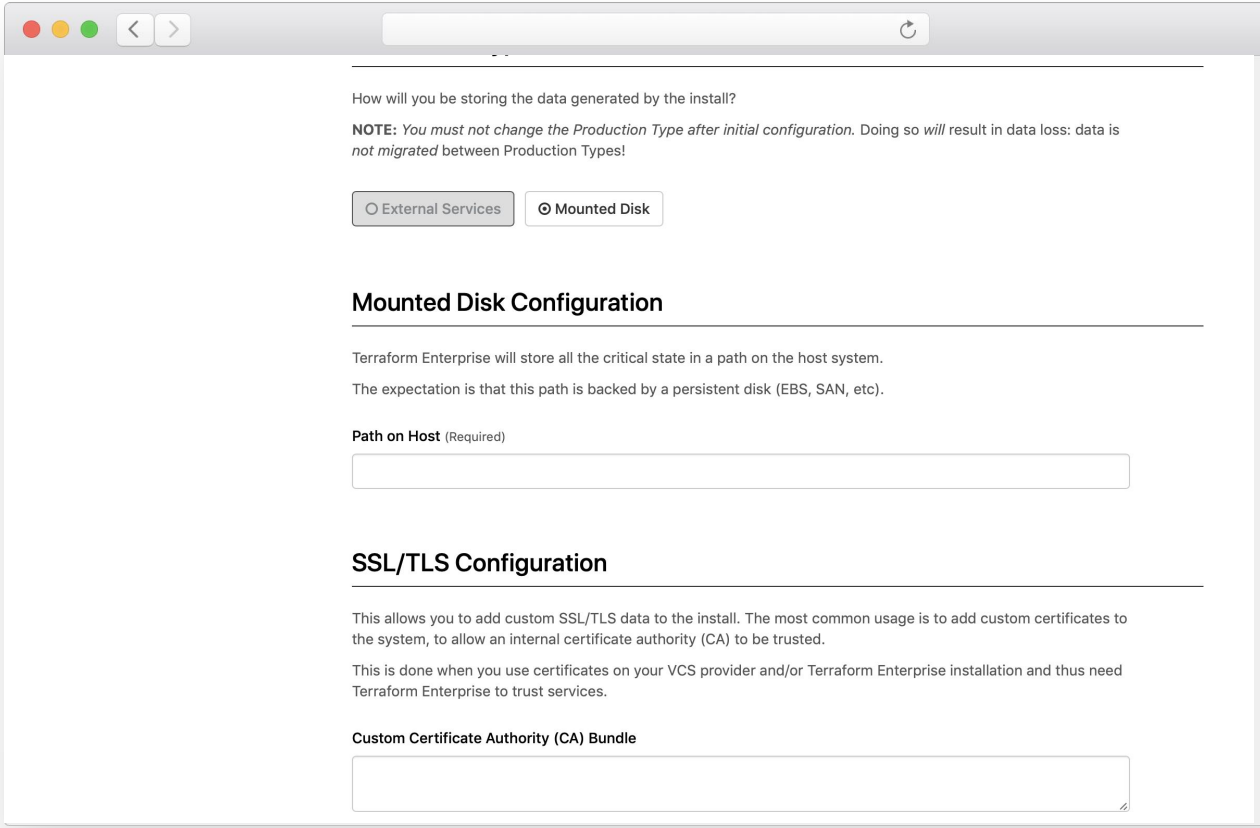


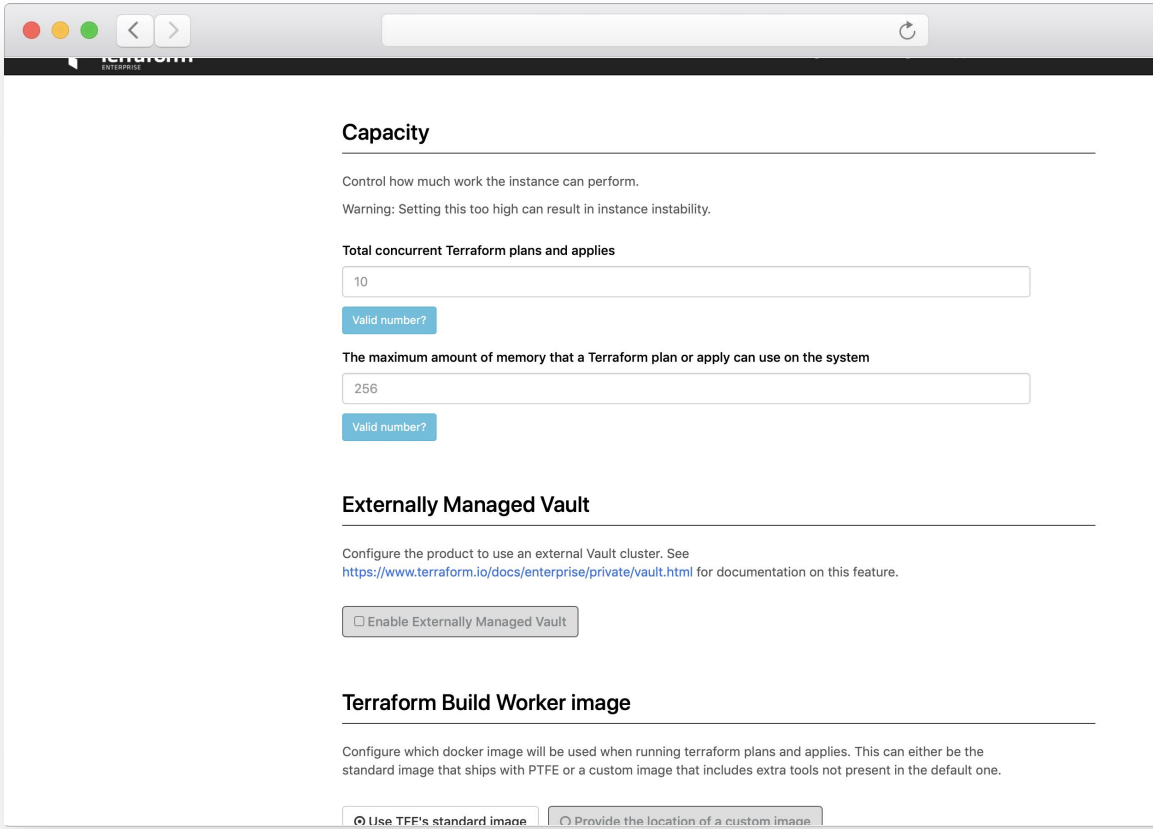
Installer Settings

Specify installation type and operation mode.

Installer Settings

Specify installation type and operation mode.



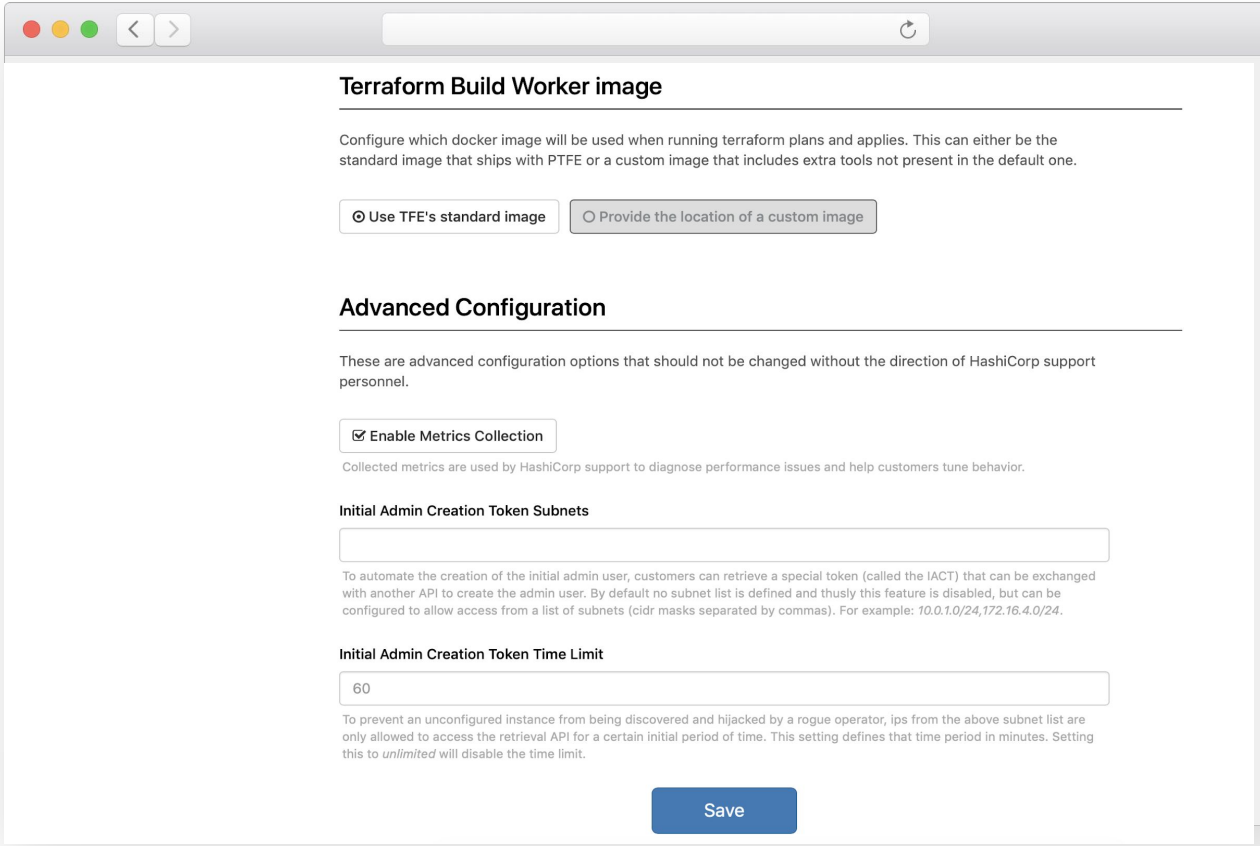


Installer Settings

Specify installation type and operation mode.

Installer Settings

Click save and the installer will configure deploy the TFE docker containers and start the app.

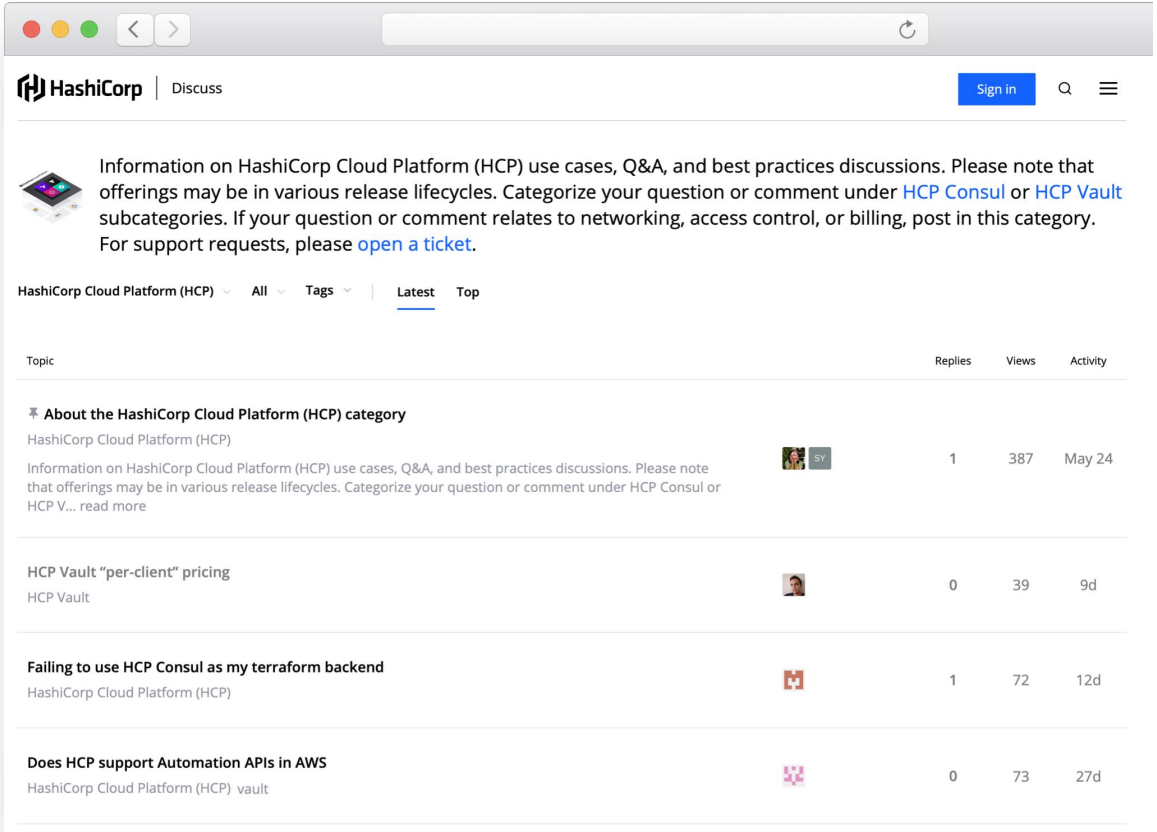


Resources

Resources

- Reference Architecture
  - AWS / Azure / GCP
  - VMware
- Pre-deploy Requirements
  - OS Requirements
  - Software Requirements
  - Network Requirements
  - Certificates
- Components
  - Services & Data Flow Diagram
  - Terraform Enterprise Containers
- Operational Modes
- Deployment Patterns
  - Online
  - Airgapped
    - Interactive Installation

Next Steps



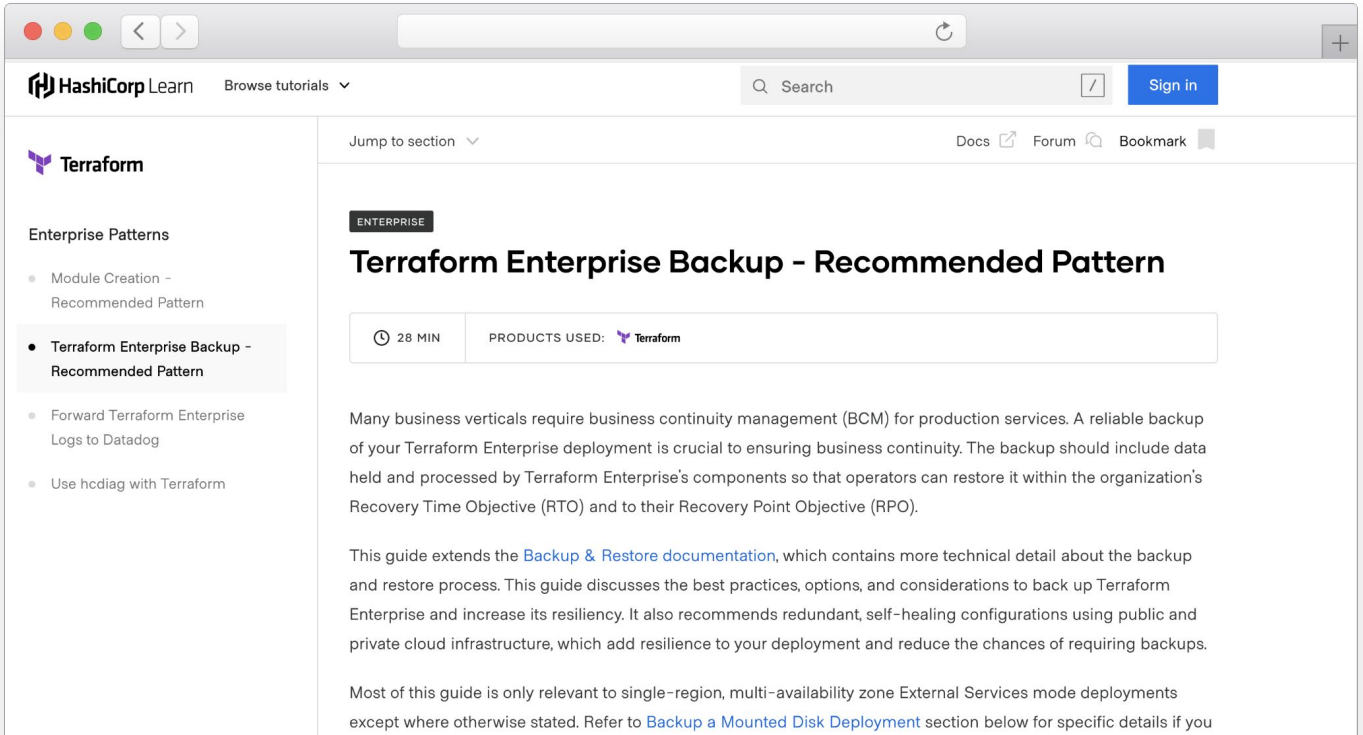
Discuss

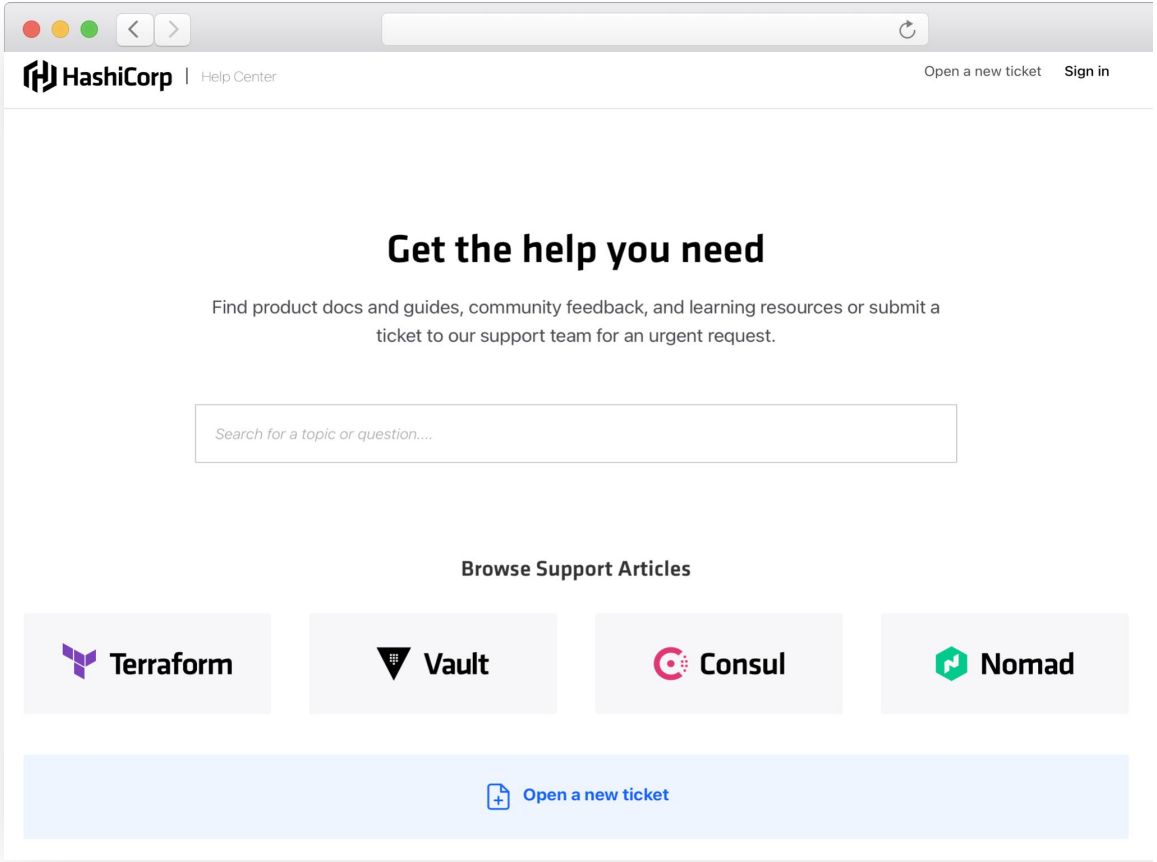
Engage with the HashiCorp Cloud community including HashiCorp Architects and Engineers.

discuss.hashicorp.com

Learn

Step-by-step guides to accelerate deployment of Terraform





Support

https://support.hashicorp.com

Need Additional Help?

Customer Success

Contact our Customer Success Management team with any questions. We will help coordinate the right resources for you to get your questions answered.

customer.success@hashicorp.com

Technical Support

Something not working quite right? Engage with HashiCorp Technical Support by opening a new ticket for your issue at support.hashicorp.com.

Q & A

Upcoming Onboarding Webinars

May 5, 2022

Importing Resources and State into Terraform Enterprise

Learn Best Practices for importing resources into Terraform State and moving Terraform OSS into Enterprise.

May 10, 2022

Community Office Hours

An interactive open forum to discuss specific questions about your environment and Use Cases. Please bring your questions.

May 17, 2022

Lifecycle Management

Learn Best Practices for monitoring, upgrades, and backups in Terraform Enterprise.

Thank You

hello@hashicorp.com
www.hashicorp.com