HashiCorp
Terraform

Terraform Cloud for Business (TFCB) Getting Started

April 27, 2022

Agenda

- TFCB Overview
- Getting Started with TFCB
- Demo
- Next Steps
- Q & A

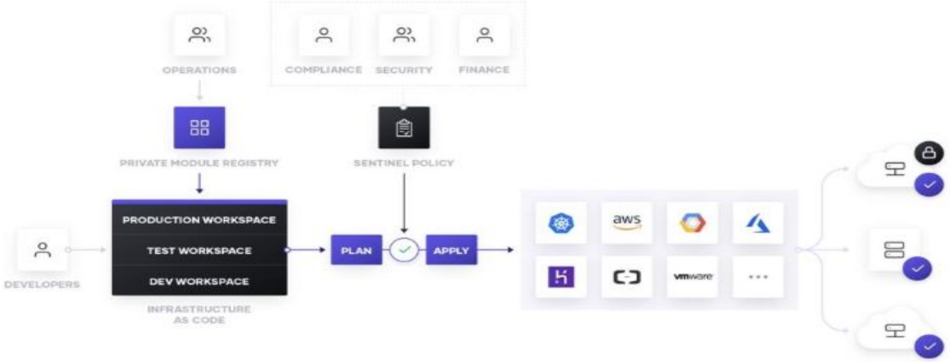
Terraform Cloud for Business Overview

Terraform Cloud for Business Overview

- Overview
- Organizations
- SSO, Teams, Users
- Private Module Registry
- Workspaces
- Run Types
- Cloud Agents

Terraform Cloud for Business

- Central platform running in the cloud, that HashiCorp manages for you.
- Provides common Workflows for Users across Teams and Clouds (UI/VCS/API/CLI)
- Has an API, SSO, Teams, Users, Tokens, RBAC, VCS Connections, Private Module Registry, Workspaces, State Management, Variables, Cost Estimation, Run Triggers, Run Tasks, Run Notifications, Policy as Code with Sentinel, and Cost Estimation.
- Allows you to establish a Producer/Consumer model, for a separation of duties.
- Provides a Policy as Code Framework via Sentinel, to ensure governance across workspaces.



Organizations

Overview

Organizations is a security boundary. TFCB can have multiple Organizations. During new account setup an initial Organization is created, and additional Organizations can be created by Site Admins.

Users can belong to multiple Organizations and can use the selector in the UI to choose which Organization to operate in. If users are added to an Organization Team, they are added to the Organization automatically.

Organizations Components

- SSO Settings
- Teams
- Users
- API Tokens (Org, Teams, Users)
- VCS Provider / Git Connections
- Private Module Registry
- Workspaces (TF Code + Statefile)
- Variables, ENV Variables, CLI Flags
- SSH Keys
- Sentinel Policy Sets
- Cloud Agents

Teams

Overview

Teams are groups of users within an organization that can be assigned to workspaces within the organization. Teams can be assigned to multiple workspaces and have different permissions in each workspace. Workspace-level permissions include: Read, Plan, Write, and Admin.

Teams + Organizations

Teams can also be assigned organization-level permissions including: Managing Policies, Manage Workspaces, and Manage VCS Settings. Each organization contains an Owners team which has the above permissions.

Users

Overview

Users in TFCB are members of Teams within Organizations. When TFCB is not configured with an identity provider, users can self-register.

Users do not belong to any organization or workspaces until an owner of them has added them to a team.

User Settings

Users can control these account level settings:

- Username
- Email
- Avatar
- Password
- Two Factor Authentication
- Multiple User API Tokens

Authentication Methods

Username/Password

The default authentication is username / password. This mode allows users to self register. They will need to provide an email address and password.

SAML SSO

TFCB includes integrations with Azure AD and Okta for single sign on. TFCB can also integrate with your SAML capable identity provider.

API Tokens

Once a user has logged into TFCB they will be able to generate an API token. API tokens are necessary for:

- Auth with TFCB API
- Auth with TF remote backend for CLI runs
- Using private modules in command-line runs on local machine

SAML SSO

TFCB supports integrating with SAML 2.0 compliant identity solutions. When TFCB is configured for SAML, the login prompt will change to redirect users to the IDP to complete the login and then will be redirected back to TFCB.
Team membership mapping can be enabled to have user added to teams based on an attribute in the SAML assertion.

Identity Provider Guides
Azure Active Directory
Okta
SAML

Service Accounts

Team Service Accounts

Designed to perform API operations on workspaces. The API token will have same access and permissions as their team. This token is generated in the team page and can be used interactively.

Organization Service Accounts

Designed to create and configure workspaces and teams. Not recommended to be used for all-purpose interface to TFCB. Should be used for initial setup and then delegate a workspace to a team. The team service account should then be used.

Workspaces

Overview

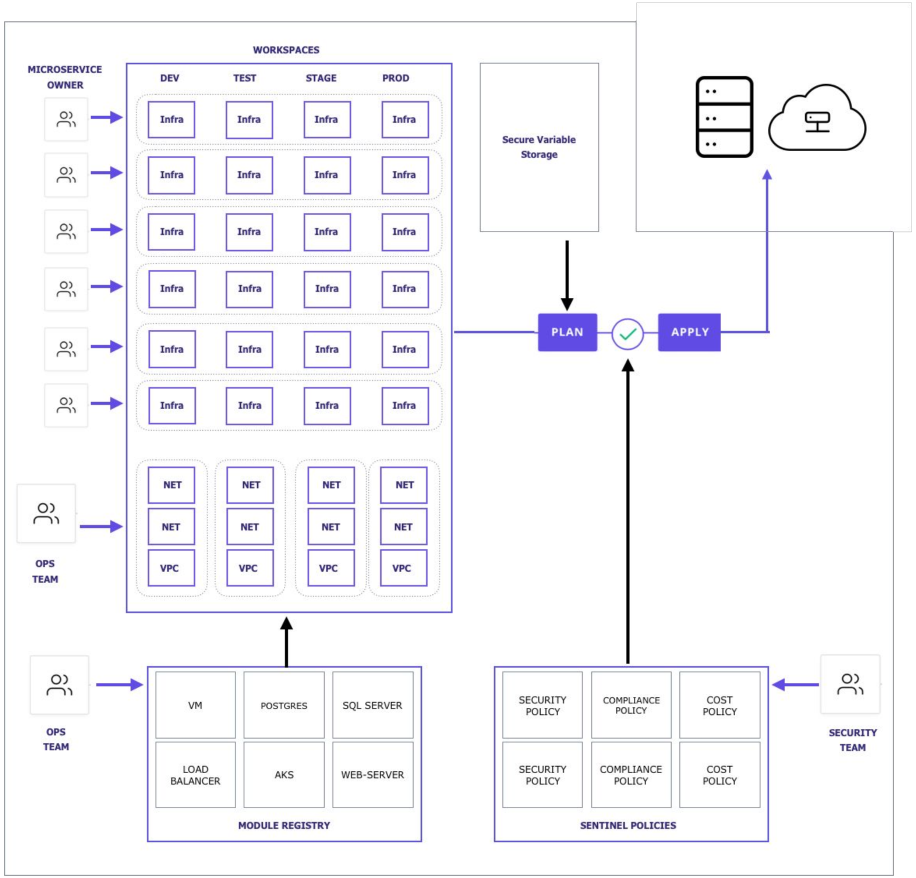
TFCB arranges infrastructure by Workspaces within Organizations. Workspaces can be run by uploading a .zip file of TF code to the API, or be connected to a Git Repository from your VCS provider and will monitor for changes using Git Webhooks. Variables for TF Input Variables, Sensitive TF Input Variables, and Environment Variables, can be managed in the Workspace.

Workspaces Contain:

- Terraform Code, from a VCS Git Repo or uploaded as a .zip file to the API
- Variables, which can be Marked as Sensitive
- Environment Variables
- Persistently stored TF Statefiles for cloud resources that are managed
- Historical TF Statefiles and Run logs

Workspaces

- Organize and decompose monolithic infrastructure into micro-infrastructures.
- Match the organization of your application or teams with your infrastructure.
- “Micro-infrastructures” are linked to create the complete infrastructure for the application.



UI/VCS-Driven Runs

Workflow

UI and VCS workflows are the primary mode of operation in TFCB. In this configuration each Workspace is connected with a specific Git Branch in a Git Repo on your VCS Provider. TFCB registers Git Webhooks with your VCS Provider during Workspace creation. As new Git Commits are Merged into a Git Branch, TFCB will automatically queue a Workspace Run.

Auto-apply

By default, runs require confirmation before Terraform Cloud for Business will apply them. If you would prefer to auto apply plans that do not contain errors, you can enable auto apply in the workspaces “General Settings” page.

CLI-Driven Runs

Remote Backend

Terraforms remote backend enables developers who are already familiar with Terraform CLI workflow to integrate with Terraform Cloud for Business. Once integrated with the remote backend, runs will execute remotely in Terraform Cloud for Business while displaying progress in the terminal where the run is executed.

Terraform CLI Tool

The Terraform CLI tool provides a CLI interface that leverages Terraform Cloud for Business API. This tool can be useful for modifying variables and workspace settings from the terminal.

API-Driven Runs

Workflow

API-driven workflow provide a flexible workflow but require you to build tooling to determine when configuration has changed and a run should occur. In this workflow TFCB workspaces are not directly associated with a VCS repo and runs are not triggered by webhooks on your VCS provider.

Unsupported VCS Integration

This allows you to work with configurations from unsupported version control systems, automatically generate Terraform configurations from some other source of data, or build a variety of other integrations.
Example:
https://www.terraform.io/docs/enterprise/run/api.html#pushing-a-new-configuration-version

Terraform Cloud Agents

Terraform Cloud Agents allow Terraform Cloud to communicate with isolated, private, or on-premises infrastructure. By deploying lightweight Docker-based agents within a specific network segment, you can establish a simple connection between your environment and Terraform Cloud which allows for provisioning operations and management. This is useful for on-premises infrastructure types such as vSphere, Nutanix, OpenStack, enterprise networking providers, and anything you might have in a protected enclave.

The agent architecture is pull-based, so no inbound public internet connectivity is required. Any agent you provision will poll Terraform Cloud for work and carry out execution of that work locally.

https://www.terraform.io/docs/cloud/agents/index.html

VCS Integration

TFCB is most powerful when integrated with a VCS provider. TFCB registers Git Webhooks with your VCS Git Provider to monitor for new Git Commits and Git Pull Requests.

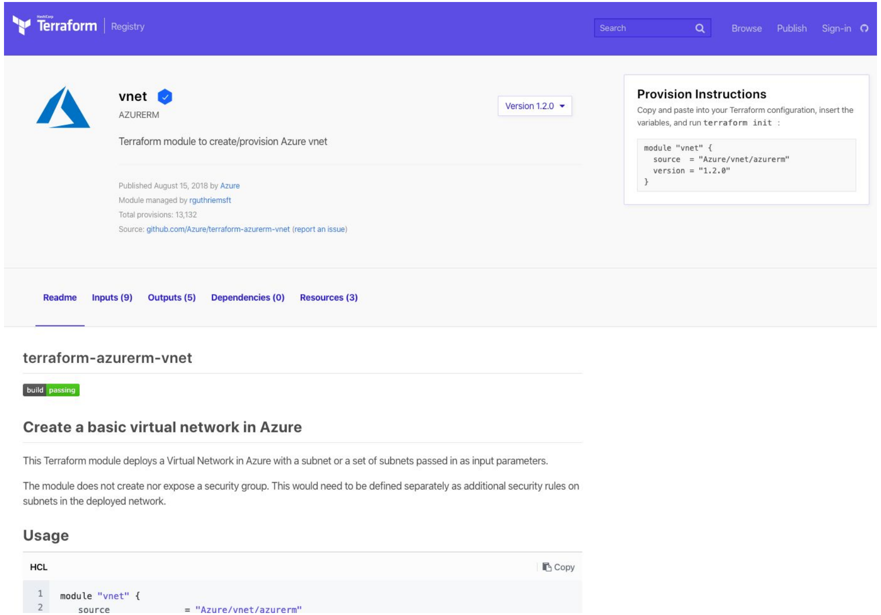
TFCB will interact with most providers using the providers API and OAuth token.
BitBucket Server does require an SSH key for downloading repo contents.
TFCB supports integrating with multiple VCS providers within an Organization.
During workspace creation you will select a configured provider.

Supported VCS Providers
GitHub
GitHub Enterprise
GitLab.com
GitLab EE and CE
BitBucket Cloud
BitBucket Server
Azure DevOps

Private Module Registry

A module is a container for multiple resources that are used together. Modules can be used to create lightweight abstractions, so that you can describe your infrastructure in terms of its architecture, rather than directly in terms of physical objects.

We will cover in more detail in a later webinar. Please check with your HashiCorp CSM for the registration details.



vnet

AZURERM

Terraform module to create/provision Azure vnet

Published August 15, 2018 by Azure

Module managed by rguthriemsft

Total provisions: 13,132

Source: github.com/Azure/terraform-azurerm-vnet (report an issue)

Provision Instructions

Copy and paste into your Terraform configuration, insert the variables, and run `terraform init` :

```
module "vnet" {
  source  = "Azure/vnet/azurerm"
  version = "1.2.0"
}
```

terraform-azurerm-vnet

Create a basic virtual network in Azure

This Terraform module deploys a Virtual Network in Azure with a subnet or a set of subnets passed in as input parameters.

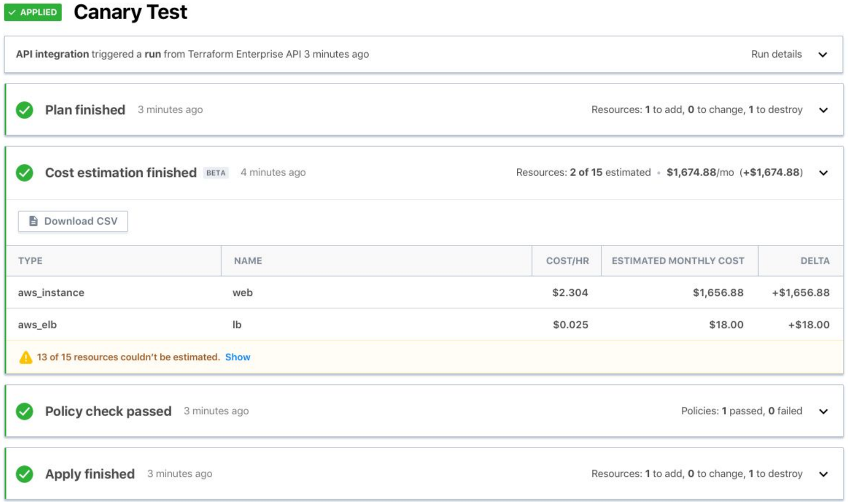
The module does not create nor expose a security group. This would need to be defined separately as additional security rules on subnets in the deployed network.

Usage

```
module "vnet" {
    source              = "Azure/vnet/azurerm"
```

Cost Estimation

Terraform Cloud provides cost estimates for many resources found in your Terraform configuration. For each resource an hourly and monthly cost is shown, along with the monthly delta. The total cost and delta of all estimable resources is also shown.



✓ APPLIED **Canary Test**

API integration triggered a **run** from Terraform Enterprise API 3 minutes ago — Run details

✓ **Plan finished** 3 minutes ago — Resources: **1** to add, **0** to change, **1** to destroy

✓ **Cost estimation finished** BETA 4 minutes ago — Resources: **2 of 15** estimated • **$1,674.88**/mo (**+$1,674.88**)

Download CSV

TYPE	NAME	COST/HR	ESTIMATED MONTHLY COST	DELTA
aws_instance	web	$2.304	$1,656.88	+$1,656.88
aws_elb	lb	$0.025	$18.00	+$18.00

⚠ 13 of 15 resources couldn't be estimated. Show

✓ **Policy check passed** 3 minutes ago — Policies: **1** passed, **0** failed

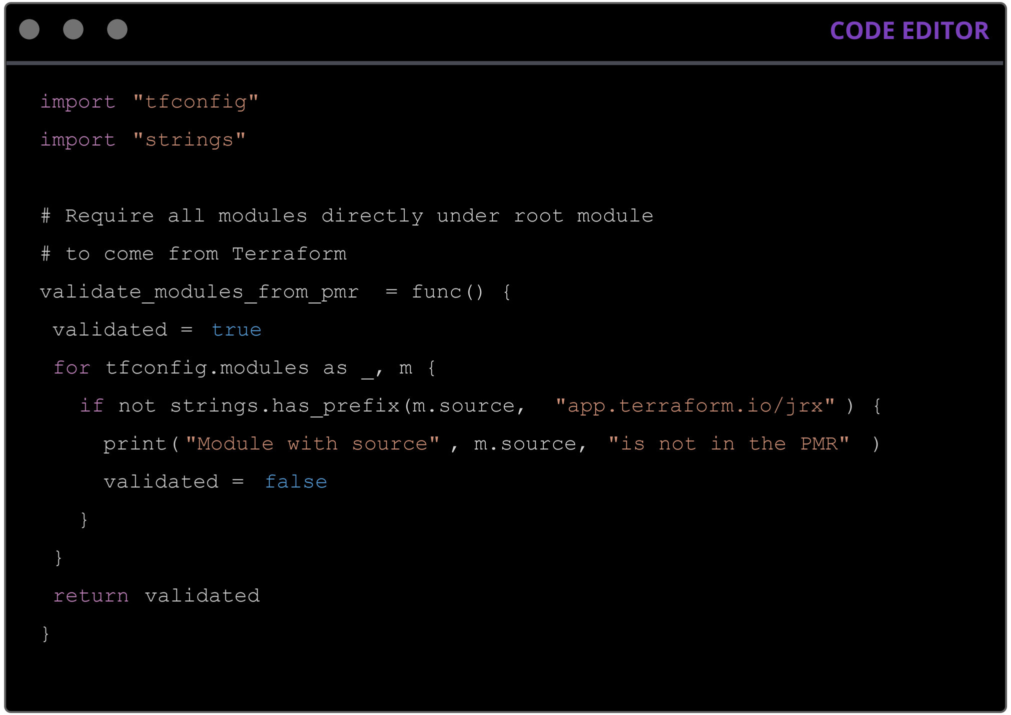
✓ **Apply finished** 3 minutes ago — Resources: **1** to add, **0** to change, **1** to destroy

Sentinel

Sentinel is a framework for Policies as Code (PaC) similar to how Terraform implements Infrastructure as Code (Iac).

- Sandboxing
- Codification
- Version Control
- Automation
- Testing

We will cover in more detail in a later webinar. Please check with your HashiCorp CSM for the registration details.



```
import "tfconfig"
import "strings"

# Require all modules directly under root module
# to come from Terraform
validate_modules_from_pmr  = func() {
 validated = true
 for tfconfig.modules as _, m {
   if not strings.has_prefix(m.source,  "app.terraform.io/jrx" ) {
     print( "Module with source" , m.source,  "is not in the PMR"  )
     validated =  false
   }
 }
 return validated
}
```

TF OSS to TFCB Migration

If you already use Terraform to manage infrastructure, you're probably managing some resources that you want to transfer to TFCB. By migrating your Terraform state to Terraform Cloud, you can continue managing that infrastructure without de-provisioning anything.

https://www.terraform.io/docs/cloud/migrate/index.html

Getting Started with TFCB

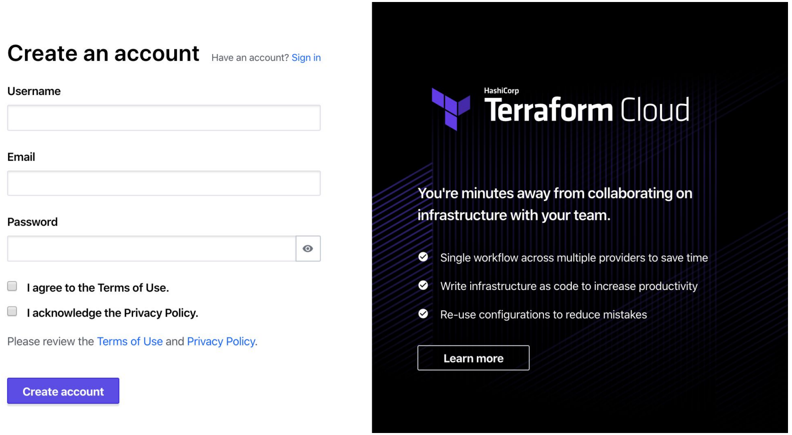
Getting Started with TFCB

- Creating an Account
- Create an Organization
- Create a VCS Connection
- Create a Workspace
- VCS-Driven Run

Create an Account

To get started, you will first need to create an account. This can be completed by navigating to https://app.terraform.io in your web browser and clicking on Create Account.

You will provide a username, email, and password to setup your account. Once you have created your account you can setup single sign-on for your organization to enable other users to access your organization.



Create an account

Have an account? Sign in

Username

Email

Password

□ I agree to the Terms of Use.

□ I acknowledge the Privacy Policy.

Please review the Terms of Use and Privacy Policy.

Create account

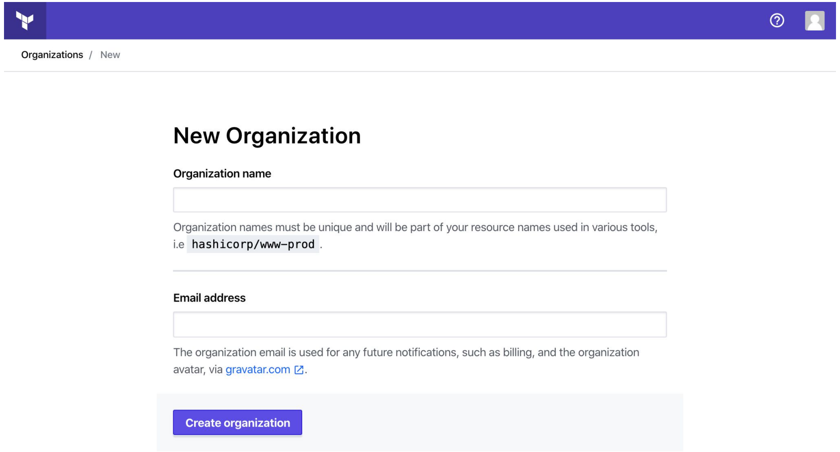
HashiCorp Terraform Cloud

You're minutes away from collaborating on infrastructure with your team.

- ✓ Single workflow across multiple providers to save time
- ✓ Write infrastructure as code to increase productivity
- ✓ Re-use configurations to reduce mistakes

Learn more

Create Organization



Organizations / New

New Organization

Organization name

Organization names must be unique and will be part of your resource names used in various tools, i.e `hashicorp/www-prod`.

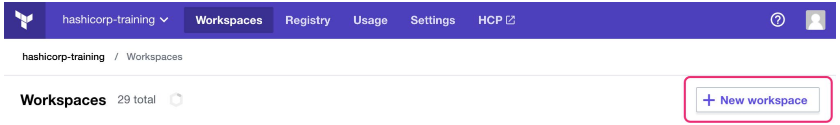
Email address

The organization email is used for any future notifications, such as billing, and the organization avatar, via gravatar.com.

The next screen will prompt you to enter an organization name and email address. Once entered, click create organization to proceed.

You will then need to contact your HashiCorp CSM to have your Terraform Cloud for Business entitlements enabled on this organization.

Create Workspace

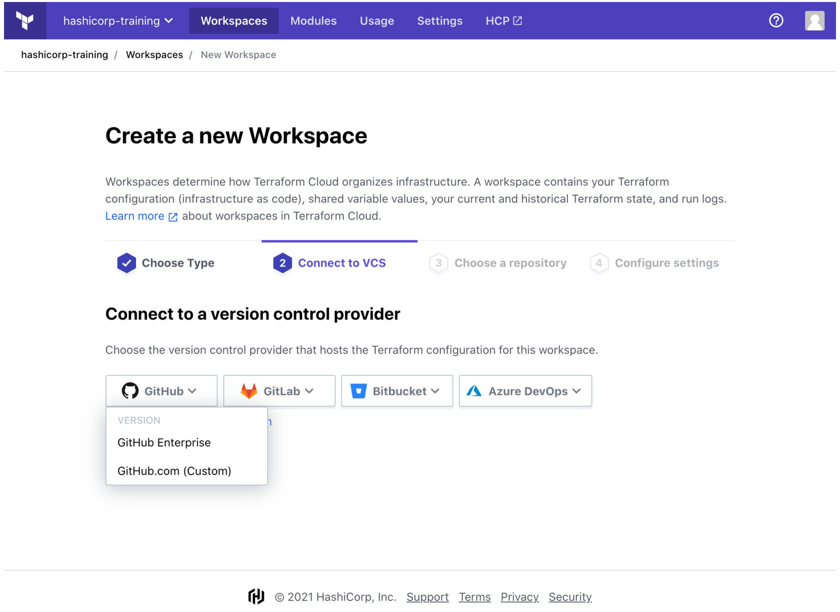


Next you will want to create a workspace to begin provisioning your infrastructure.

Select VCS backed workspace and provide the workspace name.

Add VCS Connection

TFCB can connect directly with your version control system (VCS) to access your configuration files and provision infrastructure. The first step is to make a connection to the VCS so we can connect and monitor repositories for changes.



Create a new Workspace

Workspaces determine how Terraform Cloud organizes infrastructure. A workspace contains your Terraform configuration (infrastructure as code), shared variable values, your current and historical Terraform state, and run logs. Learn more about workspaces in Terraform Cloud.

Connect to a version control provider

Choose the version control provider that hosts the Terraform configuration for this workspace.

Demo

Demo

- Access TFCB
- Create an organization
- Connect to a VCS
- Create a Workspace
- Deploy Infrastructure
- Modifying Infrastructure
- Destroying Infrastructure

Next Steps

Upcoming Onboarding Webinars

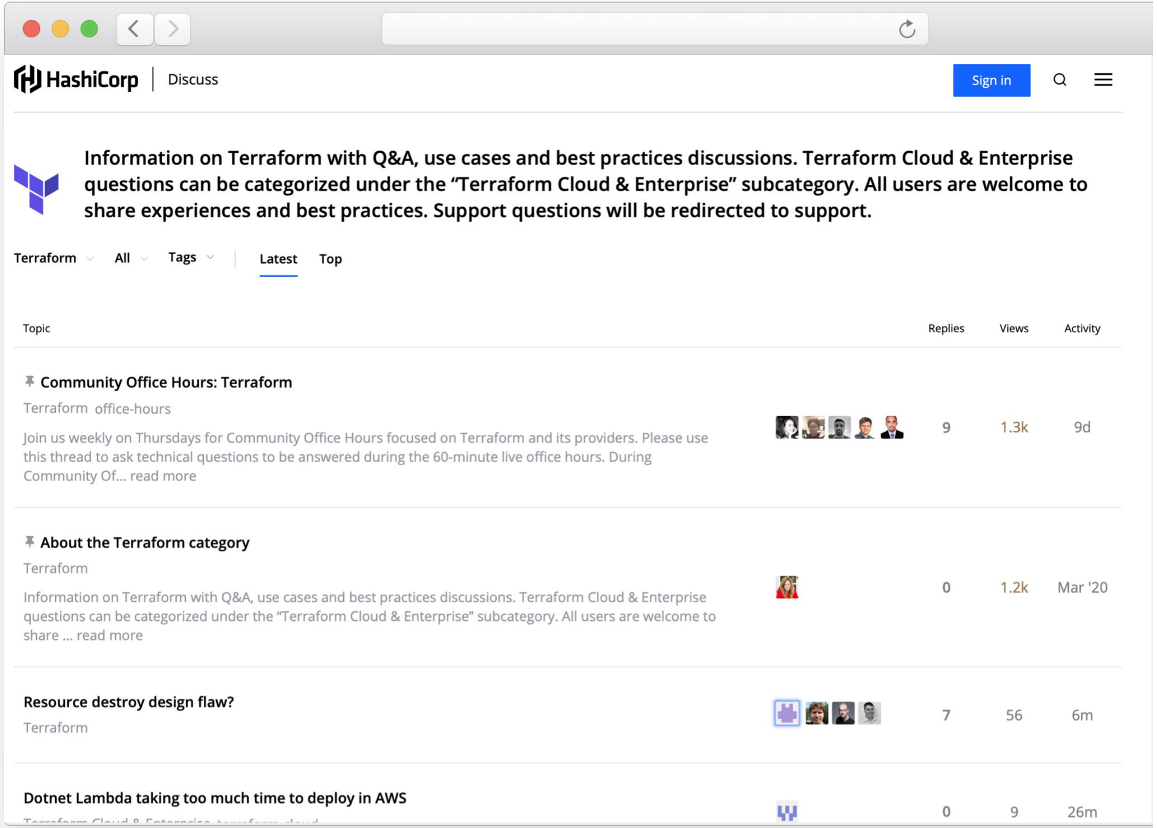
Check with your CSM for invites to our upcoming Webinars

Modules, Workspaces, Git Repository

Learn how to architect your Terraform Modules, how to segment your Workspaces, and how to organize your Terraform Code within your Git Repository.

Cloud Agents, RBAC, Sentinel

Learn how to use Cloud Agents for provisioning resources on-prem and across multiple cloud accounts; how to configure Workspace RBAC permissions, and use Sentinel.


HashiCorp | Discuss

Information on Terraform with Q&A, use cases and best practices discussions. Terraform Cloud & Enterprise questions can be categorized under the “Terraform Cloud & Enterprise” subcategory. All users are welcome to share experiences and best practices. Support questions will be redirected to support.

Topic	Replies	Views	Activity
Community Office Hours: Terraform Terraform office-hours Join us weekly on Thursdays for Community Office Hours focused on Terraform and its providers. Please use this thread to ask technical questions to be answered during the 60-minute live office hours. During Community Of… read more	9	1.3k	9d
About the Terraform category Terraform Information on Terraform with Q&A, use cases and best practices discussions. Terraform Cloud & Enterprise questions can be categorized under the “Terraform Cloud & Enterprise” subcategory. All users are welcome to share … read more	0	1.2k	Mar '20
Resource destroy design flaw? Terraform	7	56	6m
Dotnet Lambda taking too much time to deploy in AWS Terraform Cloud & Enterprise terraform-cloud	0	9	26m

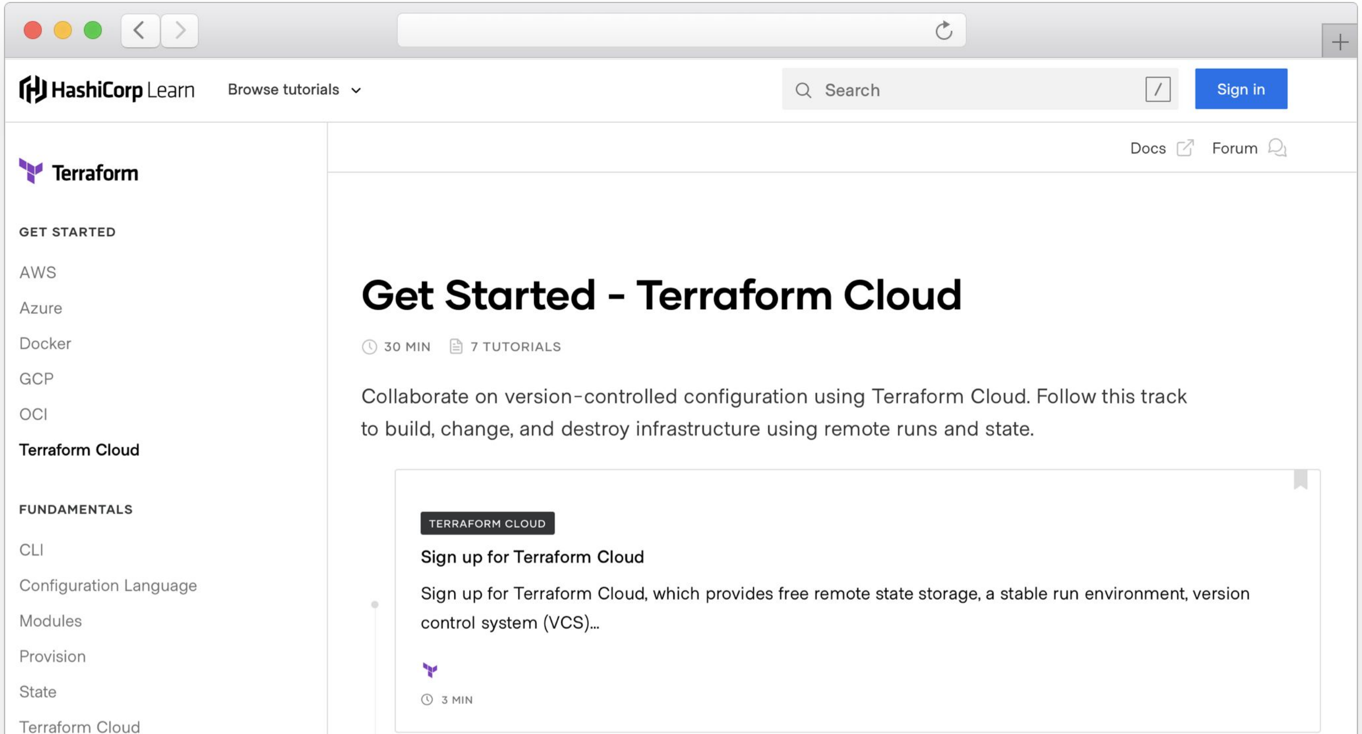
Discuss

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

discuss.hashicorp.com

Learn

Step-by-step guides to implement features in TFCB



Terraform

GET STARTED

- AWS
- Azure
- Docker
- GCP
- OCI
- Terraform Cloud

FUNDAMENTALS

- CLI
- Configuration Language
- Modules
- Provision
- State
- Terraform Cloud

Get Started - Terraform Cloud

30 MIN 7 TUTORIALS

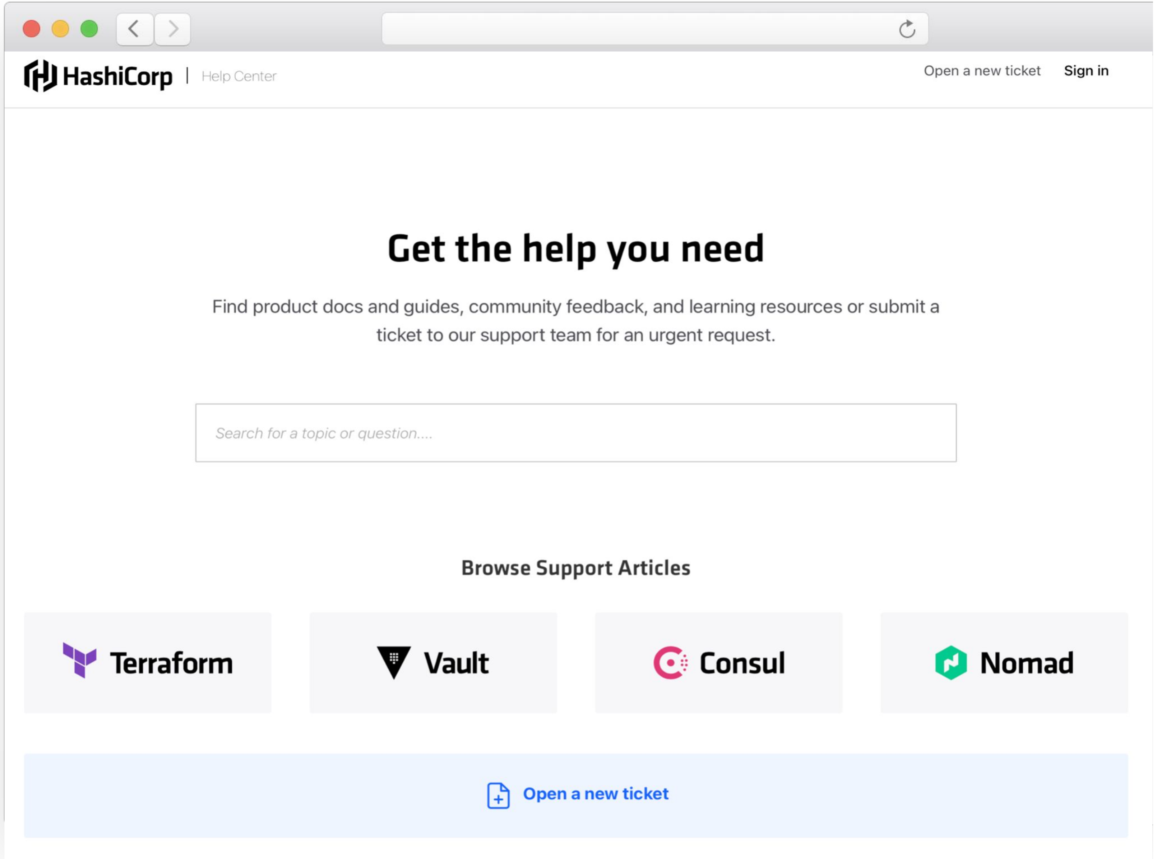
Collaborate on version-controlled configuration using Terraform Cloud. Follow this track to build, change, and destroy infrastructure using remote runs and state.

TERRAFORM CLOUD

Sign up for Terraform Cloud

Sign up for Terraform Cloud, which provides free remote state storage, a stable run environment, version control system (VCS)...

3 MIN



Get the help you need

Find product docs and guides, community feedback, and learning resources or submit a ticket to our support team for an urgent request.

Support

https://support.hashicorp.com

Need Additional Help?

CSM

Contact the HashiCorp Customer Success Management team with any questions. They 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.

Services

Need additional assistance through hands-on implementation support. Your Customer Success Manager can help get you in touch with an implementation services partner to assist with your implementation.

Q & A

Thank You

hello@hashicorp.com
www.hashicorp.com