



# Terraform Cloud Onboarding Program Kickoff Overview



---

# Agenda

- Welcome/Code of Conduct
- Customer Success Overview
- TFCB Onboarding Program
- Customer Support
- Next Steps

# Code of Conduct



**HashiCorp is dedicated to providing a harassment-free Terraform Cloud OnBoarding experience for everyone, regardless of gender, gender identity, sexual orientation, disability, physical appearance, body size, race, national origin, or religion. We value your attendance and do not wish anyone to feel uncomfortable or threatened at any time.**

The bottom line is that we do not tolerate harassment of conference participants in any form. Harassment includes but is not limited to offensive verbal comments related to gender, gender identity, sexual orientation, disability, physical appearance, body size, race, national origin, religion; sexual or inappropriate images in public spaces; deliberate intimidation; stalking; trolling; sustained disruption of talks or other events; and unwelcome sexual attention. Participants asked to stop any harassing behavior are expected to comply immediately. If you are being harassed, notice that someone else is being harassed, or have any other concerns, please let the HashiCorp event representative know immediately or email [customer.success@hashicorp.com](mailto:customer.success@hashicorp.com).

# Customer Success Overview

Partnering Together

# HashiCorp Customers



## FINANCIAL SERVICES



## ENTERTAINMENT & TELCO



## MANUFACTURING & LOGISTICS



## SOFTWARE & TECHNOLOGY



## INSURANCE &





# What You Can Expect from CS

## **Customer Success Manager (CSM)**

Account & Success Management

- Providing a community-based onboarding program designed to get you up and running quickly
- Facilitating sessions to keep your team current with HashiCorp technology
- Joint discovery of objectives and success criteria
- Your customer advocate within HashiCorp

## **Customer Success Architect (CSA)**

Technical Success & Advisory

- Technical resource for the onboarding process
- Providing product reference architecture information for better decision-making
- Thought leadership on best practices of product architecture and use-case patterns
- Timely education and enablement from a technical perspective

# Other resources available to you



## Ensure your team's success



### Worldwide Support

With HashiCorp Worldwide Support, you can get assistance when you need it from anywhere in the world with our ready-to-serve ticketing system and expert support team.

[Learn More](#)



### Implementation Services

Let highly skilled product domain experts help you achieve success by simplifying and accelerating the adoption of our cloud solutions starting at the implementation phase.

[Learn More](#)

Further information located at <http://hashicorp.com/customer-success>

# TFC Onboarding Program

Preview

# Customer Responsibilities

These are critical for your onboarding success



## Training Consumption

Ensure team members attend workshops, training

## Use Case Guidance

Provide timely information on your intended use cases

## Project Team Participation

Inclusive of any stakeholder required for successful completion of your onboarding

## Single Point of Contact

Main contact for decision making

## Escalation Process

Understanding of escalation process

## Surveys Responses

Provide timely responses to surveys



# Onboarding Checklist



## Terraform Cloud Configured

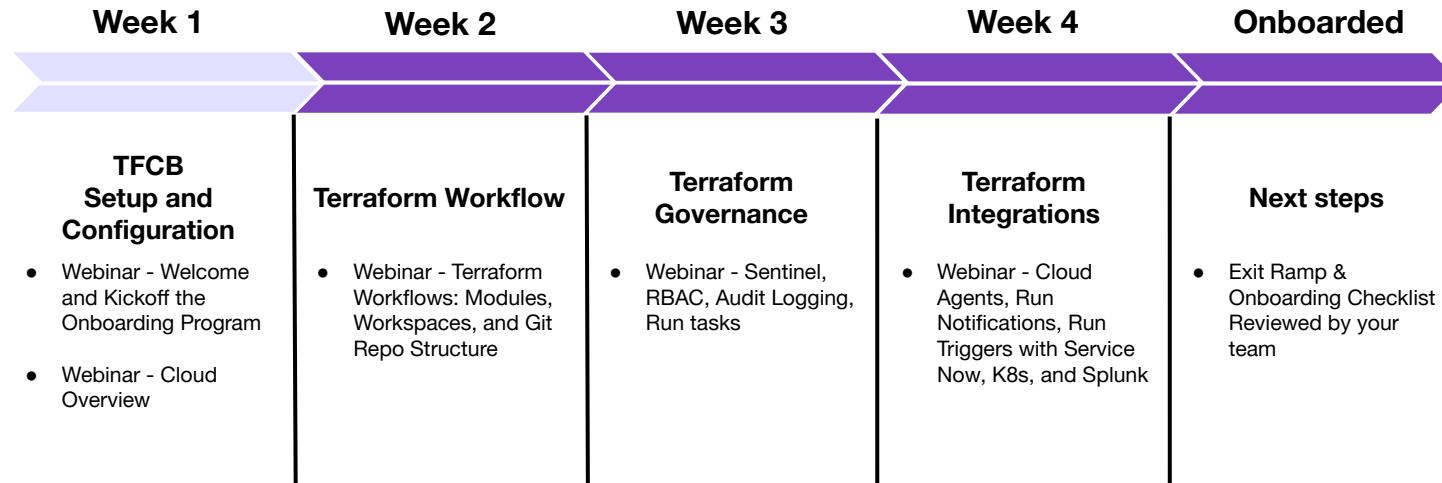
- Terraform Organization created
- Terraform workspaces configured to at least 1 workflow (i.e.: API, CLI, VCS or UI)
- Standardize deployments using modules and private registry
- Enforce policy across workspaces

## Terraform Cloud Adoption

- Getting the first use case (team/service/application) onboarded and consuming Terraform Cloud
- A roadmap created for onboarding additional use cases and validated with a HashiCorp CSM

**Completed within 30 days**

# TFCB Path to Production



# Customer Support

SLA, Contact Methods, Services, etc.

# Support Levels

This info can also be accessed from our [Support SLA Page](#)



GOLD

SILVER

BRONZE

		24 X 7 (SEV-1 URGENT)	9-5, Monday - Friday US LOCAL TIME EUROPEAN CENTRAL TIME AUSTRALIA EASTERN TIME	N/A
SEVERITY 1	FIRST RESPONSE	60 minutes	4 business hours	N/A
	UPDATE FREQUENCY	4 hours	8 business hours	N/A
SEVERITY 2	FIRST RESPONSE	4 business hours	8 business hours	N/A
	UPDATE FREQUENCY	8 business hours	2 business days	N/A
SEVERITY 3	FIRST RESPONSE	8 business hours	24 business hours	N/A
	UPDATE FREQUENCY	3 business days	5 business days	N/A
SEVERITY 4	FIRST RESPONSE	24 business hours	24 business hours	24 business hours
	UPDATE FREQUENCY	Reasonable best effort	Reasonable best effort	Reasonable best effort
Technical contacts allowed		4	3	2



# Severity Definitions

<b>Sev-1 (Urgent)</b>	A Sev-1 incident is an operational outage as defined below: Any error reported by customer where majority of the users for a particular part of the software are affected, the error has high visibility, <b>there is no workaround</b> , and <b>it affects the customer's ability to perform its business</b> .
<b>Sev-2 (High)</b>	Any error reported by customer where the majority of the users for a particular part of the software are affected, the error has high visibility, <b>a workaround is available</b> ; however, <b>performance may be degraded or functions limited and it is affecting revenue</b> .
<b>Sev-3 (Normal)</b>	Any error reported by customer where the majority of the users for a particular part of the software are affected, the error has high visibility, a workaround is available; however, performance may be degraded or functions limited and it is NOT affecting revenue.
<b>Sev-4 (Low)</b>	Any error reported by customer where a single user is severely affected or completely inoperable or a small percentage of users are moderately affected or partially inoperable and the error has limited business impact.

For reference only - Subject to Change  
Current info can also be accessed at the bottom of our [Support SLA Page](#)



# Contacting Support

There are two ways to contact our support team:

1) **Support Portal:** Open a ticket through [our support portal](#)

- Once customer access is setup, authorized users can submit a ticket using the email address they provided us.
- The portal provides faster routing via product and sub-product selection, the ability to send encrypted attachments, and set ticket priority.

2) **Email Support:** Send an email to [support@hashicorp.com](mailto:support@hashicorp.com)

- All emailed support tickets default to “normal” priority - and cannot be changed.
- Don’t raise a SEV-1 over email. Please use the support portal!

# Support Portal



Authorized technical contacts can log in through the “Sign in” button

The screenshot shows the HashiCorp Help Center homepage. At the top left is the HashiCorp logo and "Help Center". To the right are links for "Open a new ticket" and "Sign in", with "Sign in" circled in yellow. Below this is a large heading "Get the help you need" and a subtext: "Find product docs and guides, community feedback, and learning resources or submit a ticket to our support team for an urgent request." A search bar below contains the placeholder text "Search for a topic or question....". At the bottom center is a "Browse Support Articles" link. The footer features logos for Terraform, Nomad, and Consul.

HashiCorp | Help Center

Open a new ticket **Sign in**

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.

Search for a topic or question....

Browse Support Articles

Terraform

Support Portal Link  
<https://support.hashicorp.com/hc/en-us>

Nomad

# Interacting with HashiCorp Support



## Terraform - Best Practices

When submitting a ticket, provide as much detail as possible...

Terraform Cloud Specific		If using CLI, provide...
<b>Organization Name &amp; Workspace Name</b>	<ul style="list-style-type: none"><li>Name of your organization in Terraform Cloud and the name of the workspace you are working with directly.</li></ul>	<b>Operating System (version)</b>
<b>Run ID</b>	<ul style="list-style-type: none"><li>The ID for the run you are working with. (e.g. #run-XXX1234)</li></ul>	<b>Platform Details (physical/virtual)</b>
<b>Run Errors</b>	<ul style="list-style-type: none"><li>Provide debug logs by setting <code>TF_LOG</code> environment variable.</li></ul>	<b>Cloud Provider(s)</b>
<b>Terraform CLI Version</b>	<ul style="list-style-type: none"><li>Terraform version can be found within Workspace settings</li><li>If using Terraform CLI, the CLI version can be found using <code>terraform version</code>.</li></ul>	

# Recommended additional resources



We strongly urge you to subscribe to the Terraform Cloud status web page, this can be done here

<https://status.hashicorp.com/>

We also recommend and hope you will take an active part in the Hashicorp community, you can find more information about that here

<https://www.hashicorp.com/community>

# Next Steps

# Next Steps



- Share all stakeholders contact information HashiCorp & Your Organization
- Share your authorized technical contacts for support
- Identify your use case and define your goals with TFCB
- Schedule for the next meetings:
  - **Week 1** June 23rd Kickoff/Cloud Overview - **YOU ARE HERE**
  - **Week 2** June 30th Terraform Workflow Management
  - **Week 3** July 7th Terraform Governance
  - **Week 4** July 14th Terraform Integrations/Closing

# Terraform Cloud for Business (TFCB) Getting Started

April 27, 2022



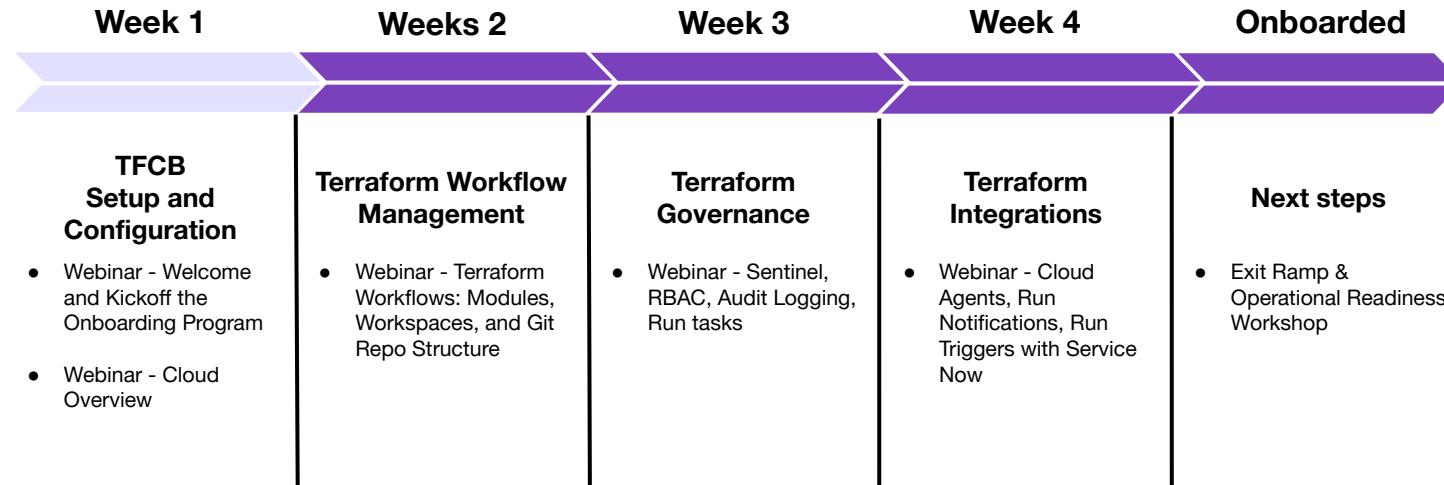
---

# Agenda

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

# Terraform Cloud for Business Overview

# TFCB Path to Production





---

# Terraform Cloud for Business Overview

- Overview
- Organizations
- SSO, Teams, Users
- Private Module Registry
- Workspaces
- Run Types
- What's new in Terraform

# Terraform Cloud for Business



- Central platform running in the cloud, that HashiCorp manages for you.
- Terraform Cloud has a robust set of enterprise-ready features, this includes:
  - SSO, Teams, Users, Tokens, RBAC which are a priority to set up during our onboarding initiatives.
  - VCS Connections, Private Module Registry, Workspaces, State Management, Variables to facilitate collaboration across your users and teams.
  - Cost Estimation, Run Triggers, Run Tasks, Run Notifications, Policy as Code with Sentinel to support governance and compliance needs.
- Terraform was built to establish a Producer/Consumer model, to create a separation of duties across your ops and devs teams. We will go into more details of this during our session for Terraform Workflows



# Organizations

## Overview

Organizations is a security boundary for TFCB. Users can belong to multiple organizations and they can self-select in the UI to operate in the organization they choose.

## 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.

## Teams + Organizations

Teams can also be assigned organization-level permissions including:

- Managing Policies,
- Manage Workspaces,
- 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

Workspaces can be run through the following ways:

1. Uploading a .zip file of TF code via the API
2. Connected to a Git Repository from your VCS provider and will monitor for changes using Git Webhooks.

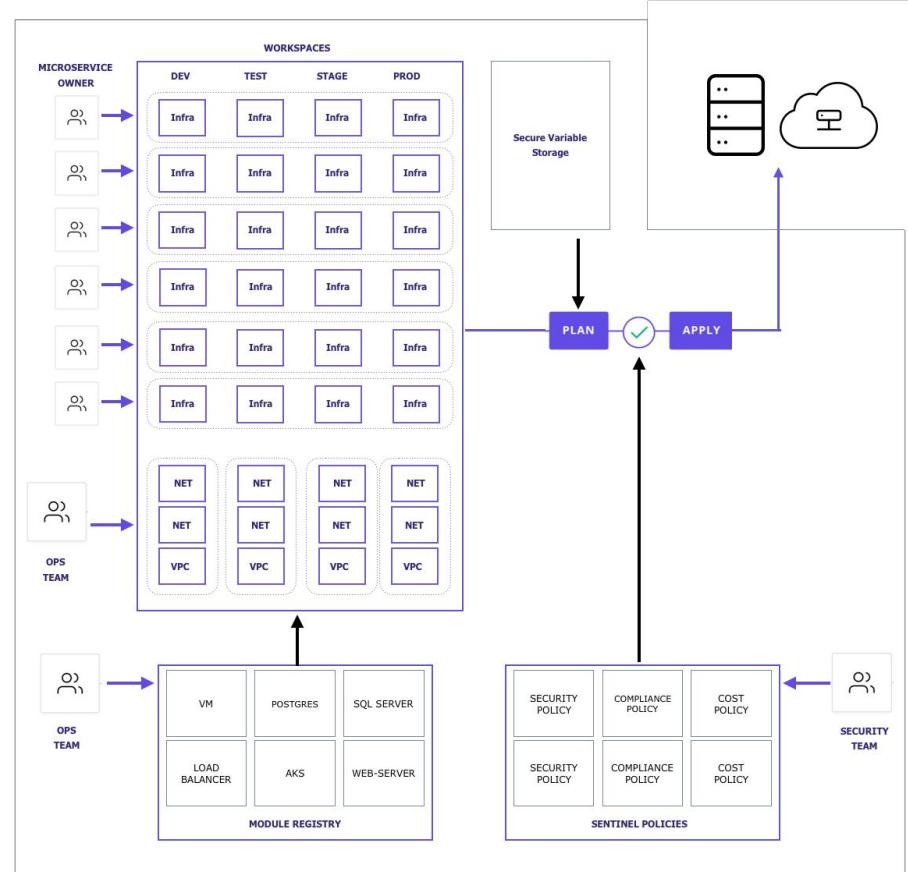
## Workspaces Contain:

- 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.

1. Each Workspace is connected with a specific Git Branch in a Git Repo on your VCS Provider.
2. TFCB registers Git Webhooks with your VCS Provider during Workspace creation.
3. 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.

# VCS Integration



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](#)

# 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.

## Use API-runs for 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>



# 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.

The screenshot shows the Terraform Registry interface. At the top, there's a search bar, a browse button, and a sign-in link. Below the header, a module card for 'vnet' by 'AZURERM' is displayed. The card includes a thumbnail of the Azure logo, the module name 'vnet' with a blue checkmark, and a 'Version 1.2.0' dropdown. A brief description states: 'Terraform module to create/provision Azure vnet'. Below the card, it says 'Published August 15, 2018 by Azure', 'Module managed by rguthrie82ft', 'Total provisions: 13,132', and 'Source: [github.com/Azure/terraform-azurerm-vnet](https://github.com/Azure/terraform-azurerm-vnet) (report an issue)'. On the right side of the card, there's a 'Provision Instructions' section with a code snippet:

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

Below the card, there are navigation links for 'Readme', 'Inputs (9)', 'Outputs (5)', 'Dependencies (0)', and 'Resources (3)'. Underneath these links, the module's GitHub repository name 'terraform-azurerm-vnet' is shown, along with a green 'build passing' badge. A 'Create a basic virtual network in Azure' section follows, with a note about the module's function and security requirements. Finally, there's a 'Usage' section with an 'HCL' tab containing the following code:

```
1 module "vnet" {  
2   source = "Azure/vnet/azurerm"  
}
```

There's also a 'Copy' button next to the code block.



# 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 ▾

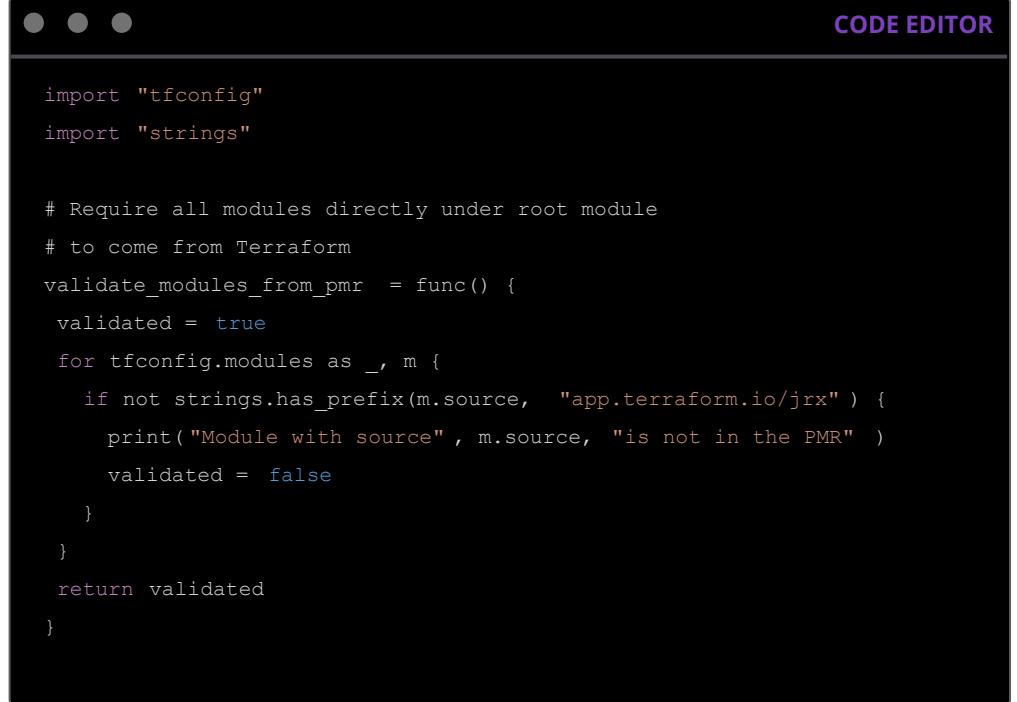
# Sentinel



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.



The image shows a dark-themed code editor interface. At the top, there are three circular navigation icons (left, center, right). To the right of these is a purple "CODE EDITOR" label. The main area contains a snippet of Python-like pseudocode:

```
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>



# HashiConf 2022 Updates



---

# What's new with Terraform?

[V1.2 Changelog](#)

- Terraform v1.2
- Drift Detection
- Run Tasks GA

# Terraform v1.2

The theme of Terraform v1.2 release was to focus on improvements to areas of our Terraform Core, improving the practitioner's experience via ***pre/post conditions***, ***Run Tasks CLI support***, and updates to the ***non-interactive cloud CI Support***.

[V1.2 Changelog](#)

Need upgrade help? Please contact  
[customer.success@hashicorp.com](mailto:customer.success@hashicorp.com)

# Drift Detection



- We recognize not every organization follows Infrastructure as Code
- Cloud or service providers make changes to their API that's not updated in your configuration
- Emergency “Break-ceiling” scenarios

## Enter Drift Detection, new in Terraform Cloud

The screenshot shows the Terraform Cloud interface for the workspace 'hashicorp-v2'. The top navigation bar includes links for Workspaces, Usage, Registry, Compliance, Settings, and HashiCorp Cloud Platform. The main content area is titled 'Workspaces' with a subtitle '20 of 122 matching'. A search bar at the top right allows for 'New workspace'. Below this, a table lists 20 workspaces, each with columns for Name, Run status, Source, and Latest change. The table includes filters for Tag, Status, Drift, and Sort, along with a search bar. The first workspace listed is 'kitchen-sink', which has an 'Errored' status and was run by 'terr...security-group@master' a few seconds ago. Other workspaces shown include 'aws-shared-east-SHARD', 'attlast-mixin', 'aws-app-c000002-shared-edo-shared-east-shard', 'c000002-shared-edo-shared-east', and 'sandbox-run-triggerrs'. Most workspaces have an 'Applied' status, while some like 'kitchen-sink' and 'attlast-mixin' show 'Drift' or 'Drift check failed' status.

Name	Run status	Source	Latest change
kitchen-sink	Errored	terr...security-group@master	a few seconds ago
aws-shared-east-SHARD	Applied	terr...security-group@master	a day ago
attlast-mixin	Applied	terr...security-group@master	2 days ago
aws-app-c000002-shared-edo-shared-east-shard	Applied	terr...security-group@master	2 days ago
c000002-shared-edo-shared-east	Applied	terr...security-group@master	2 days ago
sandbox-run-triggerrs	Applied	terr...security-group@master	a few seconds ago

# Run Tasks is now GA!



## What are Run Tasks?

Run Tasks allow you to integrate third-party tools into the pre-apply stage during a Terraform Cloud run. During the pre-apply phase an event hook is triggered and Terraform will send a payload with the details of the run. Terraform will then wait for the service to reply with either passed or failed status.

### *When to use Run Tasks and when to use Sentinel?*

The integrations can include security vulnerability scanners, resource cost management, code scanning and through these integrations, you may not need to write these Sentinel policies yourself.

# Next Steps

# Upcoming Onboarding Webinars



**Check with your CSM for invites to our upcoming Webinars**

## Terraform Workflows

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

## Terraform Governance

Learn how to configure Workspace RBAC permissions, leverage Run Tasks and use Sentinel.

The screenshot shows a web browser window with the HashiCorp logo and "Discuss" tab selected. The main content area displays a list of topics under the "Terraform" category. Each topic card includes a thumbnail, the title, a brief description, the number of replies, views, and the last update time.

- 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](#)
- 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 experiences and best practices. Support questions will be redirected to support.
- Resource destroy design flaw?**  
Terraform
- Dotnet Lambda taking too much time to deploy in AWS**  
Terraform Cloud & Enterprise [read more](#)



# Discuss

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

[discuss.hashicorp.com](https://discuss.hashicorp.com)

# Learn



Step-by-step guides to implement features in TFCB

A screenshot of a web browser displaying the HashiCorp Learn platform. The page title is "HashiCorp Learn". The main content area features a large heading "Get Started - Terraform Cloud" with a subtext "Collaborate on version-controlled configuration using Terraform Cloud. Follow this track to build, change, and destroy infrastructure using remote runs and state." Below the heading, there is a summary card for the "Sign up for Terraform Cloud" tutorial. The left sidebar contains navigation links for "Terraform", "GET STARTED" (with options for AWS, Azure, Docker, GCP, OCI, and Terraform Cloud), and "FUNDAMENTALS" (with options for CLI, Configuration Language, Modules, Provision, State, and Terraform Cloud).

HashiCorp Learn

Browse tutorials

Search

Sign in

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

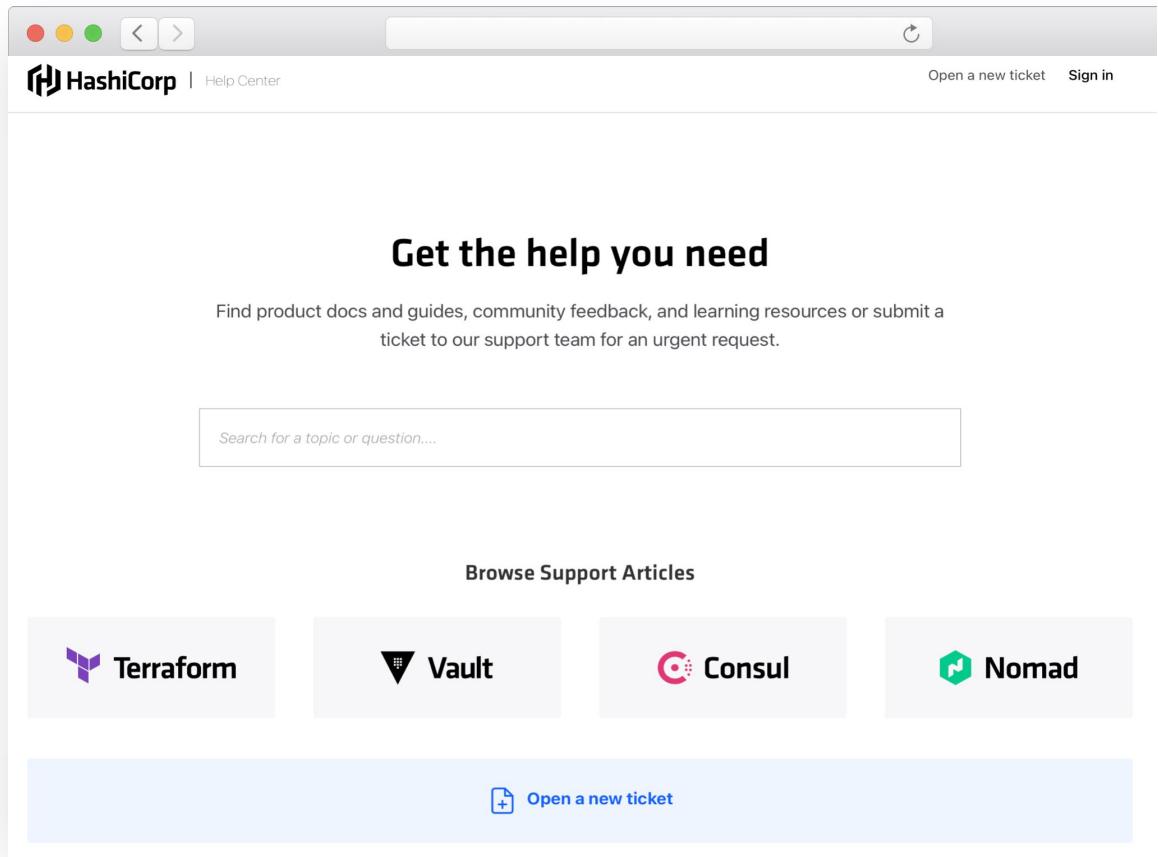
Collaborate on version-controlled configuration using Terraform Cloud. Follow this track to build, change, and destroy infrastructure using remote runs and state.

**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



A screenshot of a web browser showing the HashiCorp Support Center. The page has a light gray header with the HashiCorp logo and "Help Center" text. On the right, there are links for "Open a new ticket" and "Sign in". Below the header, a large section features the text "Get the help you need" and a subtext about finding product docs, guides, community feedback, and learning resources or submitting a ticket. A search bar is present with the placeholder "Search for a topic or question....". Below the search bar, there's a section titled "Browse Support Articles" with four categories: Terraform, Vault, Consul, and Nomad. At the bottom, a blue button says "Open a new ticket" with a plus sign icon.

HashiCorp | Help Center

Open a new ticket Sign in

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.

Search for a topic or question....

Browse Support Articles

Terraform

Vault

Consul

Nomad

Open a new ticket



# 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](mailto: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](https://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

[customer.success@hashicorp.com](mailto:customer.success@hashicorp.com)  
[www.hashicorp.com/customer-success](http://www.hashicorp.com/customer-success)