

Terraform Cloud Onboarding Program



Agenda

Customer Success Overview

01

TFCB Onboarding Program

02

Customer Support

03

TFCB Getting Started

04

Q&A

05



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.



01

Customer Success Overview



HashiCorp Customers

FINANCIAL SERVICES	ENTERTAINMENT & TELCO	MANUFACTURING & LOGISTICS	SOFTWARE & TECHNOLOGY	INSURANCE & HEALTH	
 Santander  KeyBank   SoftBank  RBC  wepay a CHASE company  Blackstone  Lincoln Financial Group®	 BNP PARIBAS  CREDICORP   ABN AMRO  Nationwide Building Society  STANDARD & POOR'S  ADB	 COMCAST  vodafone  NBCUniversal  UBISOFT  sky  RED VENTURES  DAZN  VINGROUP  ROBLOX	 gm  Lufthansa  BHP  OLD DOMINION FREIGHT LINE  AIRBUS  AirPlus INTERNATIONAL  WARE2GO  KPMG	 Booking.com  Grab  priceline.com®  cielo  shopify  SEAT GEEK  H&R BLOCK  ADT  Shipt  Q2	 PROGRESSIVE  co-operators  gsk  AXA  AstraZeneca  ellume Kansas City  athenahealth  GoodRx  surescripts



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

Solutions Architecture Specialist (SA) 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>



02

TFCB Onboarding Program



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



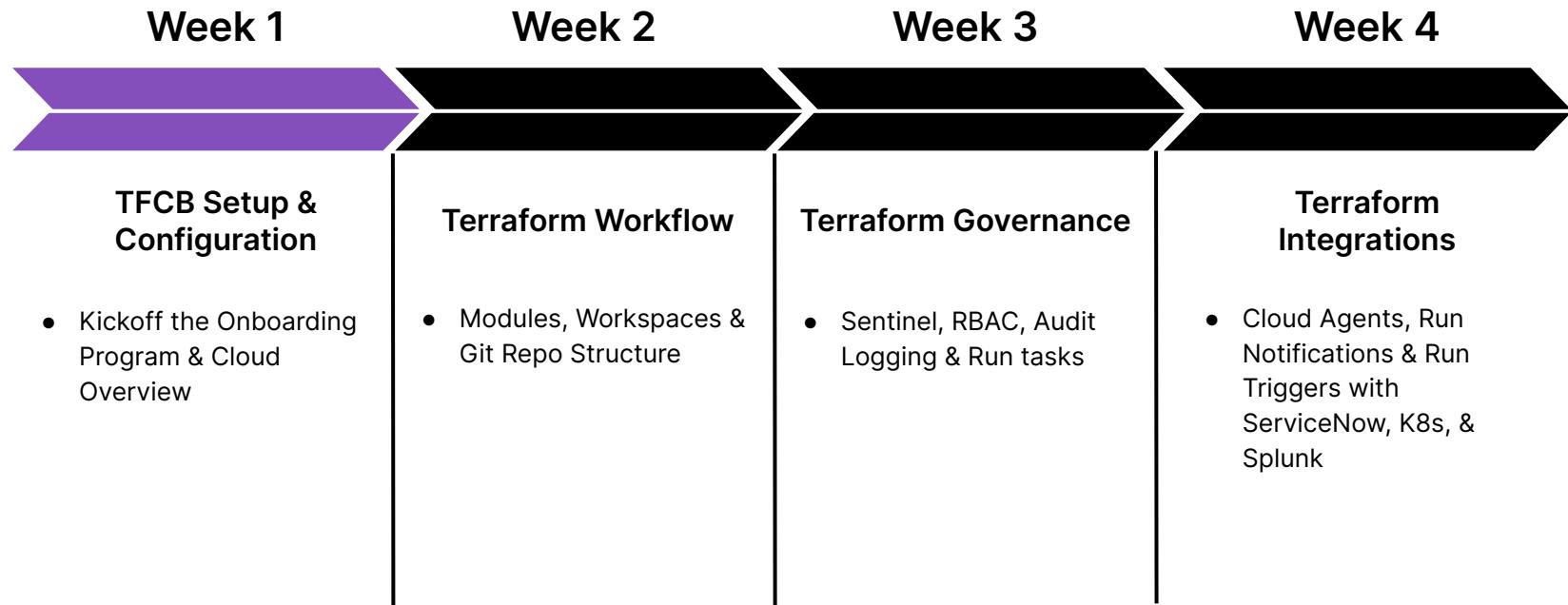
Terraform Cloud Adoption

- 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

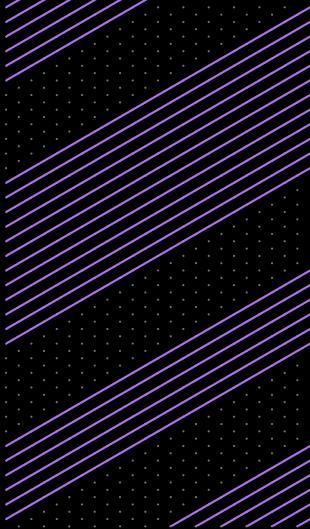


Completed within 30 days

TFCB Path to Production



03



Customer Support

Support Levels

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

		BRONZE	SILVER	GOLD
Hours of availability		N/A	9-5, Monday - Friday US PACIFIC TIME EUROPEAN CENTRAL TIME AUSTRALIA EASTERN TIME INDIA STANDARD TIME	24 X 7 (SEV-1 URGENT)
SEVERITY 1	FIRST RESPONSE	N/A	4 business hours*	60 minutes
	UPDATE FREQUENCY	N/A	8 business hours*	4 hours
SEVERITY 2	FIRST RESPONSE	N/A	8 business hours	4 business hours
	UPDATE FREQUENCY	N/A	2 business days	8 business hours
SEVERITY 3	FIRST RESPONSE	N/A	24 business hours	8 business hours
	UPDATE FREQUENCY	N/A	5 business days	3 business days
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		2	3	4
* Clock hours for HCP Cluster down and TFC Stuck Runs				



Severity Definitions

Sev-1 (Urgent)	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, there is no workaround , and it affects the customer's ability to perform its business .
Sev-2 (High)	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 affecting revenue .
Sev-3 (Normal)	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.
Sev-4 (Low)	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

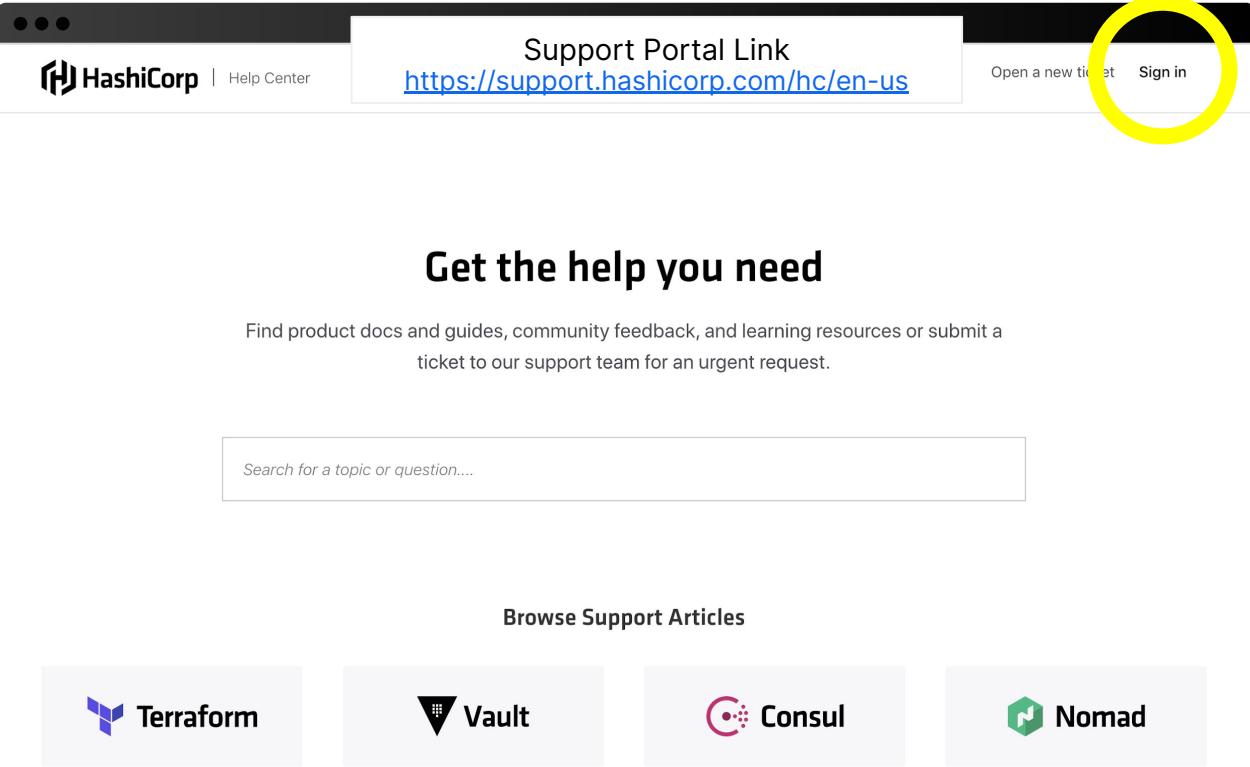
There are two ways to contact our support team:

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
 - All emailed support tickets default to “normal” priority - and cannot be changed
 - Do not 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 Support Portal homepage. At the top, there's a navigation bar with the HashiCorp logo and a "Help Center" link. Below the navigation bar, the text "Support Portal Link" is displayed along with a blue hyperlink: <https://support.hashicorp.com/hc/en-us>. To the right of the link are two buttons: "Open a new ticket" and "Sign in". The "Sign in" button is circled in yellow. The main content area features the 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." Below this is a search bar with the placeholder text "Search for a topic or question....". Further down, there's a "Browse Support Articles" section and four cards representing different products: Terraform, Vault, Consul, and Nomad, each with its respective logo and name.

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

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

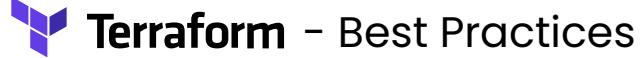
Vault

Consul

Nomad



Interacting with HashiCorp Support



Terraform – Best Practices

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

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

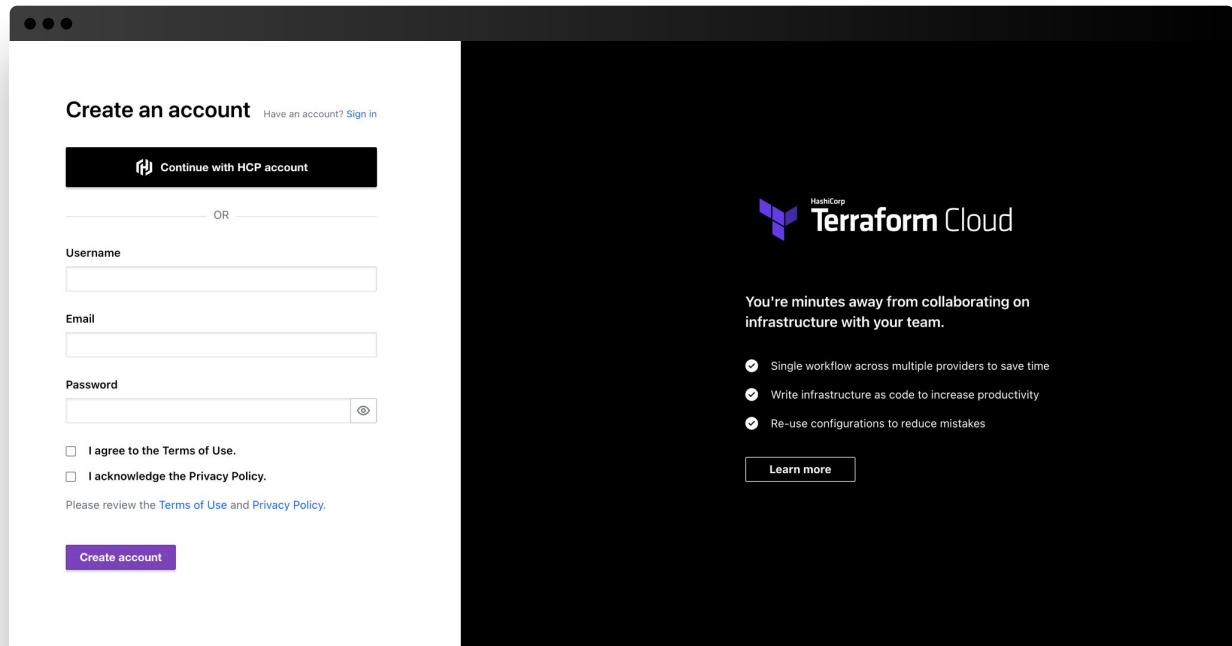
04

Terraform Cloud for Business (TFCB) Getting Started



Terraform Cloud for Business

- Central platform running in the cloud, that HashiCorp manages for you
- Terraform was built to establish a Producer/Consumer model, to create a separation of duties across your ops and devs teams (covered in depth during Terraform Workflows webinar)



Terraform Cloud for Business

Terraform Cloud has a robust set of enterprise-ready features including:



SSO, Teams, Users, Tokens, RBAC which are **a priority to set up** during our onboarding initiatives



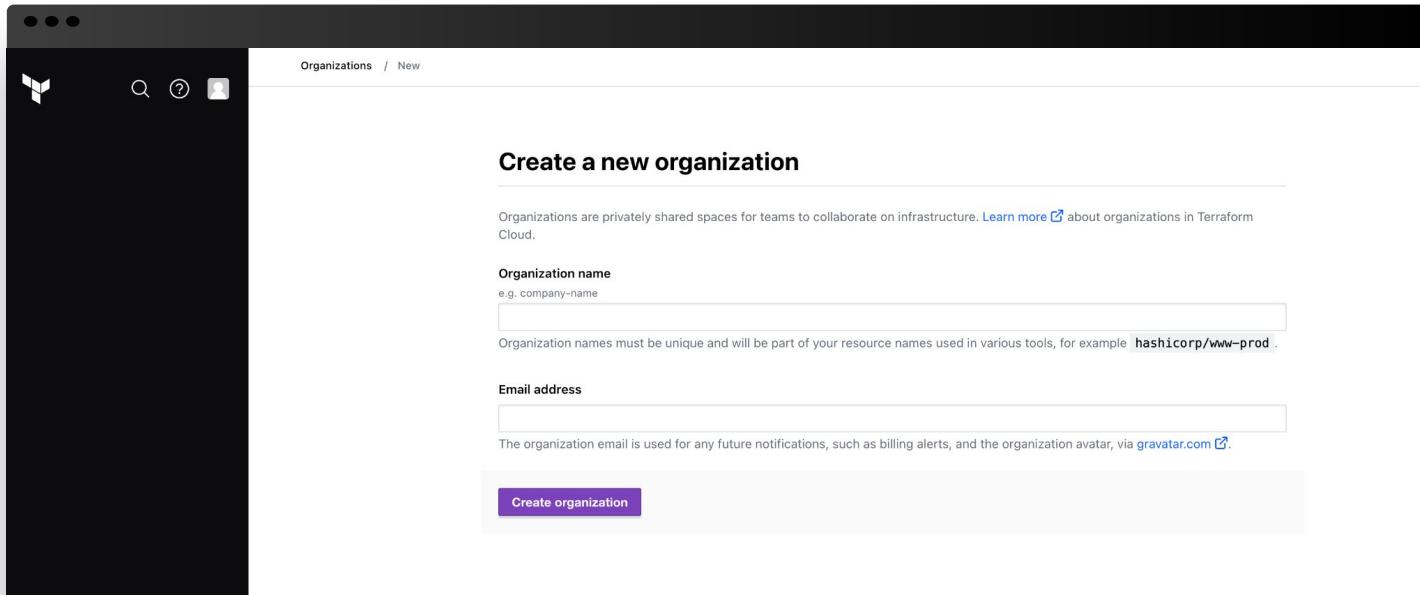
VCS Connections, Private Module Registry, Workspaces, State Management, Variables to **facilitate collaboration** across your users and teams



Cost Estimation, Run Triggers, Run Tasks, Run Notifications, Policy as Code with Sentinel to **support governance and compliance** needs

Organizations

- Security boundary and shared space for teams to collaborate on workspaces
- Users can belong to multiple organizations, the UI allows users to self-select and operate in the organization they choose



Organizations Components

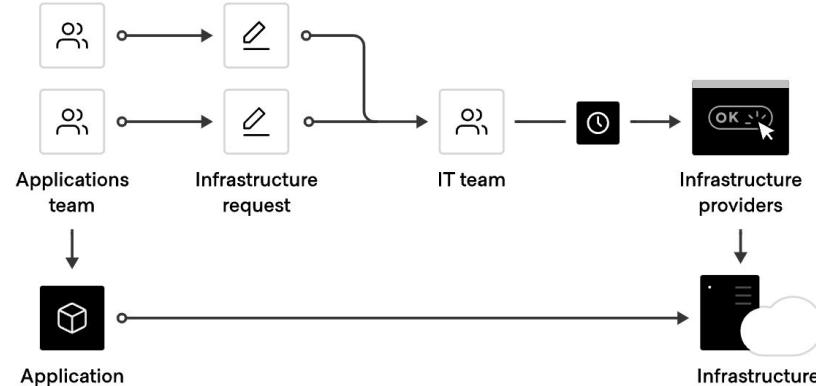
- SSO Settings
- Teams
- Users
- API Tokens (Org, Teams, Users)
- VCS Provider / Git Connections
- Private Module Registry
- Workspaces (TF Code + Statefile)
- Variables, ENV Variables, CLI Flags
- SSH Keys
- Sentinel Policy Sets
- Cloud Agents



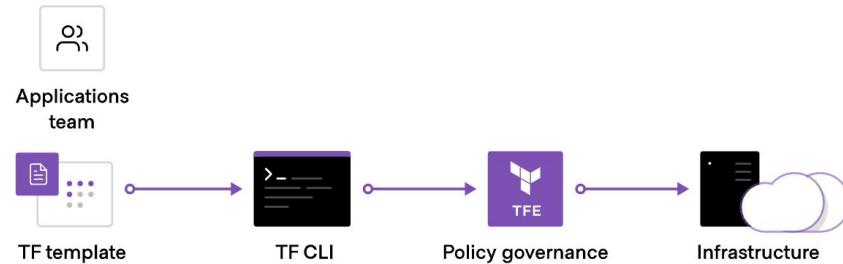
Teams

- 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

Before Terraform



After Terraform



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 all of the above permissions



Users

- 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

- Default authentication method
- Allows users to self register
- Requires users to provide an email address and password

SAML SSO

- Direct integrations with Azure AD and Okta for single sign on are included
- TFCB can also integrate with your SAML capable identity provider

API Tokens

Once logged in, users can generate API token(s)

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
- Enabling SAML causes the login prompt to redirect users to the IDP for login and then redirects back to TFCB upon authentication
- Team membership mapping can be enabled so users are added to teams based on SAML attribute assertion

Identity Provider Guides

[Azure Active Directory](#)

[Okta](#)

[SAML](#)



Service Accounts

Team Service Accounts

- Designed to perform API operations on workspaces
- API token will have the same access and permissions as team
- 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 delegation of workspace(s) to team(s)
- Team service account should be used for regular operations



Workspaces

Workspaces consist of...

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



Workspaces

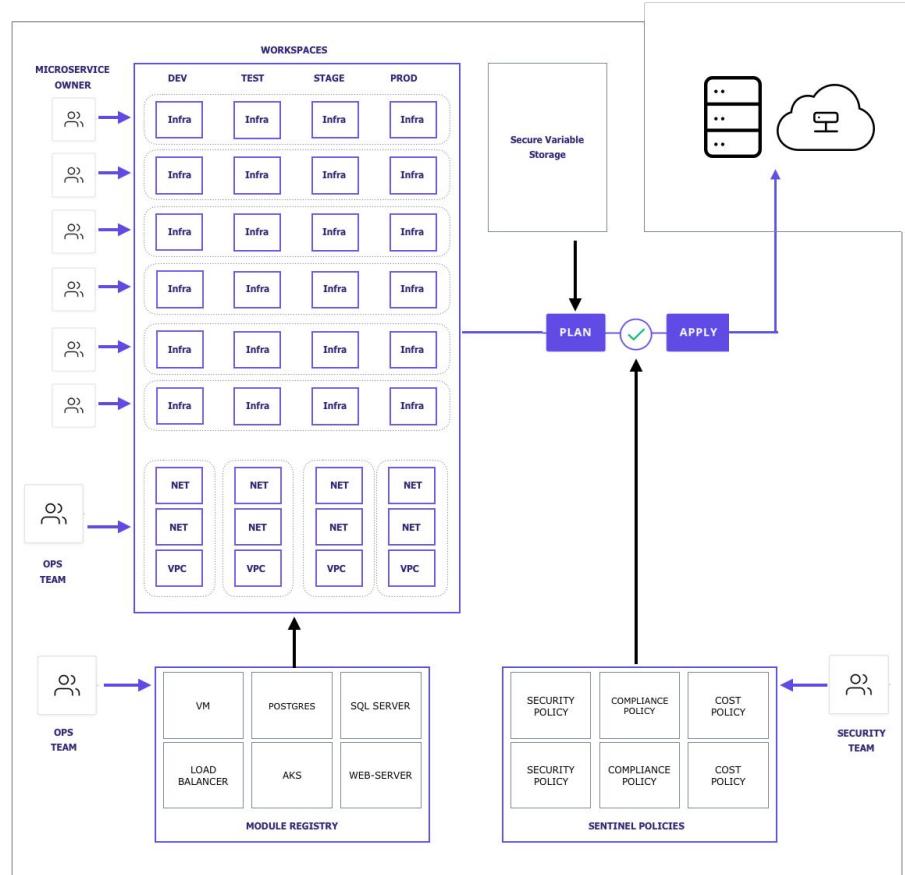
Workspaces can be run in the following ways:

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

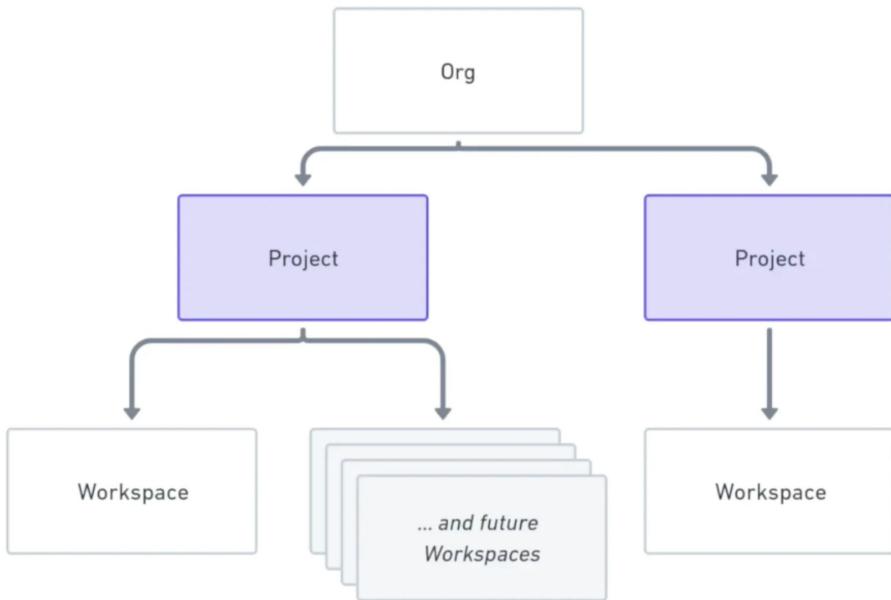
The screenshot shows the Terraform Cloud interface. On the left, there's a dark sidebar with a navigation menu: 'Manage', 'Projects & workspaces' (which is highlighted), 'Registry', 'Settings', 'Cloud Platform', and 'HashiCorp Cloud Platform'. The main content area has a header 'learn-training / Projects & workspaces / New Workspace'. Below the header, it says 'Create a new Workspace'. A descriptive text explains that Workspaces determine how Terraform Cloud organizes infrastructure, mentioning Terraform configuration, shared variable values, current and historical state, and run logs. It includes a 'Learn more' link. Below this, there are four numbered steps: 1. Choose Type, 2. Connect to VCS, 3. Choose a repository, and 4. Configure settings. The first step, 'Choose Type', is currently selected. The 'Choose your workflow' section contains three options: 'Version control workflow' (selected, described as 'Most common'), 'CLI-driven workflow' (described as 'Trigger remote Terraform runs from your local command line.'), and 'API-driven workflow' (described as 'A more advanced option. Integrate Terraform into a larger pipeline using the Terraform API.'). Each workflow option has a 'Learn More' link.

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



Projects



- Projects let you organize workspaces and scope access to workspace resources
- Each project has a separate permissions set which can be used to manage access to all workspaces in the project
- Project-level permissions
 - More granular than Org-level permissions
 - More specific than workspace-level grants
- Projects added in TFE 202302-1 (Feb 2023)

UI/VCS-Driven Runs

- UI and VCS workflows are the primary mode of operation in TFCB
 - 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



UI/VCS-Driven Runs

Auto-apply

1. By default, runs require confirmation before TFCB will apply them
2. Auto-apply can be configured in the workspaces “General Settings” page

The screenshot shows a Terraform Cloud workspace interface. At the top, a green checkmark icon indicates "Plan finished 34 minutes ago". Below it, a timeline shows "Queued 34 minutes ago > Started 34 minutes ago > Finished 34 minutes ago". On the right, a status bar says "Resources: 1 to add, 0 to change, 1 to destroy". The main area displays a "Terraform will perform the following actions:" block with a detailed list of resources being created or destroyed. At the bottom, a button labeled "Apply pending" is shown, along with a note: "Needs Confirmation: Check the plan and confirm to apply it, or discard the run." Three buttons are available: "Confirm & Apply" (highlighted in blue), "Discard Run", and "Add Comment".

```
✓ Plan finished 34 minutes ago
Resources: 1 to add, 0 to change, 1 to destroy
queued 34 minutes ago > started 34 minutes ago > finished 34 minutes ago
View raw log
Top Bottom Expand Full Screen
/+ destroy and then create replacement
Terraform will perform the following actions:
/+ random_id.random (new resource required)
  id:      "3zkmEFTYik8" => <computed> (forces new resource)
  b64:    "3zkmEFTYik8" => <computed>
  b64_std: "3zkmEFTYik8" => <computed>
  b64_url: "3zkmEFTYik8" => <computed>
  byte_length: "8" => "8"
  dec:     "16684929395824298575" => <computed>
  hex:     "df3926105b588a4f" => <computed>
  keepers.%: "1" => "1"
  keepers.uuid: "b5bfe2b7-ca6e-c0cd-7a01-b659aac398f7" => "f87d124d-9155-90d1-c994-a14f4ff8ad" (forces new resource)

Plan: 1 to add, 0 to change, 1 to destroy.

Apply pending
Needs Confirmation: Check the plan and confirm to apply it, or discard the run.
Confirm & Apply Discard Run Add Comment
```

VCS Integration

- TFCB will interact with most providers using the providers API and OAuth token
- Azure DevOps Server & BitBucket Server require an SSH key for downloading repo contents
- TFCB supports integrating with multiple VCS providers within an Organization
- During workspace creation a configured Git provider is selected

Supported VCS Providers

[GitHub](#)

[GitHub Enterprise](#)

[GitLab.com](#)

[GitLab EE and CE](#)

[BitBucket Cloud](#)

[BitBucket Server](#)

[Azure DevOps](#)



CLI-Driven Runs

Remote Backend

- Enables developers who are already familiar with Terraform CLI workflow to integrate with Terraform Cloud for Business
- Runs execute remotely in TFCB while displaying progress in the terminal where the run is executed

Terraform CLI Tool

- Provides a CLI interface that leverages the Terraform Cloud for Business API
- Useful for modifying variables and workspace settings from the terminal

```
hashicorp@root ~ % terraform -help
Usage: terraform [global options] <subcommand> [args]
The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init      Prepare your working directory for other commands
  validate   Check whether the configuration is valid
  plan      Show changes required by the current configuration
  apply      Create or update infrastructure
  destroy    Destroy previously-created infrastructure

All other commands:
  console    Try Terraform expressions at an interactive command prompt
  fmt        Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get        Install or upgrade remote Terraform modules
  graph      Generate a Graphviz graph of the steps in an operation
  import     Associate existing infrastructure with a Terraform resource
  login      Obtain and save credentials for a remote host
  logout     Remove locally-stored credentials for a remote host
  metadata   Metadata related commands
  output     Show output values from your root module
  providers Show the providers required for this configuration
  refresh    Update the state to match remote systems
  show       Show the current state or a saved plan
  state      Advanced state management
  taint      Mark a resource instance as not fully functional
  test       Experimental support for module integration testing
  untaint   Remove the 'tainted' state from a resource instance
  version    Show the current Terraform version
  workspace  Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR Switch to a different working directory before executing the
             given subcommand.
  -help      Show this help output, or the help for a specified subcommand.
  -version   An alias for the "version" subcommand.
```



API-Driven Runs

- Provide a flexible workflow for teams to build tooling to determine when configuration has changed and a run should occur
- Allows for custom integration and configuration from unsupported version control systems
- Automatically generate Terraform configurations from a non-VCS source of data
- Build a variety integrations



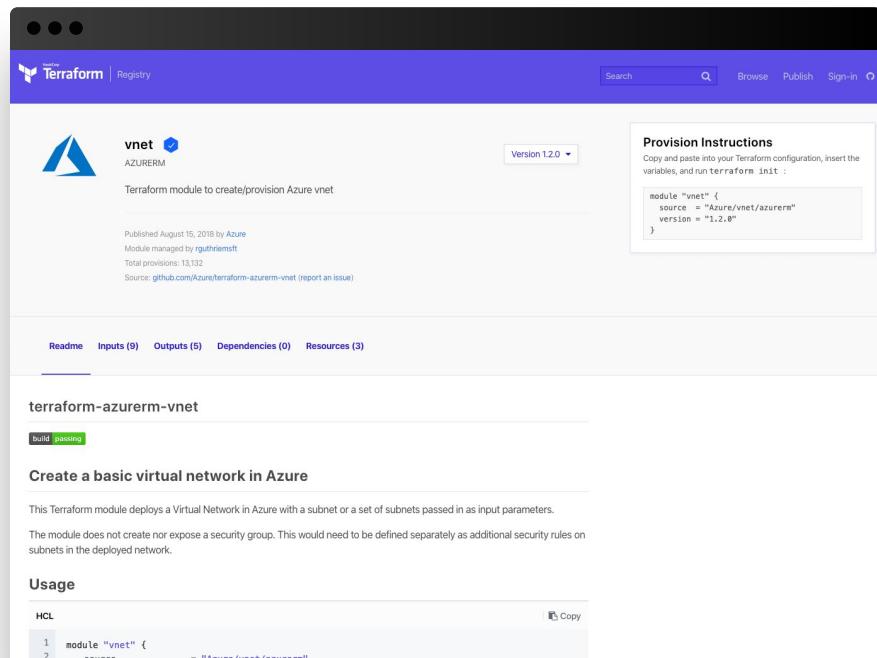
Terraform Cloud Agents

- [Terraform Cloud Agents](#) allow TFCB to communicate with isolated, private, or on-premises infrastructure
- Deployed as lightweight docker container or a binary on x86_64 Linux within a specific network segment
- Useful for on-premises infrastructure types such as vSphere, Nutanix, OpenStack, enterprise networking providers, and anything in a protected enclave
- **The agent architecture is pull-based, so no inbound public internet connectivity is required**
- Agents poll Terraform Cloud for work and carry out execution of that work locally



Private Module Registry

- Terraform modules are a container for multiple cloud resources that are used together
- Modules can be used to create lightweight abstractions, to describe infrastructure in terms of its architecture, rather than directly in terms of specific cloud resources
- The [Private Module Registry](#) (PMR) works similarly to the [public registry](#) and includes support for versioning and a searchable list



Cost Estimation

- TFCB provides cost estimates for many resources found in Terraform configuration
- For each resource an hourly and monthly cost is shown, along with the monthly delta
- The display includes total cost and delta of all estimable resources

The screenshot shows a Terraform Cloud run titled "Canary Test". The run status is "APPLIED" and "Plan finished" 3 minutes ago. It shows 1 resource to add, 0 to change, and 1 to destroy. The "Cost estimation finished" step (BETA) 4 minutes ago estimated 2 of 15 resources at \$1,674.88/mo (+\$1,674.88). A table details the cost for two resources:

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

A warning message indicates 13 of 15 resources couldn't be estimated. The "Policy check passed" step 3 minutes ago showed 1 policy passed, 0 failed. The final step, "Apply finished" 3 minutes ago, resulted in 1 resource to add, 0 to change, and 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

Sentinel is covered in detail later in the program

```
...
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

- Migrating Terraform state to Terraform Cloud allows teams to continue managing infrastructure without de-provisioning anything
- [Migrating to Terraform Cloud or Terraform Enterprise](#)
- [Medium Blog on State Migration via Terraform API](#)



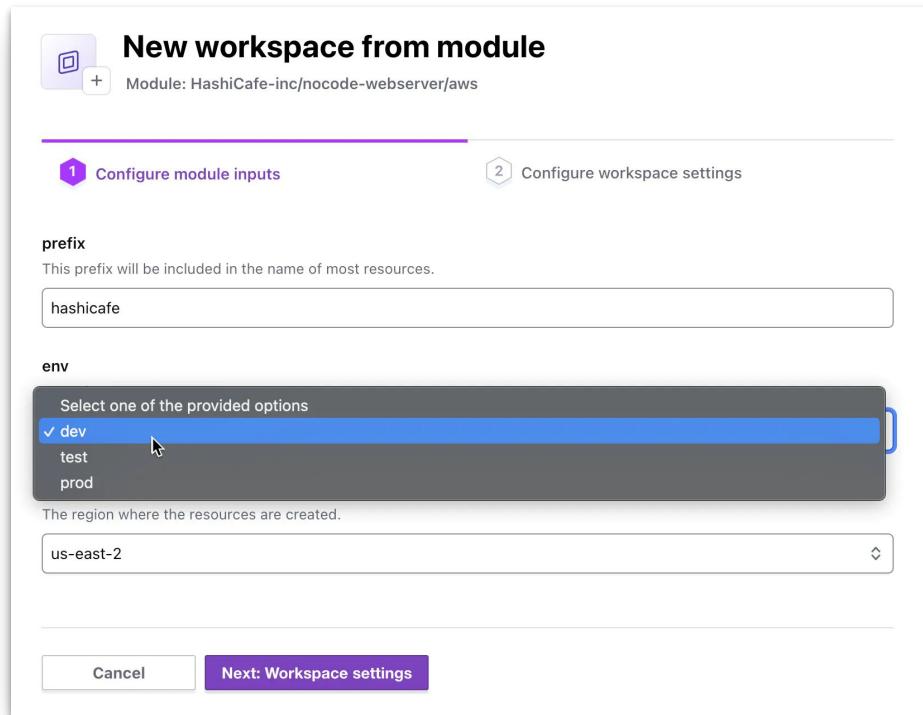
Recent Terraform Additions & Feature Updates



No-Code Provisioning

Deploy cloud resources using Terraform, without learning HCL

Teams can spend less time defining configurations and rebuilding the wheel, and spend more time building off the work of others and supporting the business



Dynamic Provider Credentials

Native just-in-time (JIT) provider authentication

A native solution for JIT access built on top of Terraform Workload Identity and provider OIDC support. Configure dynamic credential injection via workspace variables for:

- Vault
- AWS
- Azure
- Google Cloud

Workspace variables (7)

Variables defined within a workspace always overwrite variables from variable sets that have the same type and the same key. Learn more about variable set precedence [precedence](#).

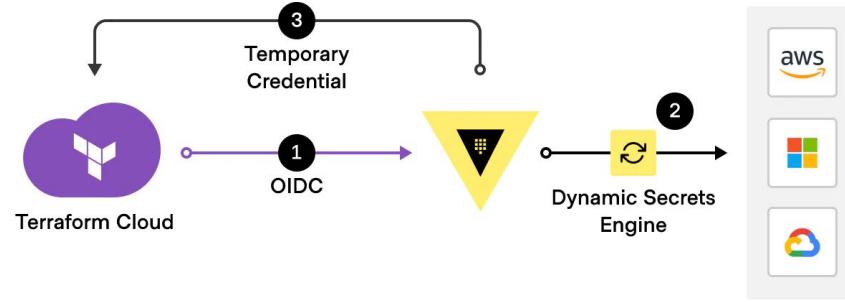
Key	Value	Category	...
TFC_VAULT_PROVIDER_AUTH	true	env	...
TFC_VAULT_RUN_ROLE	tfc-role	env	...
TFC_VAULT_ADDR	https://hashicafe-vault-1111111.hashicorp.cloud:8200	env	...
TFC_VAULT_NAMESPACE	admin	env	...
TFC_AWS_PROVIDER_AUTH	true	env	...
TFC_AWS_PLAN_ROLE_ARN	arn:aws:iam::111111111111:role/tfc_hashicafe_plan_role Read-only role for plans	env	...
TFC_AWS_APPLY_ROLE_ARN	arn:aws:iam::111111111111:role/tfc_hashicafe_apply_role Read/write role for apply	env	...

Dynamic Provider Credentials

Vault-backed Dynamic Credentials

Combine the power of dynamic provider credentials and Vault dynamic secrets engines

- Authenticate Terraform runs to Vault using JWT/OIDC auth method
- Vault generates temporary cloud credentials for AWS, Azure, or Google Cloud
- Secrets are injected into the Terraform agent environment for use with providers
- Credentials are revoked immediately after each run phase



Notable/Recent Additions & Changes

- [Projects](#)
- [Terraform Changelog](#)
- [Run Tasks](#)
- [Terraform 1.2](#)
- [Terraform 1.3](#)
- [Drift Detection](#)



Next Steps



Tutorials

<https://developer.hashicorp.com/terraform/tutorials>

Step-by-step guides to accelerate deployment of Terraform Cloud

The screenshot shows the 'Tutorials' section of the Terraform Cloud developer documentation. The left sidebar lists various categories like 'Get Started' (AWS, Azure, Docker, GCP, OCI), 'Fundamentals' (CLI, Configuration Language, Modules, Provision, State), 'Use Cases' (Terraform Cloud, Applications, AWS Services), and 'Tutorials' (Overview, Terraform Cloud). The main content area displays a breadcrumb path: Developer / Terraform / Tutorials / Terraform Cloud. It features a 'Get Started - Terraform Cloud' section with a brief description and a 'Create an account' link. Below this, three tutorials are listed:

- What is Terraform Cloud - Intro and Sign Up** (5min) - Sign up for Terraform Cloud, which provides free remote state storage, a stable run environment, version control system (VCS) driven plans and applies, a collaborative web GUI, and more. Create your first organization.
- Log in to Terraform Cloud from the CLI** (3min) - Log into Terraform Cloud or Enterprise with the Terraform CLI to migrate state, trigger remote runs, and interact with Terraform Cloud.
- Create a Credentials Variable Set** (3min) - Create a variable set for your AWS IAM credentials that you can reuse across workspaces. Apply the variable set to a workspace.



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>

- [Automation script examples](#)
- [Tutorial: API-driven runs](#)
- [Create a Workspace](#)
- [API-driven Workflow](#)
- [CLI-driven Workflow](#)
- [UI/VCS-driven Workflow](#)



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 ticket for your issue at:

support.hashicorp.com.

Discuss

Engage with the HashiCorp Cloud community including HashiCorp Architects and Engineers

discuss.hashicorp.com



Upcoming Webinars



Terraform Workflow Management

Deep dive into best practices around run workflows, workspaces, variables, modules, and Git repo structure



Terraform Governance

Topics include TFCB's RBAC model and usage, along with using Sentinel, Run Tasks, and Audit logging for governance and guardrails for your teams and infrastructure



Terraform Integrations & Series Closing

Discussion of TFCB integration with Splunk, ServiceNow, & Kubernetes alongside best practices for run triggers & notifications and Production Readiness guidelines

Action Items

- Identify your use case and define your goals with TFCB
- Share to customer.success@hashicorp.com
 - Authorized technical contacts for support
 - Stakeholders contact information (name and email addresses)
 - TFCB Organization Name
- Terraform Cloud Configured
 - Terraform Organization created
 - Terraform workspaces configured to at least 1 workflow (i.e.: API, CLI, VCS, or UI)



05

Q&A





Thank you

customer.success@hashicorp.com

www.hashicorp.com/customer-success