

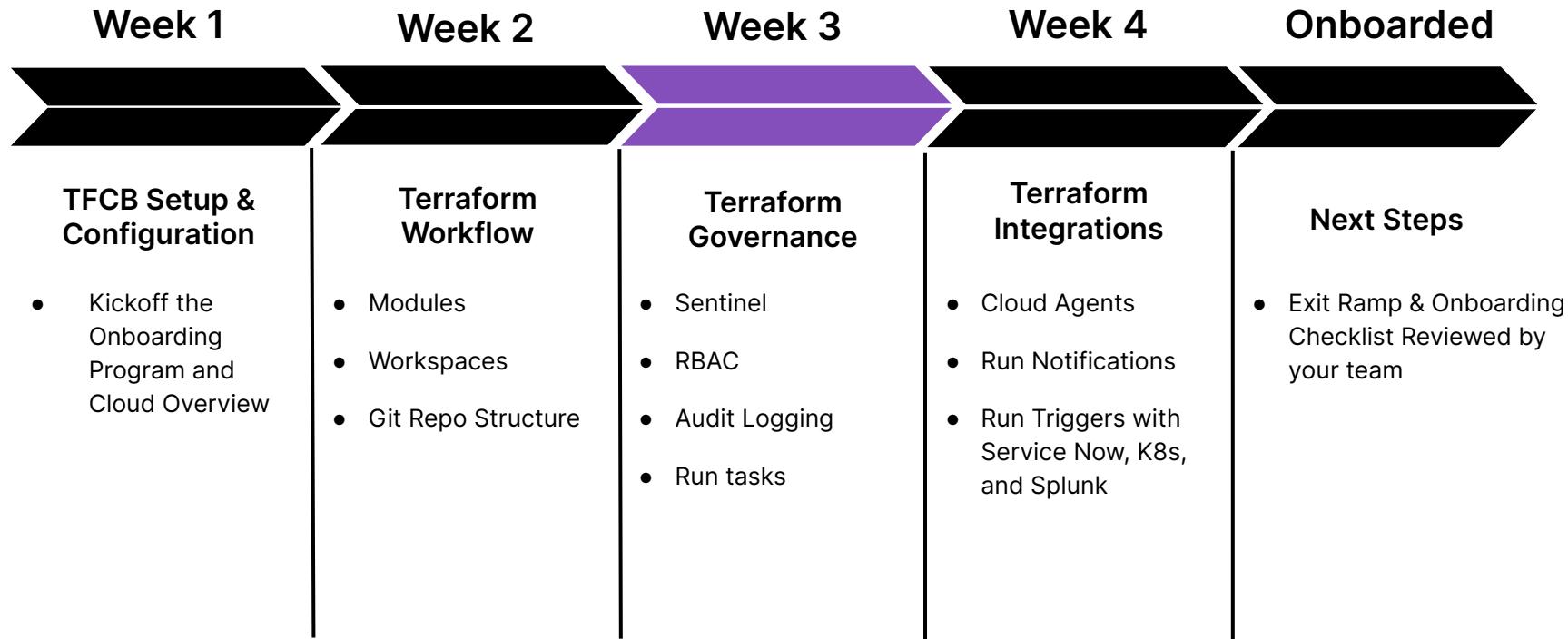


TFCB Onboarding Program Week 3

HashiCorp Customer Success



TFCB Path to Production



Terraform Cloud Governance



Agenda

- | | |
|----------------------------|----|
| Role Based Access Controls | 01 |
| Sentinel | 02 |
| Run Tasks | 03 |
| Audit Logs | 04 |
-
-
-



01

Role Based Access Controls (RBAC)



Terraform Cloud RBAC Model

- Terraform Cloud's (TFCB) access model is team-based
 - Permissions are assigned at the team level
 - Users inherit permissions based upon team assignment
- TFCB's permission model is split into organization-level & workspace-level permissions
- Every Org has an “owners” team which have every available permission in that org
- Workspace permissions allow administrators to delegate access to specific collections of infrastructure



Common Scenarios

- TFC is often used by multiple Teams (i.e. *Developers, QA, Security, Operations, Networking, SQL Admins, Filestore Admins, Accounting*)
- The best approach to managing permissions is:
 - a. Create Groups within your Single Sign-on (SSO) service for each team
 - b. Assign each group as a TFC Team
 - c. Determine how Workspaces will be divided, & assign permissions accordingly
- Data can be dynamically shared between Workspaces as read-only by using the “**tfe_outputs**” data source
- [Terraform_remote_state](#) Data Source

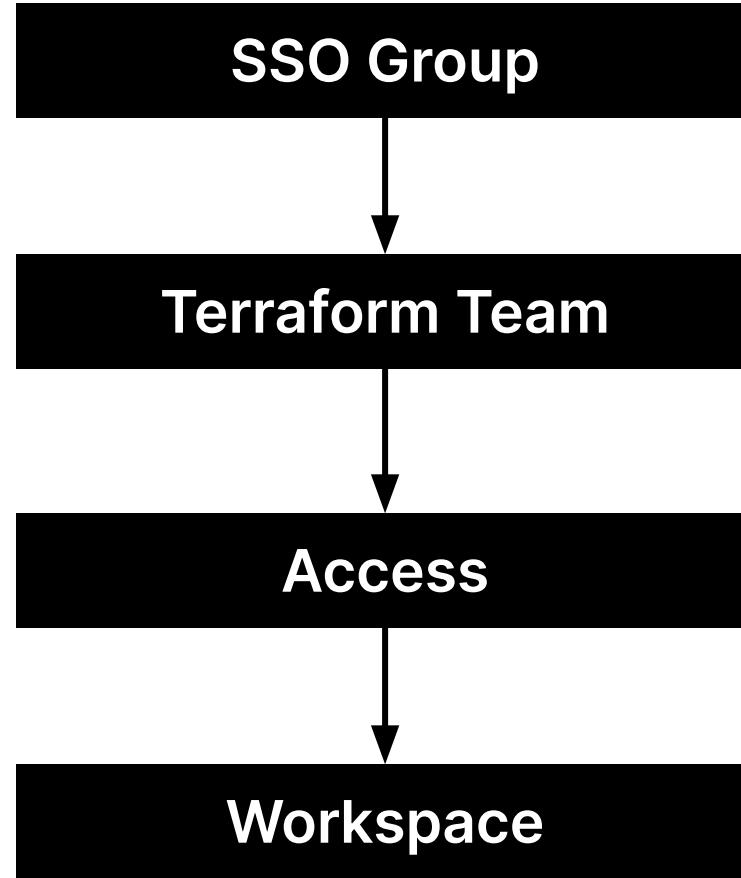




TFCB

Permissions

Flow



Workspace Permissions

There are two ways to [assign permissions](#) to a TFCB team:

1. Custom permissions
2. Fixed permission sets - bundles of specific permissions, designed for delegated access patterns

Permission Sets

Read

- Read runs
- Read variables
- Read state versions

Plan

- Queue plans
- Read variables
- Read state versions

Admin

- VCS Configuration
- Manage Team Access
- Execution Mode
- Delete Workspace
- Read & write workspace settings, general settings, notification configurations, run triggers, & more

Write

- Lock/unlock Workspace
- Download Sentinel mocks
- Read and write Variables
- Read and write State Versions
- Approve Runs





State Files

- May contain secrets, passwords, & API Tokens
- Should be handled as sensitive material when applying RBAC permissions
- Are encrypted at rest using HashiCorp Vault
- Data can still be read at runtime or directly from the TFC UI if a User has the necessary Workspace permissions



my-cool-organization / Workspaces / terraform-tests / Settings / Access

terraform-tests ⓘ

Runs

States

Variables

Settings ▾

Queue plan ▾

Team Access

Add team and permissions

NAME	PRIVILEGES
------	------------

Owners of my-cool-organization default

Policy Managers custom ...

Ops write ...

Edit permissions

Remove team



© 2020 HashiCorp, Inc. [Support](#) [Terms](#) [Privacy](#) [Security](#)





my-cool-organization

Workspaces

Modules

Settings



my-cool-organization / Workspaces / terraform-tests / Settings / Access / Add Team Permissions

terraform-tests

Runs States Variables Settings Queue plan

Add Team Permissions

Add a team and assign permissions to this workspace.

Select a team

Assign permissions

Assign permissions to Security

Assign permissions to the selected team below.



Customize permissions for this team BETA

Read

[Assign permissions](#)

Baseline permissions for reading a workspace

✓ Read runs

✓ Read variables

✓ Read TF config versions

✓ Read workspace information

✓ Read state

Plan

[Assign permissions](#)



terraform-tests

Runs

States

Variables

Settings

Queue plan

Add Team Permissions

Add a team and assign permissions to this workspace.

Select a team

Assign permissions

Assign permissions to Security

Assign permissions to the selected team below.



Customize permissions for this team BETA

Run Permissions

Runs

Read

Can read any general information on the workspace's runs, including logs and the results of policy checks and cost estimates.

Plan

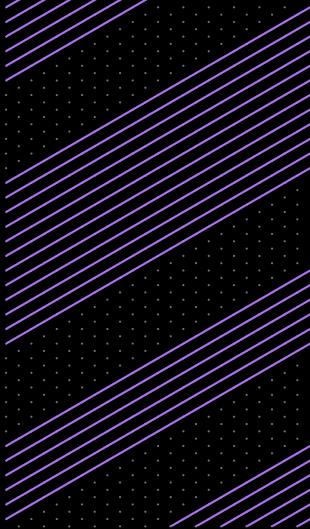
Can queue plans, in addition to all abilities of the read permission.

Apply

Can apply, discard, or cancel runs, in addition to all abilities of the plan permission.



02



Sentinel

**Sentinel is: “Policy,
Governance, & Security
as Code”**



Use Cases

- 1 Cloud Provider
- 2 Account ID
- 3 Limit Availability Zones
- 4 Cost Estimates
- 5 Cost Limiting
- 6 Resource Tagging
- 7 Resource Types
- 8 Resource Sizes
- 9 Resource Configuration
- 10 Resource Destruction





Benefits

1. Enforcement
2. Speed
3. Reproducibility
4. Reliability
5. Automation
6. Version Control
7. Auditability

Architecture

- Variables, conditionals, loops, functions
 - [Sentinel Language Reference](#)
- Validates Config and State (Create, Edit, Destroy) of Terraform resources
- terraform plan → sentinel check → terraform apply
- Enforcement Levels – all are Logged
 - **Hard-mandatory**, required, cannot bypass, fail the TF RUN (prod)
 - **Soft-mandatory**, required, TF Owner can bypass with a comment in the TF UI, will halt the TF Run
 - **Advisory**, guard-rails warning, info warnings in the TF Run



Syntax Example

```
import "units"

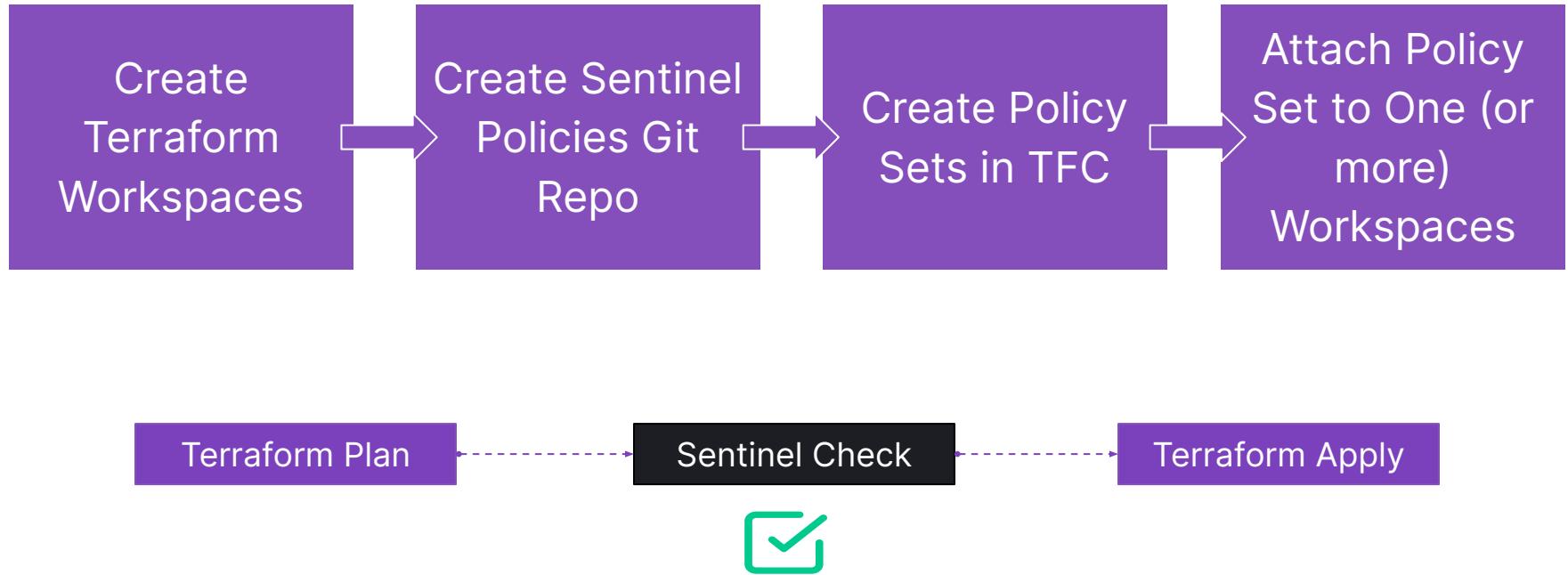
memory = func(job) {
    result = 0
    for job.groups as g {
        for g.tasks as t {
            result += t.resources.memory else 0
        }
    }

    return result
}

main = rule {
    memory(job) < 1 * units.gigabyte
}
```



Workflow



Sentinel Rule Git Repo

 [hashicorp / terraform-sentinel-policies](#) Public

[Watch 58](#) [Fork 20](#) [Star 18](#)

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

 main  1 branch  2 tags [Code](#) [About](#)

Commit	Message	Date
 rberlind	Give pshamus credit	442c23d on 3 Feb
	aws add map_key filers and check-ec2-environment-tag.sentinel	last month
	azure add links to aws, azure, and registry functions docs	2 months ago
	cloud-agnostic add links to aws, azure, and registry functions docs	2 months ago
	common-functions Give pshamus credit	last month
	gcp add gcp-functions module	last month
	vmware remove raw data	2 months ago
	.gitignore remove raw data	2 months ago
	LICENSE Initial commit	2 months ago
	README.md add map_key filers and check-ec2-environment-tag.sentinel	last month

About

Example Sentinel Policies for use with Terraform Cloud and Terraform Enterprise

 [Readme](#)
 [MPL-2.0 License](#)
 [Code of conduct](#)
 [18 stars](#)
 [58 watching](#)
 [20 forks](#)

Releases 2

 [v1.0.1](#) Latest on 1 Feb
+ 1 release



Policy Set File Structure

The screenshot shows a GitHub repository page for `hashicorp/terraform-sentinel-policies`. The repository is public, with 58 stars, 20 forks, and 18 issues. The `Code` tab is selected. The main branch is `main`. The `gcp` directory has been updated multiple times. The latest commit by `rberlind` on Jan 31 at `b3e3977` added a `gcp-functions` module. Other commits include removing raw data from `mocks`, `test`, and several sentinel files (`enforce-mandatory-labels.sentinel`, `restrict-egress-firewall-destination-ranges.sentinel`, etc.). Most of these changes occurred last month or 2 months ago.

Commit	Message	Date
<code>b3e3977</code>	<code>rberlind add gcp-functions module</code>	on 31 Jan
...		
<code>add gcp-functions module</code>		last month
<code>remove raw data</code>		2 months ago
<code>remove raw data</code>		2 months ago
<code>add gcp-functions module</code>		last month
<code>remove raw data</code>		2 months ago
<code>remove raw data</code>		2 months ago
<code>remove raw data</code>		2 months ago
<code>remove raw data</code>		2 months ago
<code>add gcp-functions module</code>		last month

Policy Sets

The screenshot shows the Pyrocumulus Settings interface. At the top, there is a navigation bar with a logo, the text "Pyrocumulus", "Workspaces", "Modules", and "Settings". The "Settings" button is highlighted with a red box. Below the navigation bar, the URL "Pyrocumulus / Settings / Policy Sets" is visible. On the left, there is a sidebar titled "ORGANIZATION SETTINGS" with sections for "Pyrocumulus" (General, Teams, VCS Providers, API Tokens, Authentication, SSH Keys, Cost Estimation, Policies), "Policies" (Policy Sets), and "Policy Sets" (which is also highlighted with a red box). The main content area is titled "Policy Sets" and contains a "Create a new policy set" button. It displays information about a policy set named "pyrocumulus": "1 Workspace · hashicorp/pyrocumulus · 1cd6d65" and "Last updated a month ago".

Pyrocumulus / Settings / Policy Sets

ORGANIZATION SETTINGS

Pyrocumulus

- General
- Teams
- VCS Providers
- API Tokens
- Authentication
- SSH Keys
- Cost Estimation
- Policies

Policy Sets

Create a new policy set

pyrocumulus

1 Workspace · hashicorp/pyrocumulus · 1cd6d65

Last updated a month ago

Create Policy Set

Pyrocumulus ▾ Workspaces Modules Settings Documentation | Status

Pyrocumulus / Settings / Policy Sets / pyrocumulus

ORGANIZATION SETTINGS

Pyrocumulus

- General
- Teams
- VCS Providers
- API Tokens
- Authentication
- SSH Keys
- Cost Estimation
- Policies
- Policy Sets**

Policy Set: pyrocumulus

Last updated September 24th 2019, 2:34:25 pm

Name

You can use letters, numbers, dashes (-) and underscores (_) in your policy set name.

Description

Policy Set Source

 GitHub HashiCorp Github

 Upload via API

 +

 hashicorp/pyrocumulus · 1cd6d65 · Last updated 3 days ago

Attach Policy Set

Scope of Policies

- Policies enforced on all workspaces
- Policies enforced on selected workspaces



Workspaces

The name of the workspace you wish to add to this policy set.

pyrocumulus



—Select item—



Add workspace

Update policy set

Delete policy set

Automate Sentinel to Workspaces

```
# Get a list of Workspace IDs, based on matching a Regex pattern
variable "workspace_name_pattern" {
    type = string
    default = ".*_dev_vdm"
}

data "tfe_workspace_ids" "all" {
    names = ["*"]
    organization = var.tf_org_name
}

output "all_workspace_ids" { value = data.tfe_workspace_ids.all.ids }

locals {
    # filter by the Workspace Name, then return the Workspace ID, or null, then remove null entries
    filtered_workspace_ids = compact(compact([
        for name, id in data.tfe_workspace_ids.all.ids : [
            (length(regexall(var.workspace_name_pattern, name)) > 0) ? id : null
        ]
    ]))
}

output "filtered_workspace_ids" { value = local.filtered_workspace_ids }
```

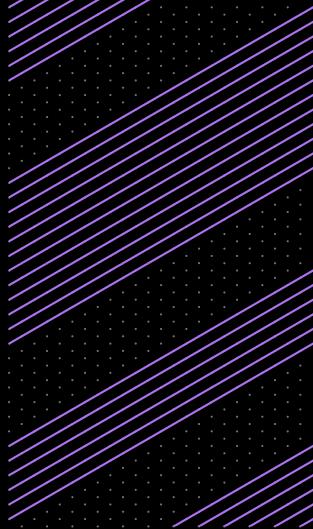




Limitations

1. Can only enforce against resources deployed & managed by Terraform
2. Cannot enforce “self-managed” services (ex: mysql on AWS EC2, Azure VM, GCP VM, VMware VM)
3. Cannot enforce against resource logs / metrics (ex: AWS CloudTrail, Azure Monitor, GCP Cloud Audit Logs)
4. Cannot continuously monitor (ex: AWS Config, Azure Policy, GCP Forseti)
5. Sentinel uses the Cloud Provider’s Cost Estimation API, which doesn’t continuously run, & does not check costs for usage-based billing (ex: AWS Athena, Azure DataBricks, GCP BigQuery, GCP Pub/Sub)

03



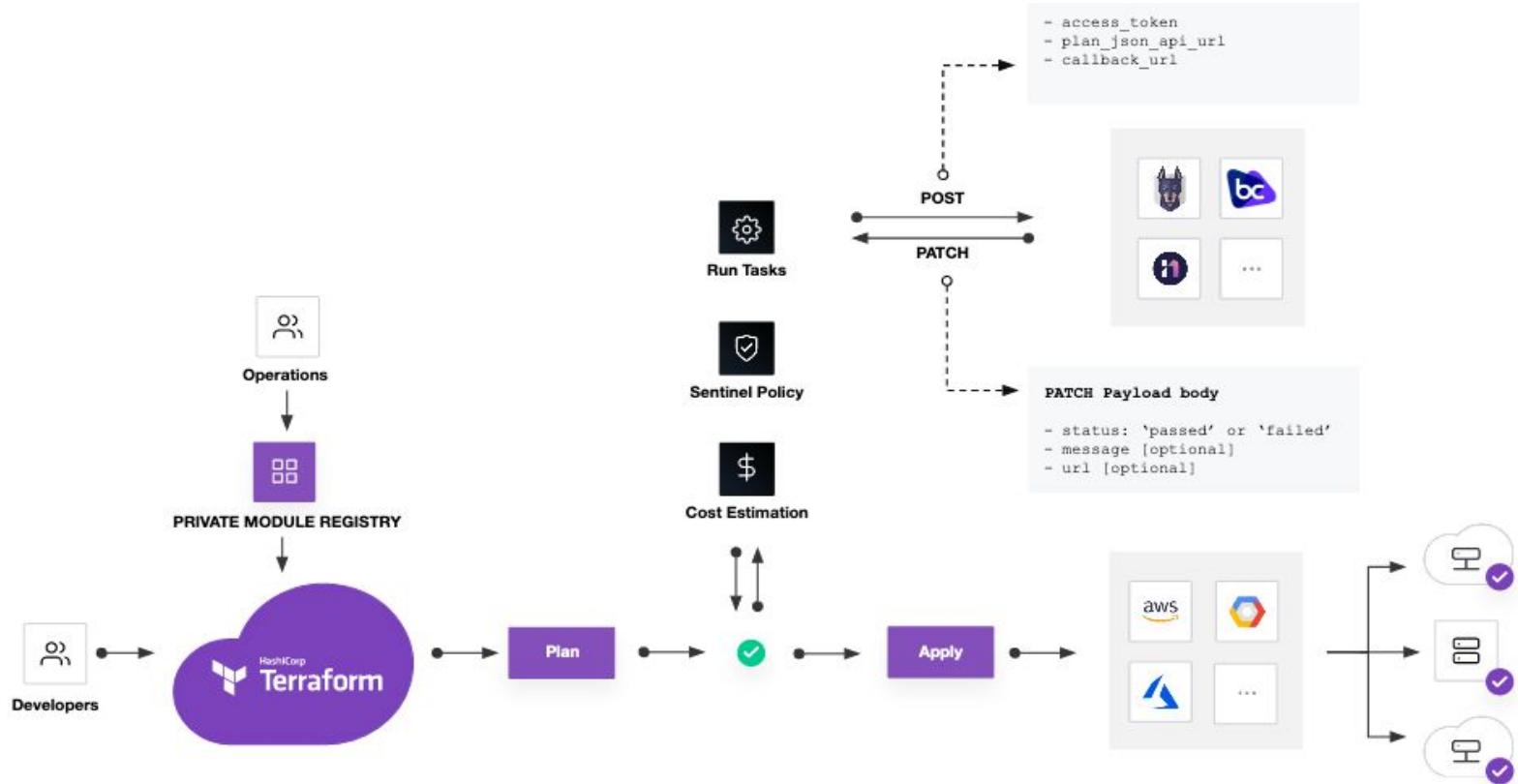
Run Tasks

Run Tasks

- Integrate 3rd-party tools into the pre-apply stage during a Terraform Cloud run
 - During pre-apply phase, an event hook is triggered & TFCB sends a payload containing run details
 - Terraform waits for the service to reply with either passed or failed status
- Supported integrations
 - Snyk, Bridgecrew, Infracost, Lightlytics, Vantage
 - HCP Packer



Architecture of Run Tasks



Payload from Terraform

```
...
{
  "payload_version": 1,
  "access_token": "4QEuyyxug1f2rw.atlasv1.iDyxqhXGVZ0ykes53YdQyHyYtF0rdAWNxcVUgWvzb64NFhjcquu8gJMEduoSLRu4Q",
  "task_result_id": "taskrs-2nH5dncYoXaMVQmJ",
  "task_result_enforcement_level": "mandatory",
  "task_result_callback_url":
    "https://app.terraform.io/api/v2/task-results/5ea8d46c-2ceb-42cd-83f2-82e54697bdd/callback",
  "run_app_url": "https://app.terraform.io/app/hashicorp/my-workspace/runs/run-i3Df5to9ELvibKpQ",
  "run_id": "run-i3Df5to9ELvibKpQ",
  "run_message": "Triggered via UI",
  "run_created_at": "2021-09-02T14:47:13.036Z",
  "run_created_by": "username",
  "workspace_id": "ws-ck4G5bb1Yei5szRh",
  "workspace_name": "tfri_github_0",
  "workspace_app_url": "https://app.terraform.io/app/hashicorp/my-workspace",
  "organization_name": "hashicorp",
  "plan_json_api_url": "https://app.terraform.io/api/v2/plans/plan-6AFmRJW1PFJ7qbAh/json-output",
  "vcs_repo_url": "https://github.com/hashicorp/terraform-random",
  "vcs_branch": "main",
  "vcs_pull_request_url": null,
  "vcs_commit_url":
    "https://github.com/hashicorp/terraform-random/commit/7d8fb2a2d601edebdb7a59ad2088a96673637d22"
}
```

CODE EDITOR

Create Run Tasks

Organization Settings → Run Tasks → Create run tasks

The screenshot shows the HashiCorp Cloud Platform interface. The top navigation bar includes the HashiCorp logo, user account, and various settings links like Workspaces, Registry, Usage, and Settings. The Settings link is highlighted. Below the navigation is a breadcrumb trail: sandraliu-tam / Settings / Run Tasks. The left sidebar contains a list of organization settings: General, Tags, Teams, Users, Variable sets, Integrations, Cost estimation, Policies, Policy sets, and Run tasks, which is currently selected and highlighted in blue. The main content area is titled "Run Tasks" and features a "Create run task" button. A message states: "Directly integrate third-party tools and services to manage cost, security, compliance and more. Or enhance your workflow with custom logic. [Learn more about run tasks.](#)" Below this, a large box says "No run tasks yet." and provides a brief description: "Run Tasks allow you to integrate third-party tools and services directly in a Terraform run." At the bottom, there's a section titled "Partner Integration Guides" with cards for Bridgecrew, Infracost, Lightlytics, and Snyk, each with a brief description and a "View" button.

Organization settings

General

Tags

Teams

Users

Variable sets

Integrations

Cost estimation

Policies

Policy sets

Run tasks

Security

Agents

API tokens

Authentication

SSH keys

Run Tasks

[Create run task](#)

Directly integrate third-party tools and services to manage cost, security, compliance and more. Or enhance your workflow with custom logic. [Learn more about run tasks.](#)

No run tasks yet.

Run Tasks allow you to integrate third-party tools and services directly in a Terraform run.

Partner Integration Guides

Get started with one of our partners' purpose-built run task integrations.

Bridgecrew
Security and compliance visibility streamlined for Terraform. [View](#)

Infracost
Cloud cost estimation and forecasting for Terraform. [View](#)

Lightlytics
Gain unmatched visibility and control across your workflow. [View](#)

Snyk
Find, prioritize, & fix security vulnerabilities in Terraform. [View](#)

Run Task Integration with HCP Packer

Run Task validates that machine images in your Terraform configuration are valid and haven't been revoked for security or other reasons

Use-cases

1. Use run tasks with HCP Packer to identify compromised images with Terraform Cloud to prevent images from being outdated
2. Enforce image compliance with Terraform Cloud and let your configuration dynamically use more up to date images as you create them



Create Run Tasks with HCP Packer

Organization Settings → Run Tasks → Create run tasks

The screenshot shows the HashiCorp Cloud Platform (HCP) interface. The top navigation bar includes the HCP logo, user name 'sandraliu-tam', workspace dropdown, Registry, Usage, Settings (selected), HashiCorp Cloud Platform, and a search bar. Below the navigation is a breadcrumb trail: sandraliu-tam / Settings / Run Tasks / Edit.

The left sidebar lists organization settings categories: General, Tags, Teams, Users, Variable sets, Integrations, Cost estimation, Policies, Policy sets, and Run tasks (which is highlighted with a red box). Other categories like Security, Agents, API tokens, Authentication, SSH keys, and SSO are also listed.

The main content area is titled 'Run Task: HCP_Packer'. It contains a brief description: 'Directly integrate third-party tools and services to manage cost, security, compliance and more. Or enhance your workflow with custom logic.' Below this is a task entry: 'task-X1zmnFYtjjE32nk2' with a link icon.

The task configuration includes:

- Enabled:** A checked checkbox with the note: 'Run task will run across all associated workspaces.'
- Name:** 'HCP_Packer' (input field)
- Endpoint URL:** 'https://api.cloud.hashicorp.com/packer/2021-04-30/terraform-cloud/validation/5424f747-8772-4a10-9014-3a03a4cf608e' (input field, highlighted with a red box)
- Description (optional):** 'e.g. A description looks like this' (input field)
- HMAC key (optional):** '89a0d40fc416bc66fef30e983dc94f33277cdb485c34a7e4c485d44ad63ffa9' (input field, highlighted with a red box)

A note below the HMAC key field states: 'A secret key that may be required by the service to verify request authenticity.'

At the bottom is a blue 'Save run task' button.

Technology Partners

Bridgecrew

Security and compliance errors in Terraform configurations.

cloudtamer.io

Cost savings or compliance findings.

Infracost

A cloud infrastructure costing, initiated right from a pull request or Terraform run.

Lightlytics

Security checks to any additional dependency changes.

Refactr

Allows users to build workflows for multiple use cases including but not limited to code scanning.

Snyk

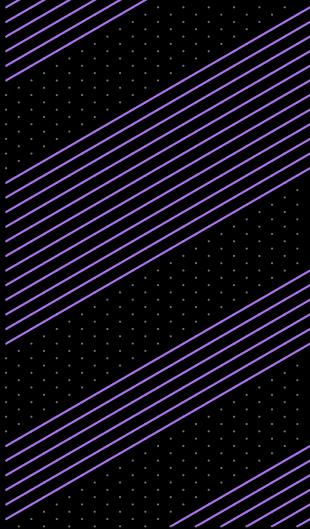
find, track, and fix security misconfigurations in their cloud infrastructure as part of their SDLC

Future

The up-to-date list is available [here](#)



04



Audit Logs

Audit Logging

- The audit trails API exposes a stream of audit events, describing changes to the application entities (workspaces, runs, etc.) for a specific TCFB Organization
- Terraform Cloud retains 14 days of audit log information
- Retention beyond 14 days requires ingestion into an external platform or solution
- Endpoint requires an organization token, user token or team token will not work



Next Steps



Tutorials

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

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

The screenshot shows the Terraform Cloud Tutorials page. The navigation bar includes links for Home, Documentation, Tutorials, Install, Registry, Try Cloud, Search, and a user profile icon. The main content area shows the 'Tutorials' sidebar with sections for Overview, Get Started (AWS, Azure, Docker, GCP, OCI), Fundamentals (CLI, Configuration Language, Modules), and Terraform Cloud. The 'Terraform Cloud' section is highlighted with a grey background. The main content area displays the 'Get Started - Terraform Cloud' track, which is described as a track for collaborating on version-controlled configuration using Terraform Cloud. It includes a 'Start' button and a note that there are 10 tutorials. One tutorial card is visible, titled 'What is Terraform Cloud - Intro and Sign Up', which describes signing up for Terraform Cloud to use its features like remote state storage and a stable run environment. Another partially visible tutorial card is titled 'Log in to Terraform Cloud from the CLI'.



Resources

Sentinel Policies

- [Example Sentinel Policies Collection](#)
- [Terraform Foundational Policies Library](#)
- [Sentinel Tutorials](#)

Run Tasks

- [Run Tasks Documentation](#)
- [Run Tasks Integration](#)
- [Tutorial: Configure Snyk Run Task in Terraform Cloud](#)

Audit Logging

- [Audit Trails API](#)
- [Blog - Cloud Compliance & Management with Terraform](#)
- [Log Forwarding](#)
- [Terraform Cloud Audit Logging with Splunk](#)
- [Medium Post - Splunk Integration with TFC](#)



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 Integrations/ Program Closing

Deep dive into Cloud Agents, Integration with K8s, ServiceNow, and Splunk, Run Triggers, Run Notifications, if your infrastructure is ready for production, and close out the program.



Q&A





Thank you

customer.success@hashicorp.com

www.hashicorp.com/customer-success