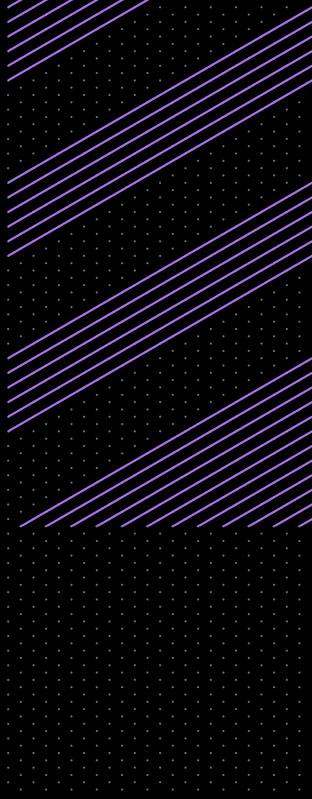
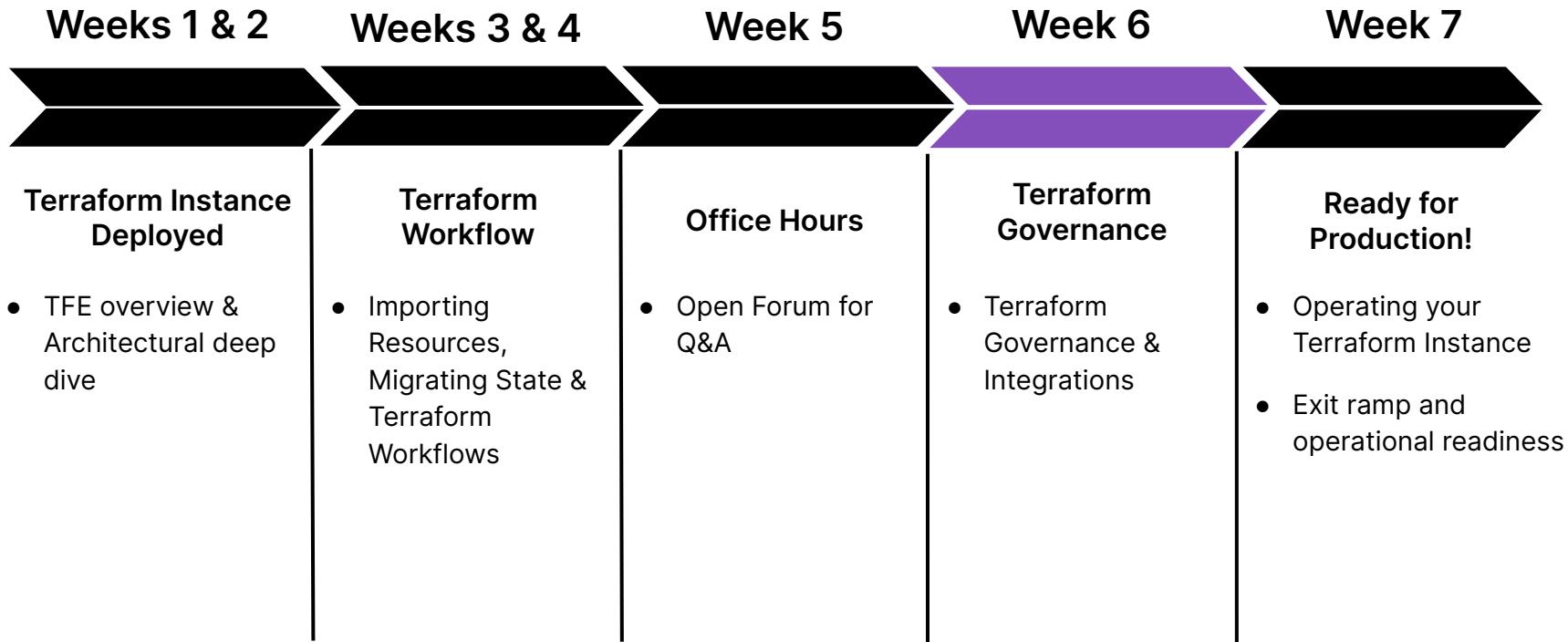


Terraform Governance & Integrations



TFE Path to Production



Agenda

- | | |
|----------------------------------|----|
| Role Based Access Controls | 01 |
| Cloud Agents | 02 |
| Sentinel | 03 |
| TFE & ServiceNow | 04 |
| Run Triggers & Run Notifications | 05 |
-
-
-
-



01

Role Based Access Controls (RBAC)



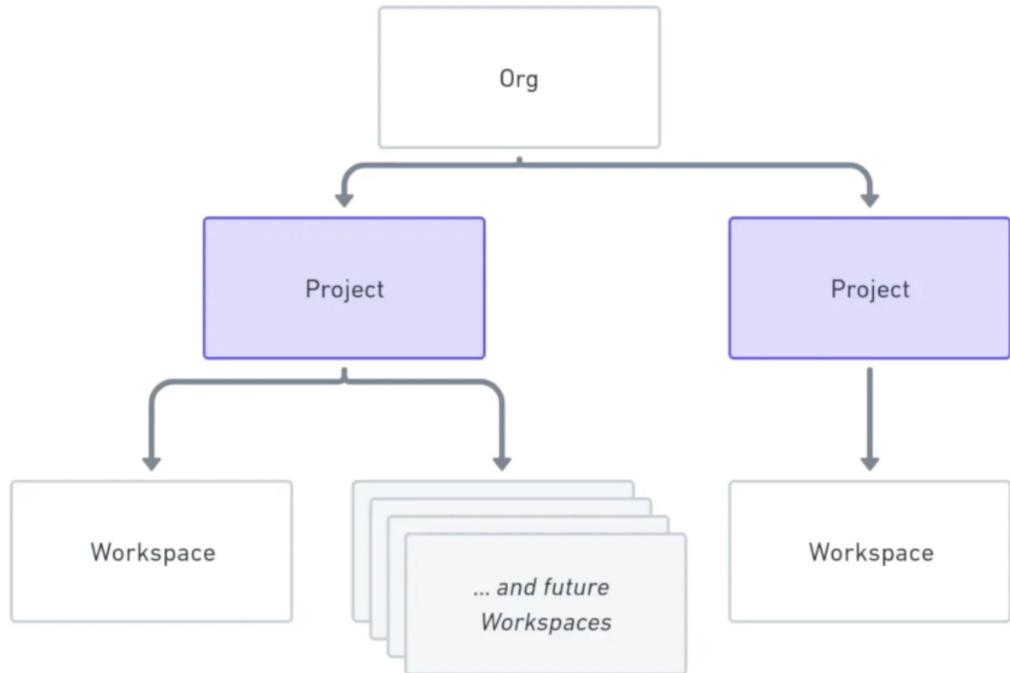
Terraform Enterprise RBAC Model

- Terraform Enterprise's (TFE) access model is team-based
 - Permissions are assigned at the team level
 - Users inherit permissions based upon team assignment
- TFE'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



Workspaces and 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)



Common Scenarios

- TFE 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

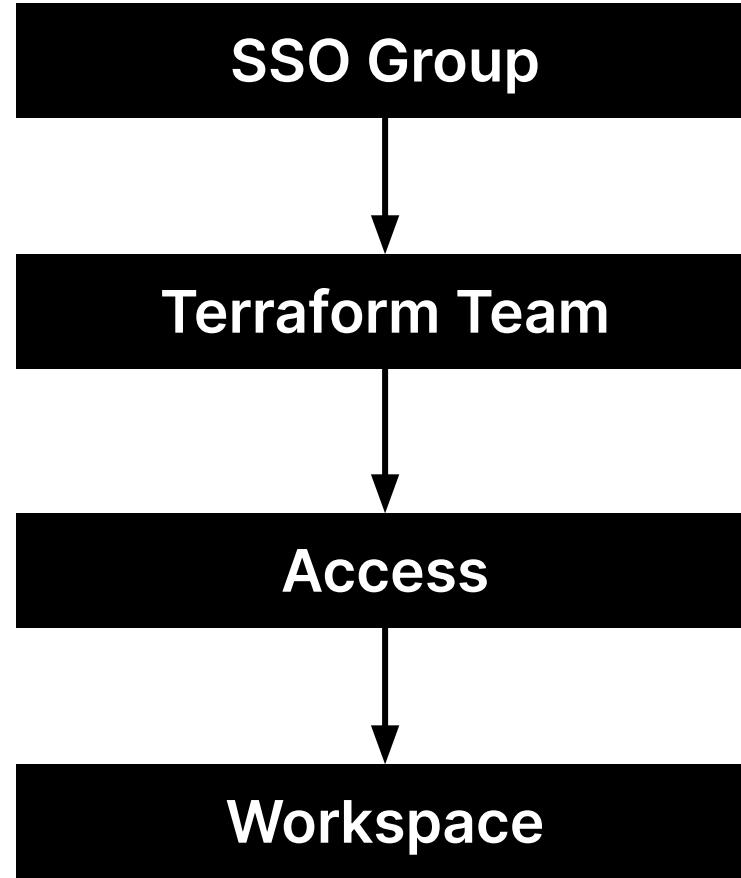




TFCB

Permissions

Flow



Workspace & Project Permissions

There are two ways to assign permissions to a TFE team:

1. Custom permissions

The screenshot shows the 'Add Team Permissions' page for a workspace named 'hashicat-aws'. At the top, there's a summary bar with the workspace name, ID, resources (0), Terraform version (1.2.5), and last update (9 months ago). Below this is a note about workspace descriptions and an 'Unlocked' status. The main section is titled 'Add Team Permissions' and contains two tabs: 'Select a team' (selected) and 'Assign permissions'. Under 'Assign permissions to admins', there's a note to assign permissions to the selected team. A red arrow points to the 'Customize permissions for this team' checkbox. The 'Run Permissions' section lists 'Read' (selected) and 'Plan' options. The 'Read' option is described as reading general workspace run information. The 'Plan' option is described as queuing plans and commenting on runs. The 'Apply' option is described as applying, discarding, or canceling runs. At the bottom, there are 'Other controls' like 'Lock/unlock workspace'.

2. Fixed permission sets

The screenshot shows the 'Add Team Permissions' page for the same workspace 'hashicat-aws'. The layout is identical to the first screenshot, but the 'Assign permissions' tab is selected. This tab displays a list of pre-defined permission sets: 'Read', 'Plan', and 'Apply'. Each set has a detailed description and an 'Assign permissions' button. The 'Read' set includes 'Baseline permissions for reading a workspace' with options like 'Read runs', 'Read workspace information', 'Read variables', 'Read state', and 'Read TF config versions'. The 'Plan' set includes 'Baseline permissions for writing a workspace' with options like 'Write runs', 'Write workspace information', 'Write variables', 'Write state', and 'Write TF config versions'.





Workspace & Project Permissions

- Each workspace has an “admin” permissions level with full control of the workspace
- Projects have specific permissions that can be assigned to teams
- Members of teams with "admin" permissions for a Project have permissions for every workspace in the project & additional permissions

Workspace 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 TFE UI if a User has the necessary Workspace permissions

A screenshot of the HashiCorp Terraform Cloud interface, specifically the 'Access' page for a workspace named 'hashicat-aws'. The left sidebar shows navigation options like 'Workspace Settings', 'General', 'Health', 'Locking', 'Notifications', 'Policies', 'Run Tasks', 'Run Triggers', 'SSH Keys', 'Team Access' (which is highlighted with a red box), and 'Destruction and Deletion'. The main content area displays the workspace details: ID: ws-58Trc2hhBJ5rbWse, Resources: 0, Terraform version: 1.2.5, Updated: 9 months ago. It also shows a status message 'No workspace description available. [Add](#) workspace description.' and a lock icon indicating it's 'Unlocked'. A blue 'Actions' button is visible. The 'Team Access' section has a heading 'Team Access' and a 'Add team and permissions' button, which is also highlighted with a red box. A note below states: 'Teams with [project-level](#) or [organization-level](#) permissions can also access this workspace, even if they are not listed on this page or are listed at a lower access level.' A table header for 'Name' and 'Privileges' is shown, but no data is present. Pagination controls and an 'Items per page' dropdown are at the bottom.

hashicorp-jennawong / Projects & workspaces / hashicat-aws / Settings / Access

hashicat-aws

ID: ws-58Trc2hhBJ5rbWse [Edit](#)

No workspace description available. [Add](#) workspace description.

Unlocked

[Actions](#)

Team Access

[Add team and permissions](#)

Heads up

Teams with [project-level](#) or [organization-level](#) permissions can also access this workspace, even if they are not listed on this page or are listed at a lower access level.

Name	Privileges
1–0 of 0	< 1 >

Items per page [50](#)

hashicorp-jennawong / Projects & workspaces / hashicat-aws / Settings / Access / Add Team Permissions

hashicat-aws
ID: ws-58Trc2hhBJ5rbWse 

No workspace description available. [Add workspace description.](#)

 Unlocked

[Actions](#) 

Add Team Permissions

Add a team and assign permissions to this workspace.

 Select a team  2 Assign permissions

Assign permissions to admins

Assign permissions to the selected team below.

Customize permissions for this team

Read

Baseline permissions for reading a workspace

<input checked="" type="checkbox"/> Read runs	<input checked="" type="checkbox"/> Read variables	<input checked="" type="checkbox"/> Read TF config versions
<input checked="" type="checkbox"/> Read workspace information	<input checked="" type="checkbox"/> Read state	

[Assign permissions](#)

Plan

[Assign permissions](#)



hashicat-aws

ID: ws-58Trc2hhBJ5rbWse

No workspace description available. [Add workspace description](#).

Unlocked

Resources

0

Terraform version

1.2.5

Updated

9 months ago

[Actions](#)

Add Team Permissions

Add a team and assign permissions to this workspace.

Select a team

Assign permissions

Assign permissions to admins

Assign permissions to the selected team below.

Customize permissions for this team



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 and comment on runs, 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.

Other controls

Lock/unlock workspace

Can manually lock and unlock the workspace. This permission is required when the workspace [execution mode](#) is set to



Team API Token

- A Team can only have a single API token
- The token can be regenerated
- The token capabilities is defined by the roles and permissions of the team
- The token can be used both by the CLI and API
- The token will bypass SSO and MFA
- Tokens can also be generated at Organisation and User level



Admin Roles

Super User Roles

Manage Policies

Create, edit and delete
Sentinel Policy Sets

Manage Workspaces

Create and administrate
all workspaces in the
organisation

Manage VCS Settings

Create and manage
VCS settings and SSH
Keys

Manage Policy Overrides

Override
'soft-mandatory' policy
checks



02

Cloud Agents



Terraform Cloud Agents

- x86-based Golang binary
- Only outbound connectivity required
- Communicate with isolated, private infrastructure, such as vSphere, Nutanix, OpenStack, or across multiple cloud accounts
- Can be hosted close to target infrastructure
- Deployable on bare metal, a VM, as a Docker container, or in a Kubernetes cluster



Requirements

Supported Platforms

- Baremetal
- Docker
- Kubernetes (K8S)
- VMware VM
- AWS EC2 VM, EKS, ECS, Fargate EKS, Fargate ECS
- Azure VM, Container Service, AKS
- GCP Compute Engine VM, GKE

Hardware Requirements

- x86-based Linux host
- 2 GB of RAM
- 4 GB of disk space

Networking Requirements

- TFE host via HTTPS (443)
- releases.hashicorp.com
- registry.terraform.io
- (Airgapped) if custom TF CLI binary is used external access is not needed.





Terraform Cloud Agents

- No restriction on Agent or Agent Pool Count
- Agents must define the TFE hostname via either:
 - -address CLI flag
 - TFC_ADDRESS environment variable
- Agents support custom Terraform bundles
- Must be able to communicate with the TFE instance via **HTTPS**
- There are some restrictions on what versions of Agents can be registered

Terraform Cloud Agents

- Configured as an agent pool with many agents
- A token is generated and applied using a command line or ENV VAR and connects back to TFE
- Once an agent is in a pool, the pool can be assigned to a Workspace

The screenshot shows the 'Tokens' section of the Terraform Cloud interface. It displays a table with one row for 'agent1'. The columns are 'Token description', 'Created', 'Last used', and 'Revoke Token'. Below the table, there's a button '+ New token' and a section titled 'Token created' containing a warning message about the token being displayed and instructions for saving it.

Tokens

Token description	Created	Last used	
agent1	a few seconds ago	never	Revoke Token

+ New token

Token created

Your new agent token, **agent1**, is displayed below.

P2gfq1JMpekzCw.atlasv1.P28yTtE8scce5T1DcZ5Tkne3K98VJ6w1kz7g79cIZZxW0953BVhCoNlqvNGJu0v14 [Copy](#)

⚠ Warning
This token **will not be displayed again**, so make sure to save it to a safe place.

Set up your agents

Connect to your Docker host and set the following environment variables. `TFC_AGENT_NAME` is optional.

```
$ export TFC_AGENT_TOKEN=  
$ export TFC_AGENT_NAME=<my_agent_name>
```

Once the environment is configured, run the Docker container with the following command or [download the agent file](#).

```
$ docker run --env TFC_AGENT_TOKEN --env TFC_AGENT_NAME hashicorp/tfc-agent:latest
```

[Read more in our documentation](#) [Copy](#)

[Cancel](#) [Finish](#)

The screenshot shows the 'Execution Mode' configuration page. It has three options: 'Remote' (selected), 'Local', and 'Agent'. The 'Remote' option is described as running plans and applies on Terraform Cloud's infrastructure. The 'Local' option is described as running on machines controlled by the user. The 'Agent' option is described as Terraform Cloud managing the agent's execution. Below the modes, there's a section for 'Agent pool' with a dropdown menu showing 'education' and an ID field with 'ID apool-Mh6n58axtCR2G8ms'.

Execution Mode

If you change the execution mode any in progress runs will be discarded.

Remote

Your plans and applies occur on Terraform Cloud's infrastructure. You and your team have the ability to review and collaborate on runs within the app.

Local

Your plans and applies occur on machines you control. Terraform Cloud is only used to store and synchronize state.

Agent

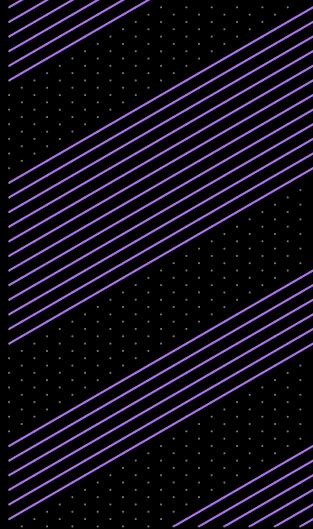
Terraform Cloud will manage the plans and applies your agents execute.

Agent pool

education [▼](#)

ID apool-Mh6n58axtCR2G8ms [Copy](#)

03



Sentinel

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





Benefits

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

Sentinel

- Has its own [language](#) that includes variables, loops, imports, conditionals, and functions
- [Modules](#) are useful for creating reusable code and libraries
- Ensures governance is applied automatically rather than relying on manual auditing
- Supports fine-grained policies using conditional logic
- Allows you to write complex logic and even call cost estimation



Sentinel

- Runs after a terraform plan and before a terraform apply
 terraform plan → sentinel check → terraform apply
- Enforcement levels:
 - Advisory: 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
 - Hard mandatory: guard-rails warning, info warnings in the TF Run
- Includes a CLI tool to allow fast policy tests and runs
- Validates Config and State (Create, Edit, Destroy) of Terraform resources
- Foundational Policies Library of premade policies is available

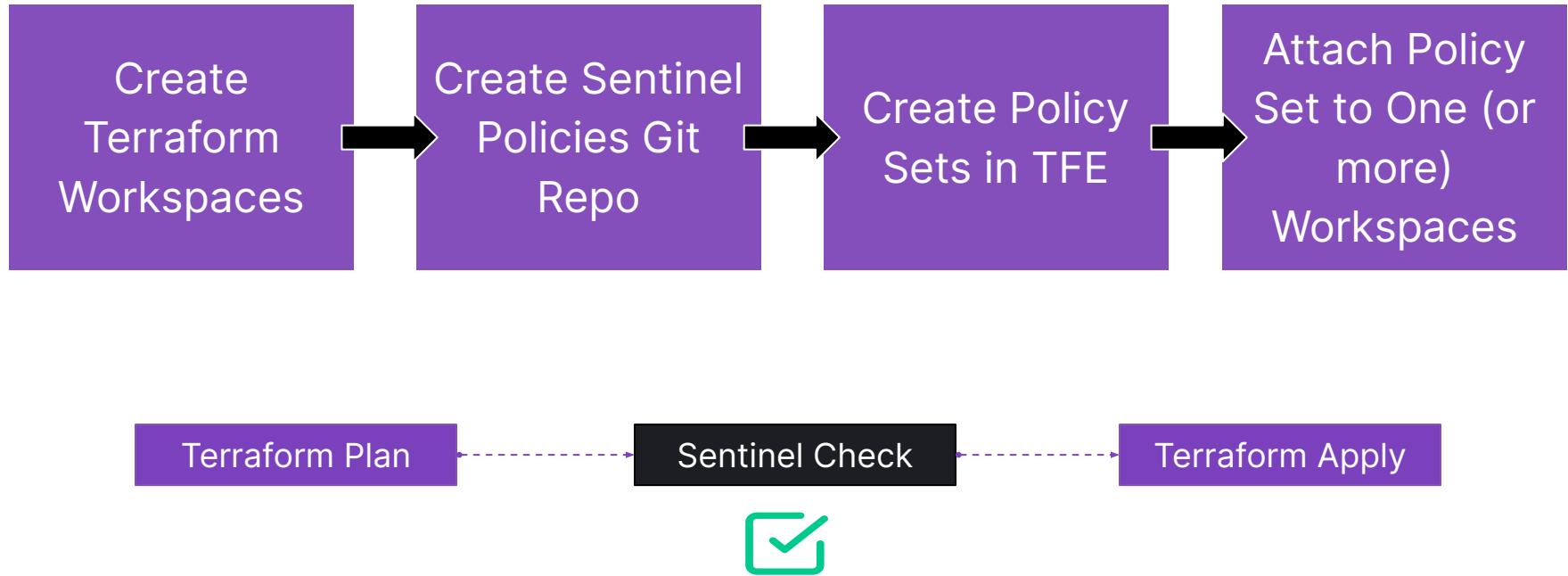


Use Cases

- 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



Workflow



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
}
```

Sentinel Rule Git Repo

The screenshot shows the GitHub repository page for `hashicorp/terraform-sentinel-policies`. The repository is public and has 58 watchers, 20 forks, and 18 stars. The `Code` tab is selected. The main content area displays a list of commits from the `main` branch. The commits are as follows:

Author	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

The right sidebar contains sections for `About`, `Readme`, `MPL-2.0 License`, `Code of conduct`, `18 stars`, `58 watching`, `20 forks`, `Releases` (with one latest release at v1.0.1), and `+ 1 release`.



Policy Set File Structure

The screenshot shows a GitHub repository page for `hashicorp/terraform-sentinel-policies`. The repository is public and has 58 watches, 20 forks, and 18 stars. The `Code` tab is selected. A dropdown menu shows the `main` branch is selected. Below the navigation bar, there's a breadcrumb trail: `terrafo... / gcp /`. On the right, there are buttons for `Go to file`, `Add file`, and three dots.

A single commit is shown:

Author	Message	Date	History
rberlind	add gcp-functions module	b3e3977 on 31 Jan	History

Below the commit, a list of files and their descriptions is shown:

- `gcp-functions`: add gcp-functions module (last month)
- `mocks`: remove raw data (2 months ago)
- `test`: remove raw data (2 months ago)
- `enforce-mandatory-labels.sentinel`: add gcp-functions module (last month)
- `restrict-egress-firewall-destination-ranges.sentinel`: remove raw data (2 months ago)
- `restrict-gce-machine-type.sentinel`: remove raw data (2 months ago)
- `restrict-gke-clusters.sentinel`: remove raw data (2 months ago)
- `restrict-ingress-firewall-source-ranges.sentinel`: remove raw data (2 months ago)
- `sentinel.hcl`: add gcp-functions module (last month)

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(flatten([  
    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)

04

Terraform & ServiceNow Integration

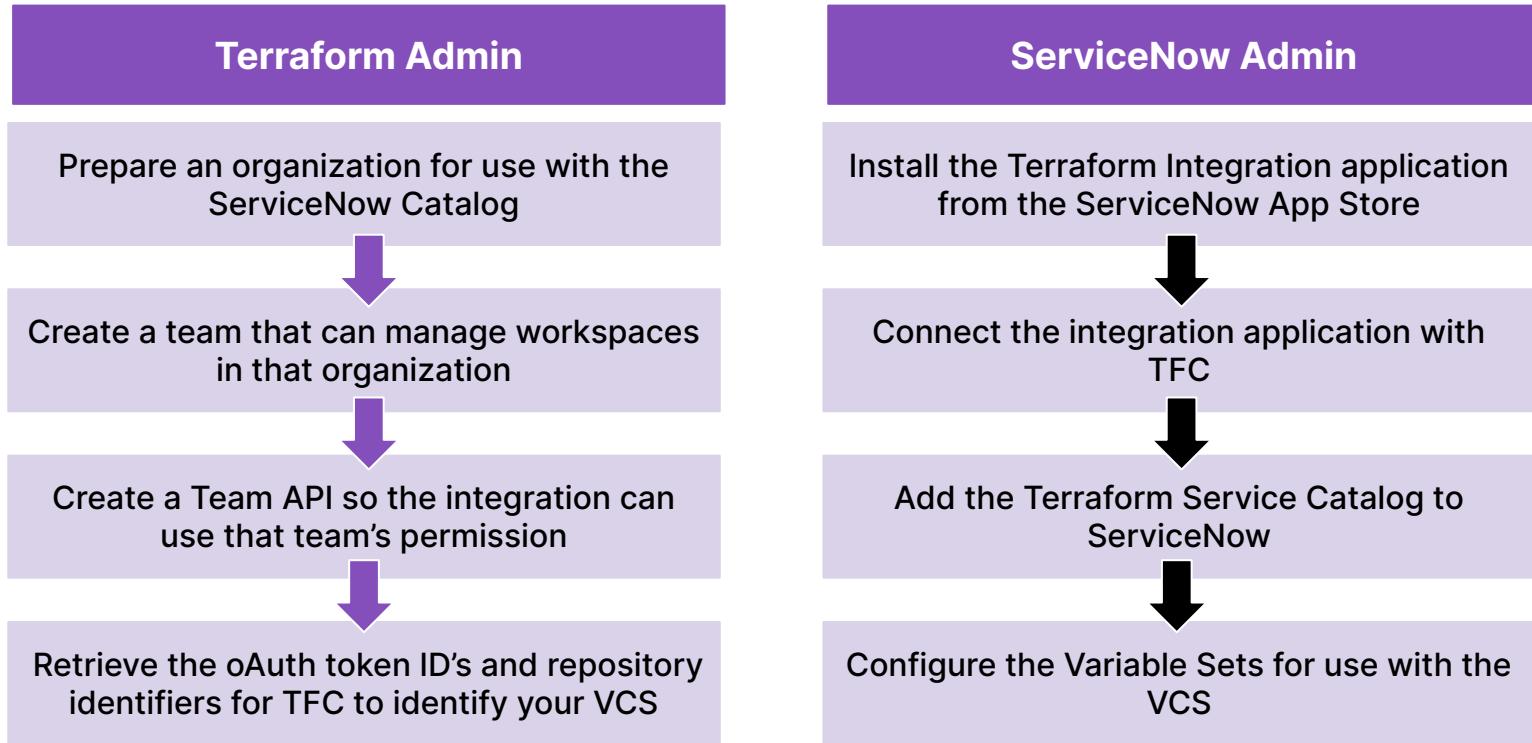


Terraform Integration with ServiceNow

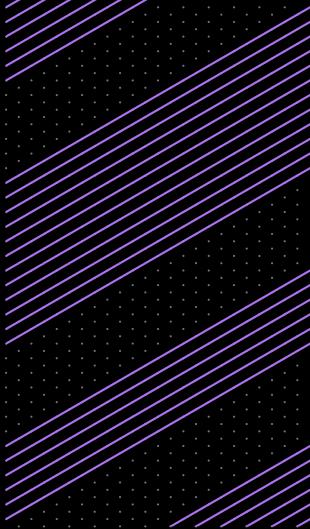
- The Terraform ServiceNow Service Catalog integration enables end-users to provision self-serve infrastructure via ServiceNow
- Connecting ServiceNow to Terraform Enterprise lets users:
 - Order Service Items
 - Create workspaces
 - Perform Terraform runs using prepared Terraform configurations hosted in VCS repositories
- [Terraform ServiceNow Service Catalog Integration Setup](#)
- [Terraform ServiceNow Integration Administrator Guide](#)



Workflow



05



Run Triggers & Run Notifications

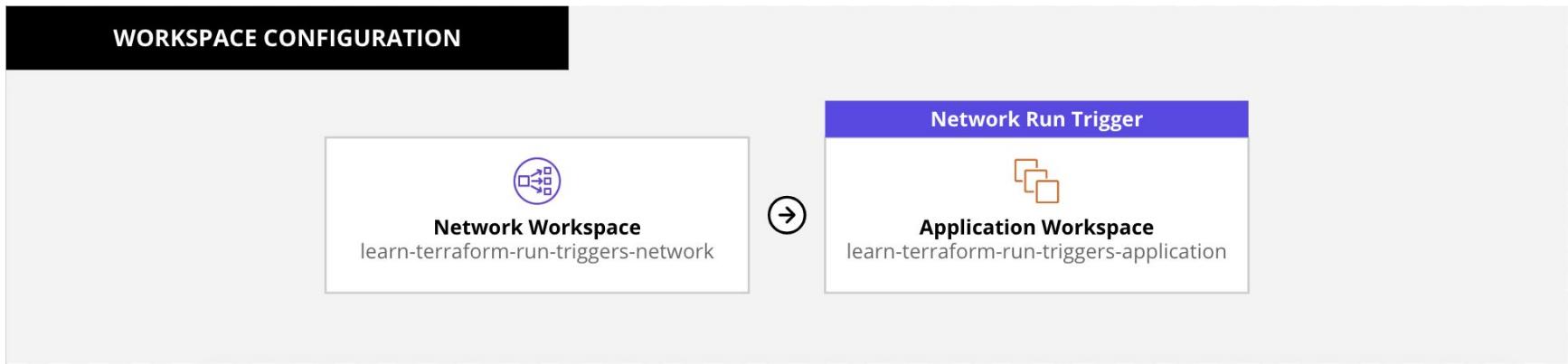
Run Triggers

- Create infrastructure pipelines in TFE
- Allow teams to manage complex infrastructure in TFE by creating infrastructure pipelines between multiple workspaces
- When a source workspace is selected, multiple dependent workspaces can be linked
- When a successful apply is executed in the source workspace, the dependent workspaces have runs triggered and can be configured to auto-apply their configurations



Use Case: Application Configuration Management

Run triggers automatically trigger updates to application configuration to rebalance servers across new subnets once they are successfully provisioned in the network workspace



Create Run Triggers

Workspace Settings → Run Triggers → Select Source Workspace

The screenshot shows a user interface for creating run triggers. At the top, there is a dark header bar with three white dots on the left. Below it, the main title is 'Run Triggers'. A descriptive text follows: 'Run triggers allow you to connect this workspace to one or more source workspaces. These connections allow runs to queue automatically in this workspace on successful apply of runs in any of the source workspaces.' Underneath, there is a warning section titled 'Auto-apply warning' with the text: 'Runs initiated as the result of a run trigger connection will not auto-apply, regardless of your auto-apply setting selection. You will need to manually apply these runs.' At the bottom, there is a section titled 'Source Workspaces' with a note: 'Select a source workspace to create a run trigger.' Below this, a message says 'No source workspaces have been selected'. There is a dropdown menu labeled '--Select item--' and a blue button labeled 'Add workspace'.

Run Triggers

Run triggers allow you to connect this workspace to one or more source workspaces. These connections allow runs to queue automatically in this workspace on successful apply of runs in any of the source workspaces.

Auto-apply warning

Runs initiated as the result of a run trigger connection will not auto-apply, regardless of your auto-apply setting selection. You will need to manually apply these runs.

Source Workspaces

Select a source workspace to create a run trigger.

No source workspaces have been selected

--Select item--

Add workspace

Run Notifications

- Run Notifications send updates/notifications to external services with details on run progress
- Notifications can be sent to up to 20 destinations
- Each workspace can be configured with it's own notification settings
- Can send either POST message to any URL via webhook, email message, or sent to Slack & post updates in channels



Notification Triggers

	Trigger	Description
Created	“run:created”	When a run is created and enters the “Pending” state.
Planning	“run:planning”	When a run acquires the lock and starts to execute.
Needs Attention	“run:needs_attention”	Human decision required. When a plan has changes and is not auto-applied, or requires a policy override.
Applying	“run:applying”	When a run begins the apply stage, after a plan is confirmed or auto-applied.
Completed	“run:completed”	When the run has completed on a happy path and can't go any further.
Errored	“run:errored”	When the run has terminated early due to error or cancellation.

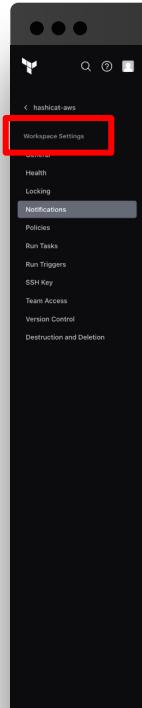


Sample Notification Payload

```
...
{
  "payload_version": 1,
  "notification_configuration_id": "nc-AeUQ2zfKZzW9TiGZ",
  "run_url": "https://app.terraform.io/app/acme-org/my-workspace/runs/run-FwnENkvDnrpyFC7M",
  "run_id": "run-FwnENkvDnrpyFC7M",
  "run_message": "Add five new queue workers",
  "run_created_at": "2019-01-25T18:34:00.000Z",
  "run_created_by": "sample-user",
  "workspace_id": "ws-XdeUVMWShTesDMME",
  "workspace_name": "my-workspace",
  "organization_name": "acme-org",
  "notifications": [
    {
      "message": "Run Canceled",
      "trigger": "run:errored",
      "run_status": "canceled",
      "run_updated_at": "2019-01-25T18:37:04.000Z",
      "run_updated_by": "sample-user"
    }
  ]
}
```

Create Notification Triggers

Workspace → Settings
→ Notifications



The screenshot shows the HashiCorp Jenkins interface. On the left, there's a sidebar with options like Workspace Settings, Notifications (which is highlighted with a red box), Policies, Run Tasks, Run Triggers, SSH Key, Team Access, Version Control, and Destruction and Deletion. The main area shows a workspace named "hashicat-aws". A red box highlights the "Notifications" link in the breadcrumb navigation at the top. Below the workspace name, it says "No workspace description available. Add workspace description." and "Unlocked". To the right, there's a "Create a Notification" form. It has sections for "Destination" (Webhook, Email, Slack, Microsoft Teams), "Name" (e.g. My Notification), "Webhook URL" (https://example.com/...), "Token" (disabled), "Health Events" (All events selected), and "Run Events" (All events selected). At the bottom are "Create a notification" and "Cancel" buttons.



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: Overview, Get Started (AWS, Azure, Docker, GCP, OCI), Fundamentals (CLI, Configuration Language, Modules, Provision, State), Use Cases (Terraform Cloud, Applications, AWS Services), and Terraform Cloud. The 'Terraform Cloud' category is currently selected. The main content area displays a breadcrumb path: Developer / Terraform / Tutorials / Terraform Cloud. A search bar is at the top right. Three tutorial cards 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

- [TFE Permissions](#)
- [API Tokens](#)
- [Organizing Workspaces with Projects](#)
- [Terraform Cloud Agents on TFE](#)
- [Manage Private Environments with Agents](#)
- [Cloud Agent Releases & Kubernetes Module](#)
- [Sentinel Language Reference](#)
- [Sentinel Foundational Policies Library](#)
- [Run Triggers & Registry: `tfe run trigger`](#)
- [Notifications API & Registry: `tfe notifications`](#)

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 Enterprise Operations

We take a deep dive into best practices for operating Terraform Enterprise instances including backup & restore operations, upgrade process, and monitoring



Program Closing

We conclude the webinar series with a short recorded session

The session and accompanying materials include an Operational Readiness Checklist for Terraform Enterprise and links to all of the program materials and recordings

Action Items

- If you haven't, share to customer.success@hashicorp.com
 - Authorized technical contacts for support
 - Stakeholders contact information (name and email addresses)
- Validate TFE architecture and design with appropriate stakeholders for approval (Security Team, Network Team, Developer Leads, etc)
- Start planning RBAC structure for the deployment (Orgs, Teams, Projects, Workspaces, Roles)
- Review Sentinel resources and determine which policy sets will be utilized in your environment(s)



Q&A





Thank you

customer.success@hashicorp.com

www.hashicorp.com/customer-success