

VODACOM

Sentinel – Policy as a code



**VERSION: 1.0**

# Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Reviewed By** | **Approved By** |
| 01/04/2021 | 1.0 | Jeyashri K |  |  |



Table of Contents

[Document History 2](#_Toc60641267)

[1. Introduction 4](#_Toc60641268)

[2. Sentinel in Terraform Cloud / Terraform Enterprise 4](#_Toc60641269)

[3. Policy Sets 5](#_Toc60641270)

[4. Enforcement Levels 5](#_Toc60641271)

[5. Defining Policies 6](#_Toc60641272)

[6. Constructing a policy set 7](#_Toc60641273)

[7. Sentinel policy code files 8](#_Toc60641274)

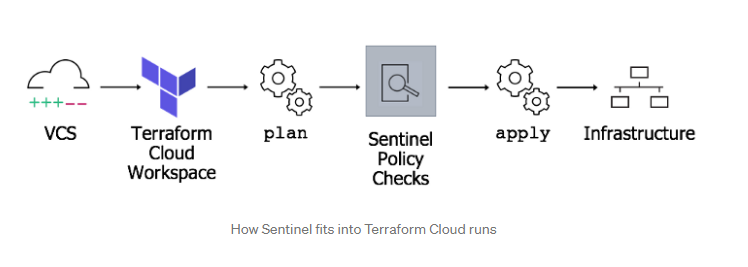
[8. Managing Policy Sets 8](#_Toc60641275)

## Introduction

Sentinel is an embedded policy-as-code framework integrated with the HashiCorp Enterprise products. It enables fine-grained, logic-based policy decisions, and can be extended to use information from external sources.

## Sentinel in Terraform Cloud / Terraform Enterprise

One of the most important features of Terraform Cloud (TFC) and Terraform Enterprise (TFE) (the self-hosted implementation of Terraform Cloud) is Sentinel, which lets you implement governance policies as code. Sentinel policies are checked between the plan and apply stages of runs in TFC and TFE.



Using Sentinel with Terraform Cloud involves:

* Defining the policies - Policies are defined using the policy language with imports for parsing the Terraform plan, state and configuration.
* [Managing policies for organizations](https://www.terraform.io/docs/cloud/sentinel/manage-policies.html) - Users with permission to manage policies can add policies to their organization by configuring VCS integration or uploading policy sets through the API. They also define which workspaces the policy sets are checked against during runs.
* [Enforcing policy checks on runs](https://www.terraform.io/docs/cloud/sentinel/enforce.html) - Policies are checked when a run is performed, after the terraform plan but before it can be confirmed or the terraform apply is executed.
* [Mocking Sentinel Terraform data](https://www.terraform.io/docs/cloud/sentinel/mock.html) - Terraform Cloud provides the ability to generate mock data for any run within a workspace. This data can be used with the Sentinel CLI to test policies before deployment.

## Policy Sets

Policy sets are groups of policies that can be enforced on [workspaces](https://www.terraform.io/docs/cloud/workspaces/index.html). A policy set can be enforced on designated workspaces, or to all workspaces in the organization.

After the plan stage of a Terraform run, Terraform Cloud checks every Sentinel policy that should be enforced on the run's workspace. This includes policies from global policy sets, and from any policy sets that are explicitly assigned to the workspace.

Policy sets are managed at an organization level; viewing and modifying them requires permission to manage policies.

## Enforcement Levels

Enforcement levels in Sentinel are used for defining behavior when policies fail to evaluate successfully. Sentinel provides three enforcement modes:

* **[hard-mandatory](https://www.terraform.io/docs/cloud/sentinel/manage-policies.html" \l "hard-mandatory)** requires that the policy passes. If a policy fails, the run is halted and may not be applied until the failure is resolved.

* **[soft-mandatory](https://www.terraform.io/docs/cloud/sentinel/manage-policies.html" \l "soft-mandatory)** is much like hard-mandatory, but allows any user with the [Manage Policies](https://www.terraform.io/docs/cloud/users-teams-organizations/permissions.html#manage-policies) permission to override policy failures on a case-by-case basis.

* **[advisory](https://www.terraform.io/docs/cloud/sentinel/manage-policies.html" \l "advisory)** will never interrupt the run, and instead will only surface policy failures as informational to the user.

## Defining Policies

Sentinel Policies for Terraform are defined using the Sentinel policy language. A policy can include imports which enable a policy to access reusable libraries, external data and functions. Terraform Cloud provides four imports to define policy rules for the plan, configuration, state, and run associated with a policy check.

* tfplan - This provides access to a Terraform plan, the file created as a result of terraform plan. The plan represents the changes that Terraform needs to make to infrastructure to reach the desired state represented by the configuration.
* [tfconfig](https://www.terraform.io/docs/cloud/sentinel/import/tfconfig.html) - This provides access to a Terraform configuration, the set of "tf" files that are used to describe the desired infrastructure state.
* [tfstate](https://www.terraform.io/docs/cloud/sentinel/import/tfstate.html) - This provides access to the Terraform state, the file used by Terraform to map real world resources to your configuration.
* [tfrun](https://www.terraform.io/docs/cloud/sentinel/import/tfrun.html) - This provides access to data associated with a run in Terraform Cloud, such as the run's workspace.

## Constructing a policy set

In order to use Sentinel in Terraform Cloud, you'll first need to create a policy set. A policy set is simply a directory structure containing a Sentinel configuration file and some policy files. These files and configurations will be treated like any other code and be checked in to a source control system. An added benefit to this is Terraform Cloud's VCS integration for Sentinel policies.

Every policy set requires a configuration file named sentinel.hcl. This configuration file defines:

* Each policy that should be checked in the set.
* The enforcement level of each policy in the set.
* Any modules which need to be made available to policies in the set.

The sentinel.hcl configuration file may contain any number of entries which look like this:

policy "terraform-maintenance-windows" {

source **=** "./terraform-maintenance-windows.sentinel

policy enforcement\_level **=** "hard-mandatory"

}

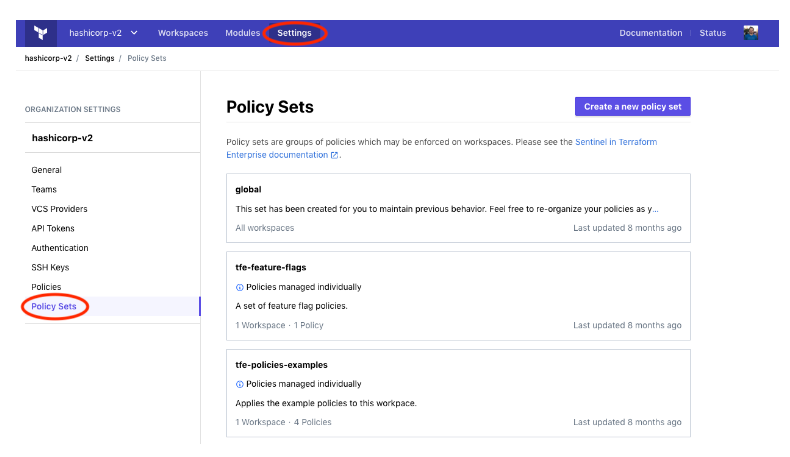
In the above, a named terraform-maintenance-windows is defined with a hard-mandatory [enforcement level](https://www.terraform.io/docs/cloud/sentinel/manage-policies.html#enforcement-levels).

## Sentinel policy code files

Sentinel policies themselves are defined in individual files (one per policy) in the same directory as the sentinel.hcl file. These files must match the name of the policy from the configuration file and carry the .sentinel suffix.

## Managing Policy Sets

To manage policy sets, go to the "Policy Sets" section on the organization settings page.



Policy sets enforced on all workspaces are marked "All workspaces" in this list; other policy sets show how many workspaces they are enforced on.

To create a new policy set, click the "Create a new policy set" button; to edit an existing set, click its entry in the list. Click the "Create policy set" or "Update policy set" button when finished.

When creating or editing a policy set, the following fields are available:

* **Name**: The name of the policy set, which is used in the UI. Must be unique to your organization. Accepts letters, numbers, -, and \_.
* **Description**: A description of the policy set's purpose. The description can be any length and supports Markdown rendering.
* **Scope of policies:** Whether the set should be enforced on all workspaces, or only on a chosen list of workspaces.
* **VCS repo or "Upload via API"**: This area allows selecting a VCS repository from an existing OAuth client connection. Choosing "Upload via API" will not configure VCS integration, and instead tarballs of policy sets may be uploaded via the API. See the policy set versions for more information on uploading policy sets using the API.
* **VCS Branch**: This field allows specifying the branch within a VCS repository from which to import new versions of policies. If left blank, the value your version control provides as the default branch of the VCS repository is used.
* **Policies Path**: This field allows specifying a sub-directory within a VCS repository for the policy set files. This allows maintaining multiple policy sets within a single repository. The value of this field should be the path to the directory containing the sentinel.hcl configuration file of the policy set you wish to configure. If left blank, the root of the repository is used. A leading / may be used, but is optional (relative paths are assumed to originate from the root of the repository).
* **Workspaces:** Which workspaces the policy set should be enforced on. This is only shown when the scope of policies is set to "Policies enforced on selected workspaces." Use the drop-down menu and "Add workspace" button to add workspaces, and the trash can button to remove them.
* **Parameters:** A list of key/value parameters that will be sent to the Sentinel runtime when a policy check is being performed for the policy set. If the value can be parsed as JSON, it will be sent to Sentinel as the corresponding type (string, boolean, integer, map or list). If it fails JSON validation, it will be sent as a string.