

Lesson End Project

Setting up Variables and Versions in Terraform Cloud

Project Agenda: To set up variables and versions in Terraform cloud to customize infrastructure configurations, track changes, and maintain consistent infrastructure states

Description: As a DevOps engineer at a tech company, you will enhance infrastructure management using Terraform Cloud. This project involves setting up variables to customize configurations and implementing version control to track changes and maintain consistent states. By leveraging Terraform Cloud's capabilities, you will ensure flexible, reusable deployments and efficient management, reducing manual interventions, and enhancing scalability.

Tools required: Terraform cloud

Prerequisites: none

Expected Deliverables: A fully configured set of variables in Terraform Cloud for customizable infrastructure, and a version-controlled environment ensuring consistent and trackable changes to infrastructure configurations.

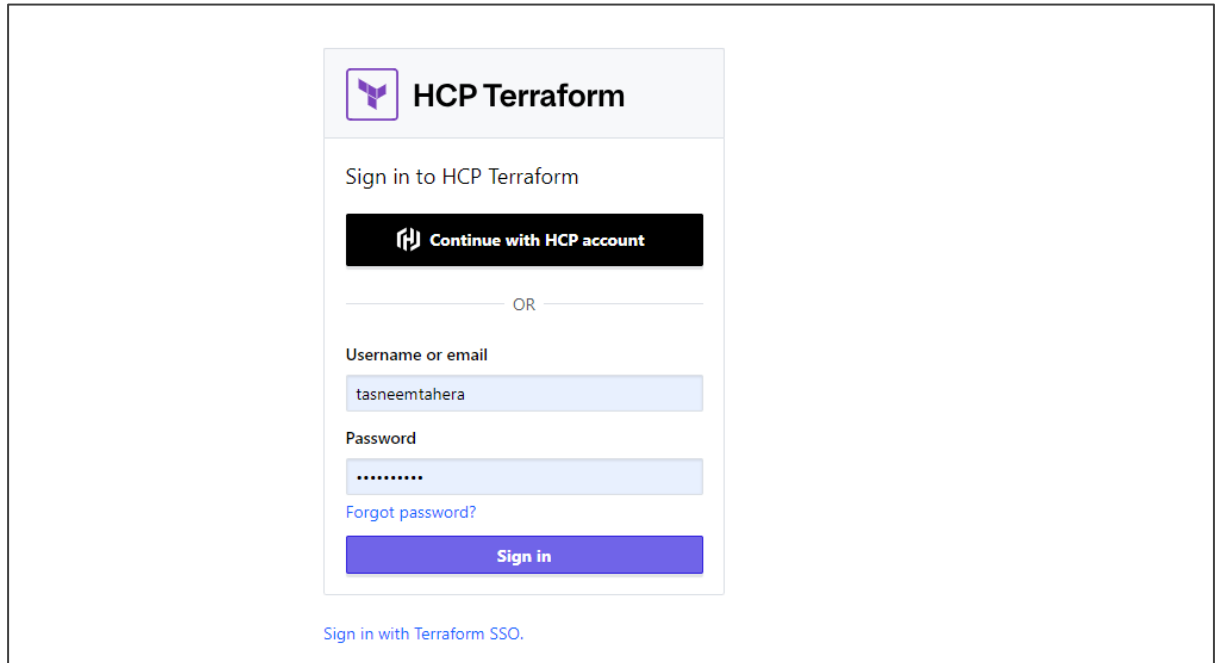
Steps to be followed:

1. Log in to Terraform cloud
2. Create an organization
3. Create a new workspace
4. Add environment variables
5. Add Terraform variables
6. Set up the version

Step 1: Log in to Terraform Cloud

1.1 Use the below link and log in to Terraform Cloud:

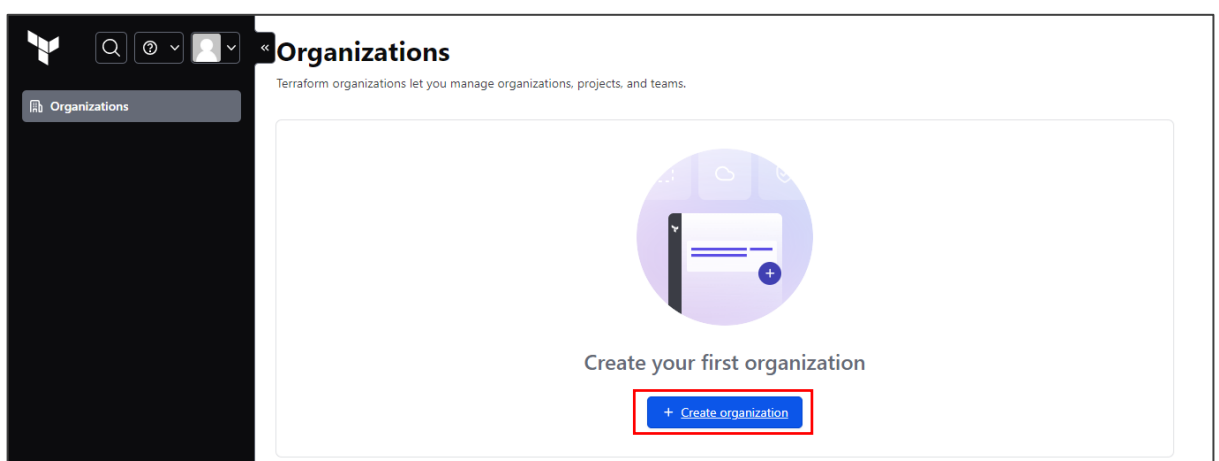
<https://app.terraform.io/app/organizations>



The screenshot shows the login interface for HCP Terraform. At the top, there is a header with the Terraform logo and the text "HCP Terraform". Below this, the text "Sign in to HCP Terraform" is displayed. A prominent black button with a white icon and the text "Continue with HCP account" is shown. Below this button, the word "OR" is centered. Underneath, there are two input fields: "Username or email" with the text "tasneemtahera" and "Password" with masked characters. A link "Forgot password?" is located below the password field. A blue "Sign in" button is at the bottom of the form. At the very bottom of the page, there is a link "Sign in with Terraform SSO."

Step 2: Create an Organization

2.1 Once you sign in, click on **+ Create organization**



2.2 Now, add **LEP-13** as the **Organization name**, then click on **Create organization**

The screenshot shows the 'Create a new organization' page in the HashiCorp Cloud Platform. The left sidebar contains navigation links: Manage, Projects, Workspaces, Registry, Usage, Settings, Visibility, Explorer, and Cloud Platform. The main content area has a header 'Organizations / New' and a title 'Create a new organization'. Below the title is a description: 'Organizations are privately shared spaces for teams to collaborate on infrastructure. [Learn more](#) about organizations in HCP Terraform.' The form has two input fields: 'Organization name' with the value 'LEP-13' and 'Email address' with the value 'tasneem.tahera@simplilearn.net'. Below the email field is a note: 'The organization email is used for any future notifications, such as billing alerts, and the organization avatar, via [gravatar.com](#).' At the bottom of the form is a blue button labeled 'Create organization'.

Organizations are privately shared spaces for teams to collaborate on infrastructure. [Learn more](#) about organizations in HCP Terraform.

Organization name

LEP-13

Organization names must be unique and can only include numbers, letters, underscores (`_`), and hyphens (`-`).

Email address

tasneem.tahera@simplilearn.net

The organization email is used for any future notifications, such as billing alerts, and the organization avatar, via [gravatar.com](#).

Create organization

2.3 Select **Version Control Workflow**, then select **GitHub.com** as your provider

The screenshot shows the 'Create a new Workspace' page in the HashiCorp Cloud Platform. The left sidebar contains navigation links: Manage, Projects, Workspaces, Registry, Usage, Settings, Visibility, Explorer, and Cloud Platform. The main content area has a header 'LEP-13 / Workspaces / New Workspace' and a title 'Create a new Workspace'. Below the title is a description: 'HCP Terraform organizes your infrastructure resources by workspaces. A workspace contains infrastructure resources, variables, state data, and run history. [Learn more](#) about workspaces in HCP Terraform.' The form has a section 'Choose your workflow' with three options: 'Version Control Workflow', 'CLI-Driven Workflow', and 'API-Driven Workflow'. The 'Version Control Workflow' option is highlighted with a red box. Below the 'Version Control Workflow' option is a note: 'Trigger runs based on changes to configuration in repositories. Best for those who need traceability and transparency'. Below the 'CLI-Driven Workflow' option is a note: 'Trigger runs in a workspace using the Terraform CLI. Best for those comfortable with Terraform CLI'. Below the 'API-Driven Workflow' option is a note: 'Trigger runs using the HCP Terraform API. Best for those with custom integrations and pipelines'. At the bottom right of the form is a 'Cancel' button.

HCP Terraform organizes your infrastructure resources by workspaces. A workspace contains infrastructure resources, variables, state data, and run history. [Learn more](#) about workspaces in HCP Terraform.

Choose your workflow

Version Control Workflow

Trigger runs based on changes to configuration in repositories.

Best for those who need traceability and transparency

CLI-Driven Workflow

Trigger runs in a workspace using the Terraform CLI.

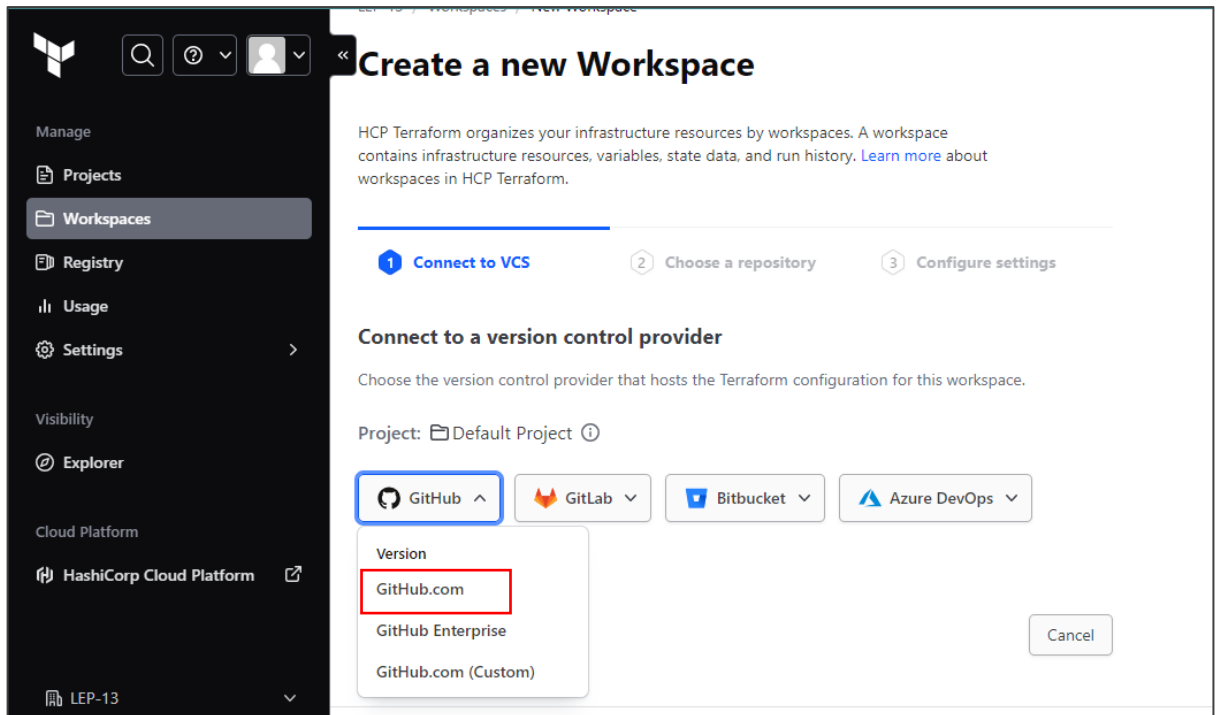
Best for those comfortable with Terraform CLI

API-Driven Workflow

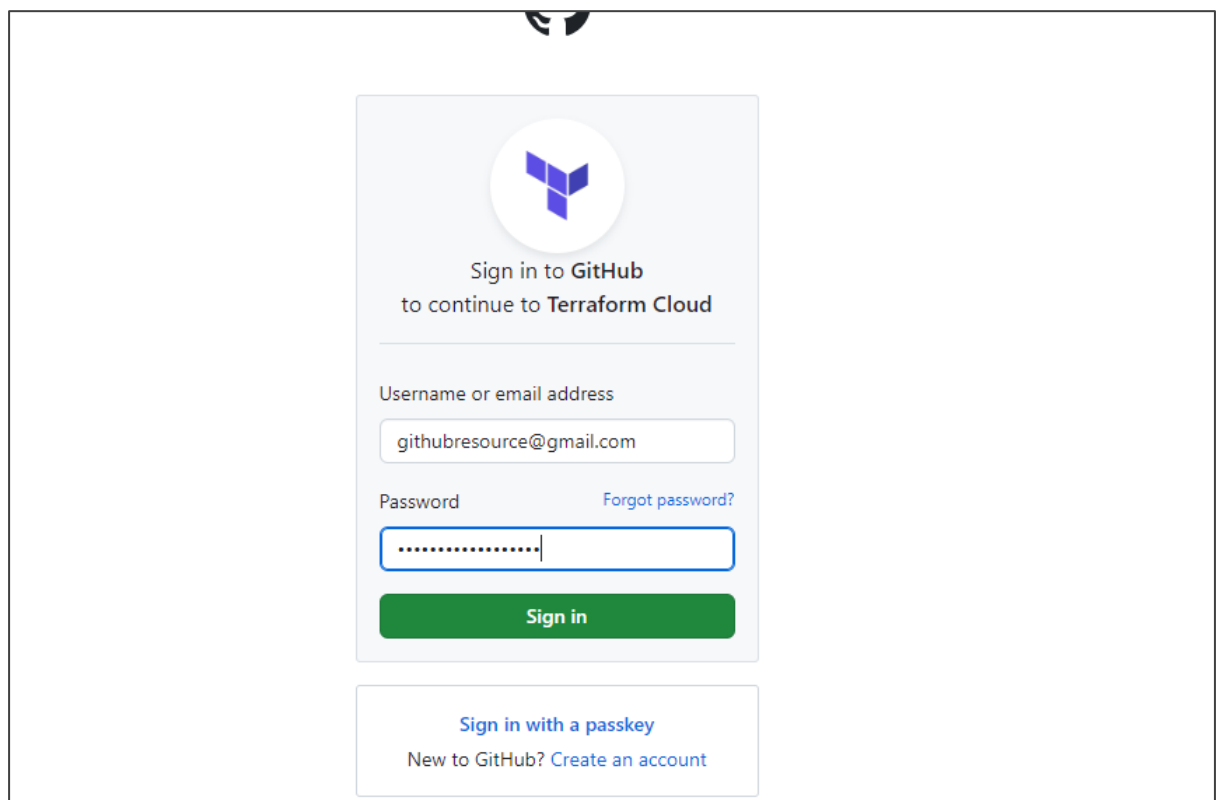
Trigger runs using the HCP Terraform API.

Best for those with custom integrations and pipelines

Cancel

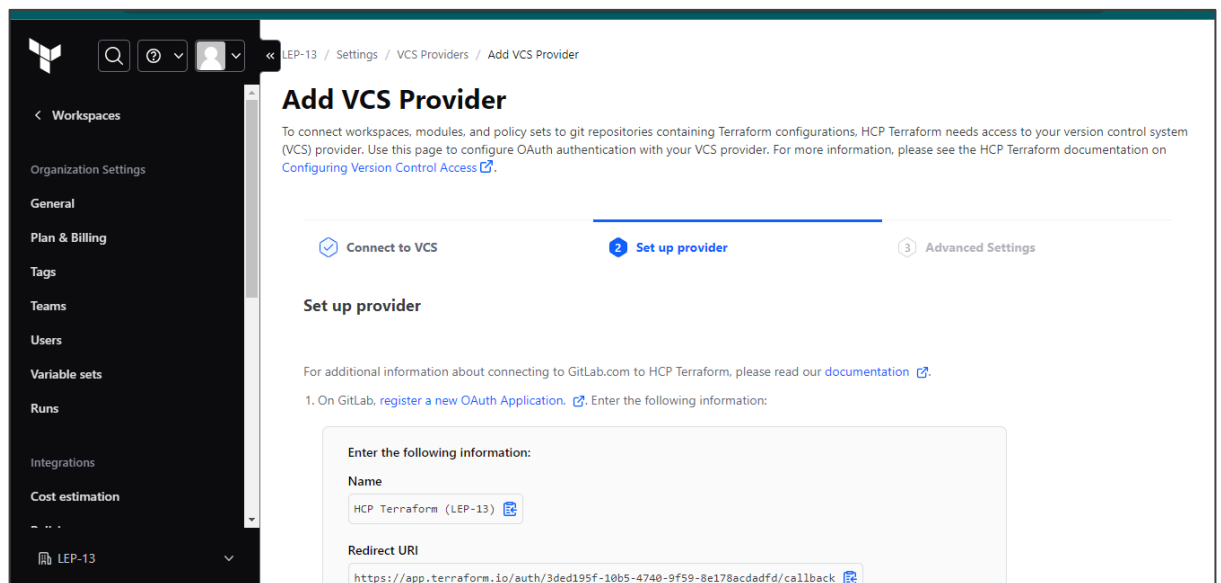
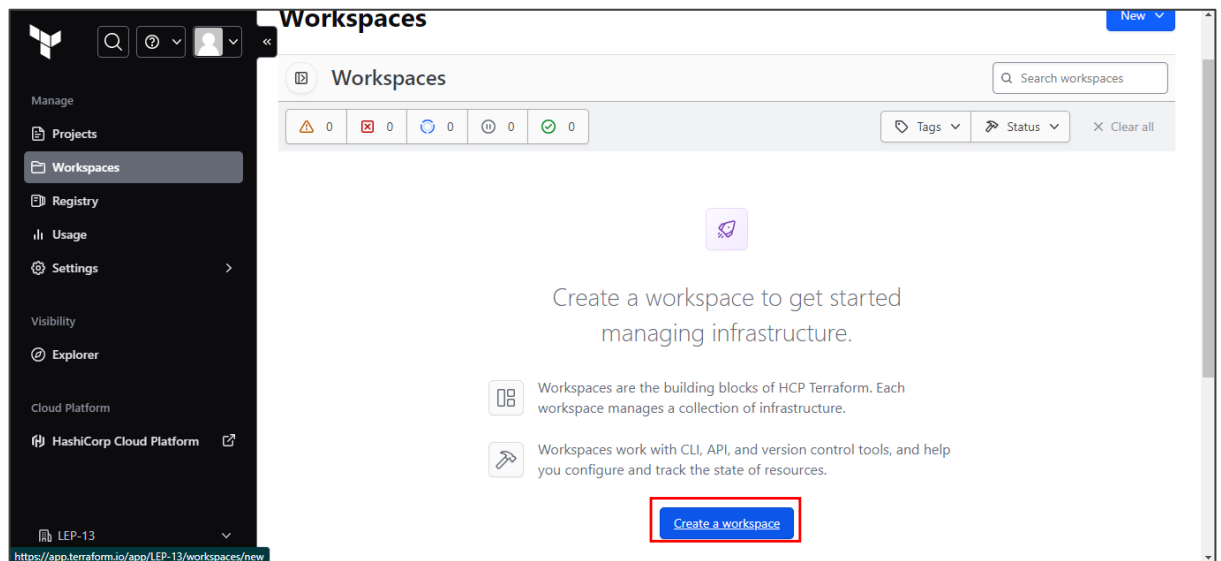


2.4 Sign in to your GitHub account



Step 3: Create a new workspace

3.1 Now click on **Create a workspace**



It will redirect you to the **Set up provider** section.

3.2 Add the **Client ID** and **Client Secret**, then click on **Connect and continue**

Authorization callback URL
<https://app.terraform.io/auth/c0c0d580-ebbf-414e-9c01-e2693aa97679/callback>

2. After clicking the "Register application" button, you'll be taken to the new application's page. Enter the Client ID below:

Name
An optional display name for your VCS Provider. This is helpful if you will be configuring multiple instances of the same provider.

Client ID

3. Next, generate a new client secret and enter the value below:

Client Secret

[< Back](#) [Cancel](#) [Connect and continue](#)

Note: Copy the Client ID and Client Secret key from your GitHub account.

3.3 Now, add **LEP-13** as **Workspace Name**, then click on **Create**

LEP-13 / Workspaces / New Workspace

Create a new Workspace

HCP Terraform organizes your infrastructure resources by workspaces. A workspace contains infrastructure resources, variables, state data, and run history. [Learn more](#) about workspaces in HCP Terraform.

Configure Settings

Workspace Name

The name of your workspace is unique and used in tools, routing, and UI. Dashes, underscores, and alphanumeric characters are permitted. [Learn more about naming workspaces](#)

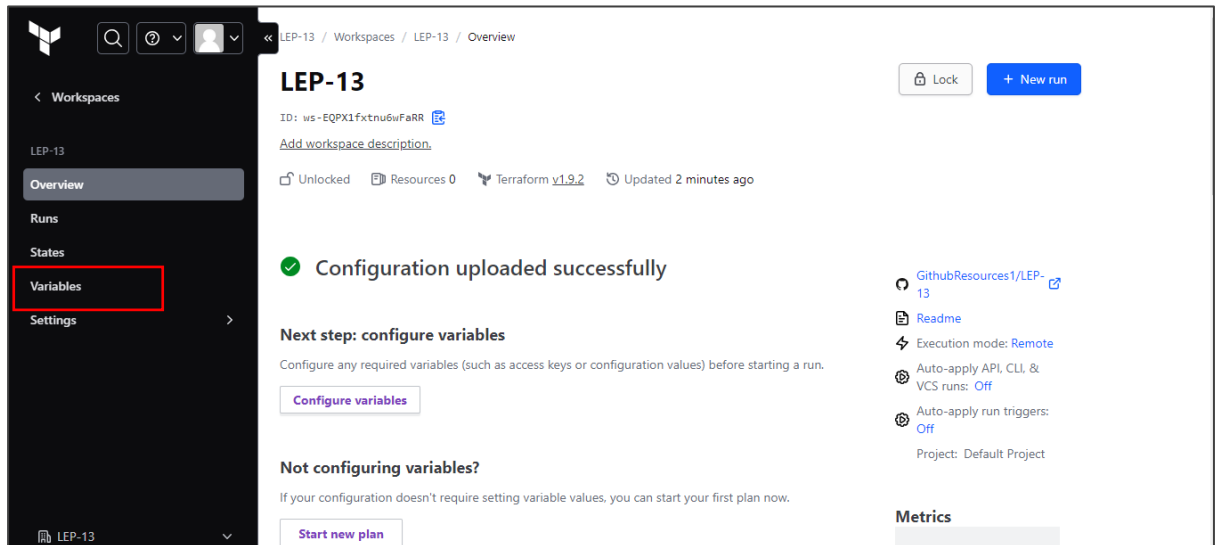
Description (Optional)

[Advanced options](#)

[< Previous](#) [Cancel](#) [Create](#)

Step 4: Add environment variables

4.1 Once the workspace is created, click on the **Variables** tab



Workspace: LEP-13 / Workspaces / LEP-13 / Overview

LEP-13

ID: ws-EQPX1fxtnu6wFaRR [Add workspace description](#)

Unlocked Resources 0 Terraform v1.9.2 Updated 2 minutes ago

Configuration uploaded successfully

Next step: configure variables

Configure any required variables (such as access keys or configuration values) before starting a run.

[Configure variables](#)

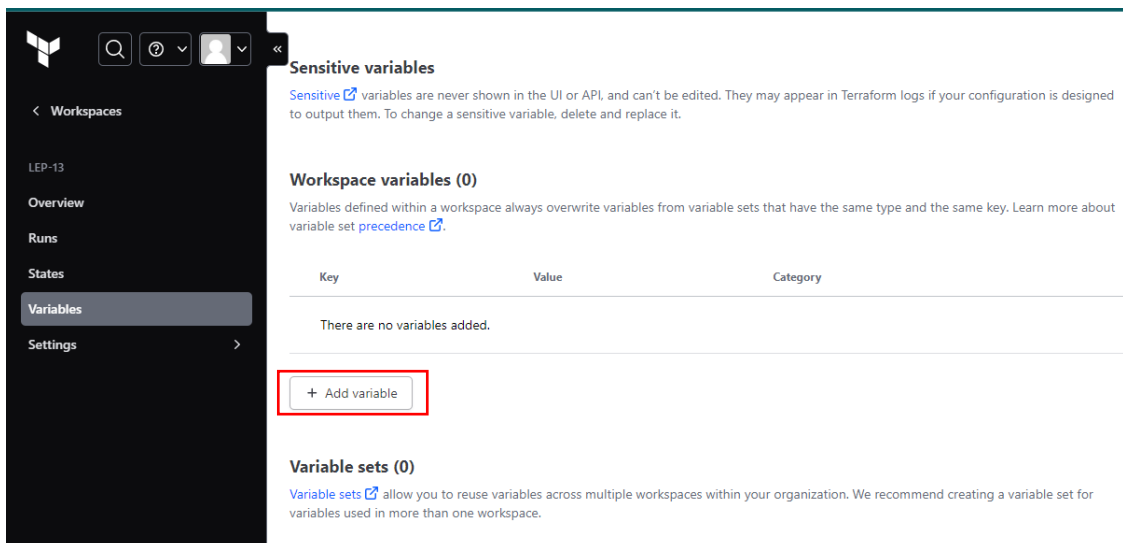
Not configuring variables?

If your configuration doesn't require setting variable values, you can start your first plan now.

[Start new plan](#)

Metrics

4.2 Click on **+Add variable**



Sensitive variables

[Sensitive](#) variables are never shown in the UI or API, and can't be edited. They may appear in Terraform logs if your configuration is designed to output them. To change a sensitive variable, delete and replace it.

Workspace variables (0)

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](#).

Key	Value	Category
There are no variables added.		

[+ Add variable](#)

Variable sets (0)

[Variable sets](#) allow you to reuse variables across multiple workspaces within your organization. We recommend creating a variable set for variables used in more than one workspace.

4.3 Click on an **Environment variable**, add **AWS_ACCESS_KEY_ID** under **Key**, and enter your AWS access key in the **Value** field. Then click on **Add variable**.

workspace variables (0)

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](#).

Key	Value	Category
There are no variables added.		

Select variable category

☐ Terraform variable
These variables should match the declarations in your configuration. Click the HCL box to use interpolation or set a non-string value.

☒ **Environment variable**
These variables are available in the Terraform runtime environment.

Key: AWS_ACCESS_KEY_ID

Value: AKIARN32ZJJKVIX5XP7 ☒ Sensitive ⓘ

Description (Optional): description (optional)

Add variable Cancel

4.4 Now, click on **+ Add variable** to add the AWS Secret key

Sensitive variables

[Sensitive](#) variables are never shown in the UI or API, and can't be edited. They may appear in Terraform logs if your configuration is designed to output them. To change a sensitive variable, delete and replace it.

Workspace variables (1)

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](#).

Key	Value	Category
AWS_ACCESS_KEY_ID SENSITIVE	Sensitive - write only	env

+ Add variable

Variable sets (0)

[Variable sets](#) allow you to reuse variables across multiple workspaces within your organization. We recommend creating a variable set for variables used in more than one workspace.

4.5 Click on the **Environment variable**, add **AWS_SECRET_ACCESS_KEY** under **Key**, and enter your **AWS Secret key** in the **Value** field. Then click on **Add variable**.

variable set [precedence](#)

Key	Value	Category
AWS_ACCESS_KEY_ID SENSITIVE	<i>Sensitive - write only</i>	env

Select variable category

☐ Terraform variable
These variables should match the declarations in your configuration. Click the HCL box to use interpolation or set a non-string value.

☒ **Environment variable**
These variables are available in the Terraform runtime environment.

Key: AWS_SECRET_ACCESS_KEY

Value: xzsX+t4GfNZAmIXu/zQ3Un4r6kdQVIVQk5ekfu ☒ Sensitive ⓘ

Description (Optional):
description (optional)

Add variable Cancel

Step 5: Add Terraform variables

5.1 Now, click on **+ Add variable**

Sensitive variables

[Sensitive](#) variables are never shown in the UI or API, and can't be edited. They may appear in Terraform logs if your configuration is designed to output them. To change a sensitive variable, delete and replace it.

Workspace variables (1)

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](#).

Key	Value	Category
AWS_ACCESS_KEY_ID SENSITIVE	<i>Sensitive - write only</i>	env

+ Add variable

Variable sets (0)

[Variable sets](#) allow you to reuse variables across multiple workspaces within your organization. We recommend creating a variable set for variables used in more than one workspace.

5.2 Click on **Terraform variable**, add **region** under **Key**, and enter **us-west-2** as Value, then click on **Add variable**

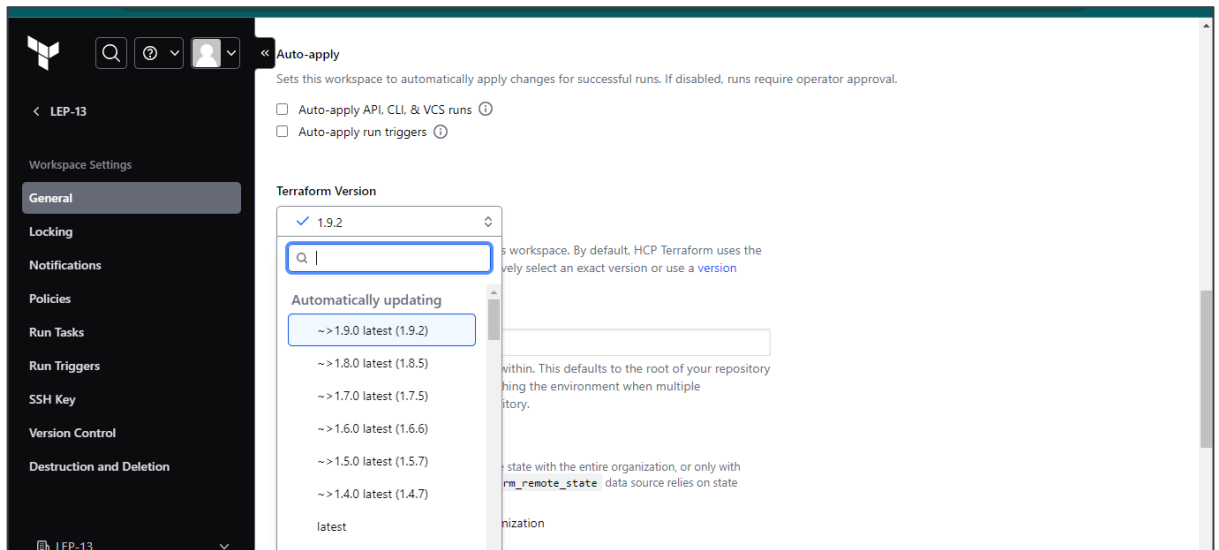
The screenshot shows the Terraform Cloud interface with a sidebar on the left containing 'Workspaces', 'LEP-13', 'Overview', 'Runs', 'States', 'Variables', and 'Settings'. The main panel displays a 'SENSITIVE' variable named 'AWS_SECRET_ACCESS_KEY' with a value of 'env'. Below this, a dialog box titled 'Select variable category' is open. It has two radio buttons: 'Terraform variable' (selected) and 'Environment variable'. The 'Terraform variable' section explains that these variables match declarations in the configuration and can use HCL interpolation. The 'Environment variable' section explains that these are available in the Terraform runtime environment. Below the radio buttons, there are input fields for 'Key' (containing 'region') and 'Value' (containing 'us-west-2'). There are also checkboxes for 'HCL' and 'Sensitive'. A 'Description (Optional)' text area is also present. At the bottom of the dialog, there are 'Add variable' and 'Cancel' buttons. The 'Add variable' button is highlighted with a red rectangle.

Step 6: Set up the version

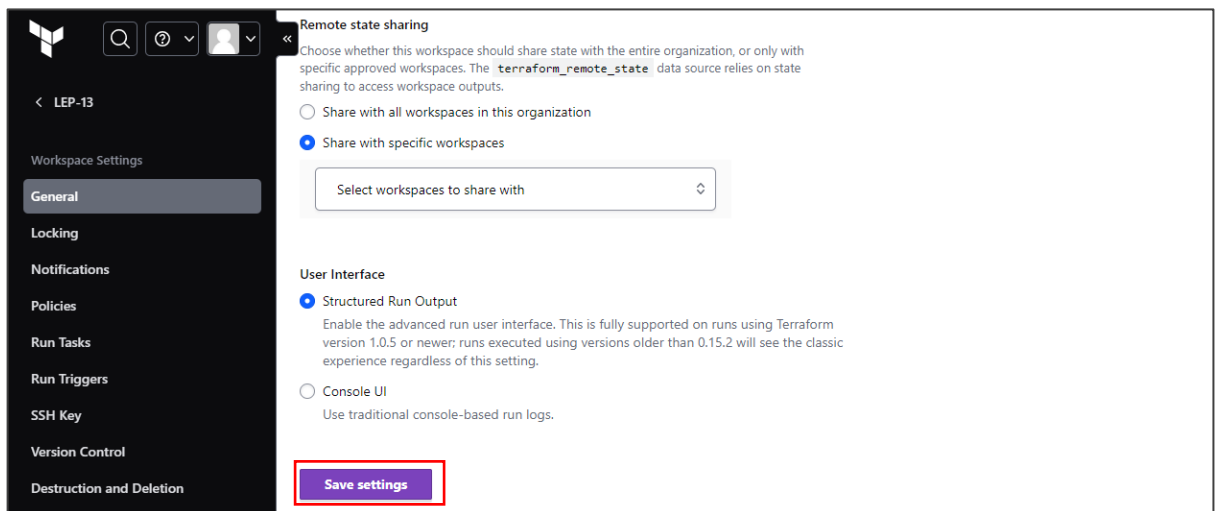
6.1 Navigate to **Settings**

The screenshot shows the Terraform Cloud interface with the sidebar on the left. The 'Settings' option is highlighted with a red rectangle. The main panel shows the 'SENSITIVE' variable 'AWS_SECRET_ACCESS_KEY' with a value of 'env'. Below this, there is a '+ Add variable' button. Further down, there is a section titled 'Variable sets (0)' with a description: 'Variable sets allow you to reuse variables across multiple workspaces within your organization. We recommend creating a variable set for variables used in more than one workspace.' At the bottom of this section, there is a button labeled 'Apply variable set'.

6.2 Scroll down to the **Terraform Version** section and select the latest version



6.3 Click on **Save settings** to set up the version



By following these steps, you have successfully set up variables and managed versions in Terraform Cloud, enabling customized infrastructure configurations, change tracking, and consistent infrastructure states.