

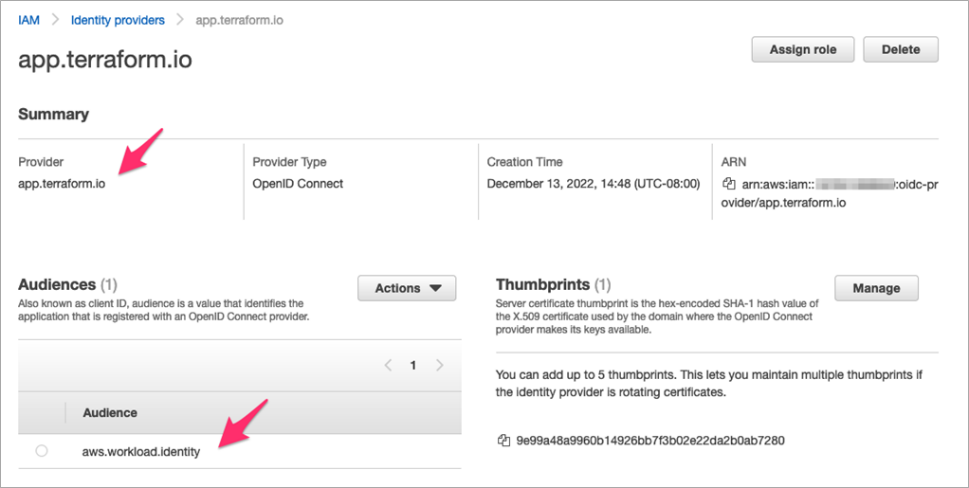
# Version History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Title | Version | Author | Summary | Date |
| Terraform Cloud: Create Workspace | 1.0 | Robert Tekoh  Cloud Engineer III,  Fredhutch | Initial Draft | March 15, 2024 |
|  |  |  |  |  |

**Terraform Cloud: Create Workspace.**

# Prerequisites for creating a workspace

1. Make sure the **app.terraform.io** identity provider is created in the account where you intend to deploy resources.



1. Make sure the terraform role with trust policy for the above IDP is present. Copy the role ARN to configure workspace variables later.

TFC\_AWS\_RUN\_ROLE\_ARN = IAM Role ARN to authenticate against.

# Create a Workspace

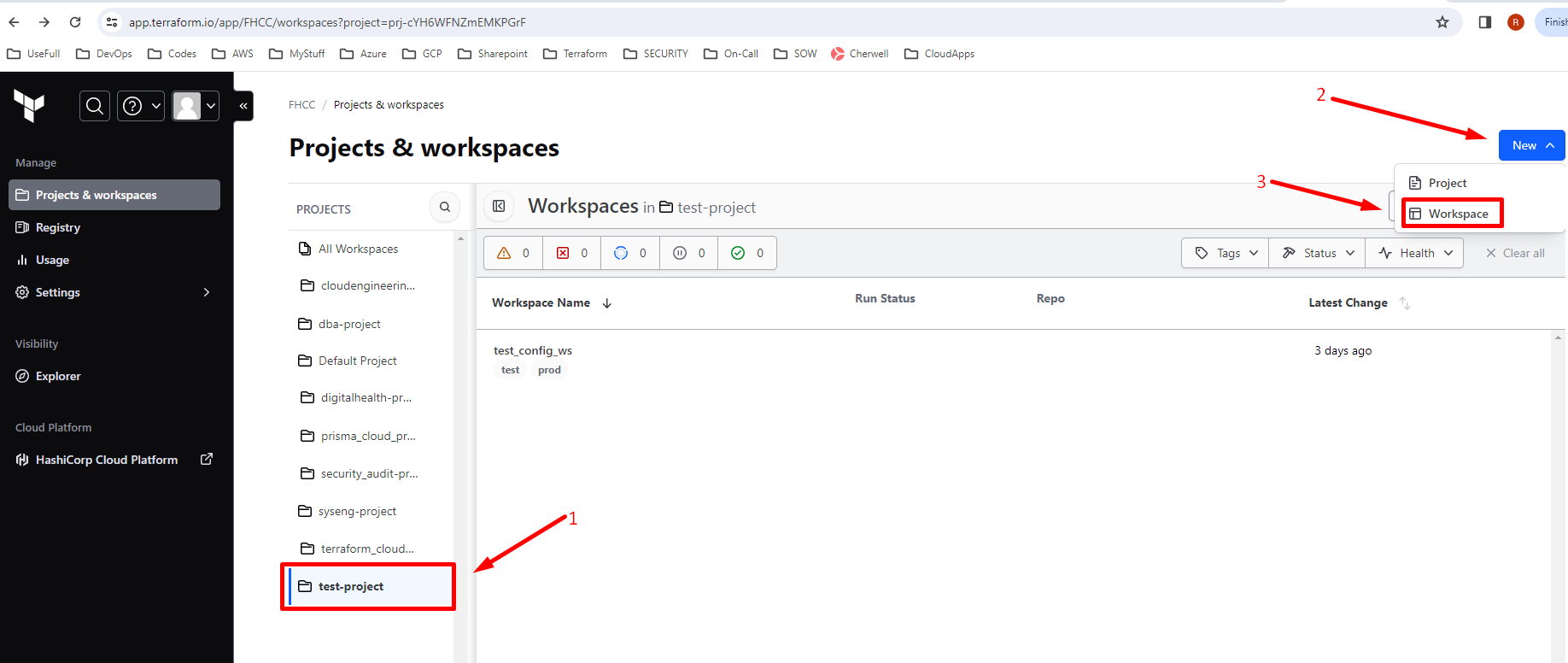
## **Steps to Create a Workspace**

## Step 1: Login to Terraform Cloud

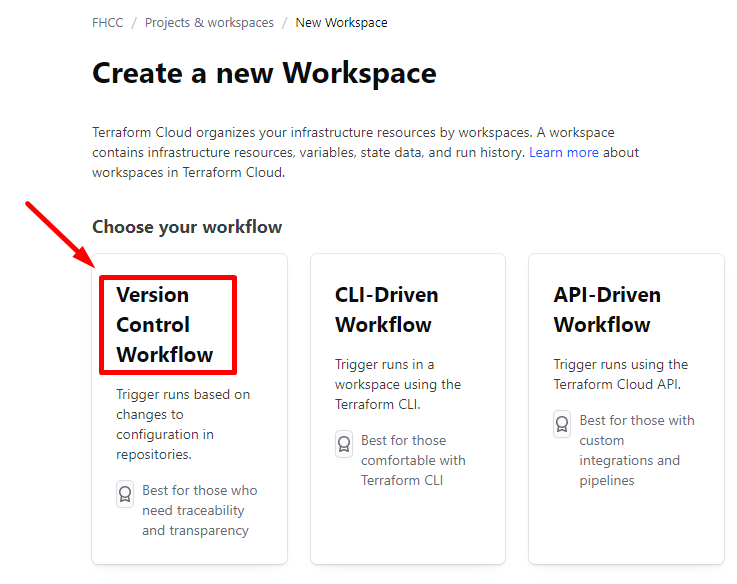
1. **Log in** to your Terraform Cloud account.

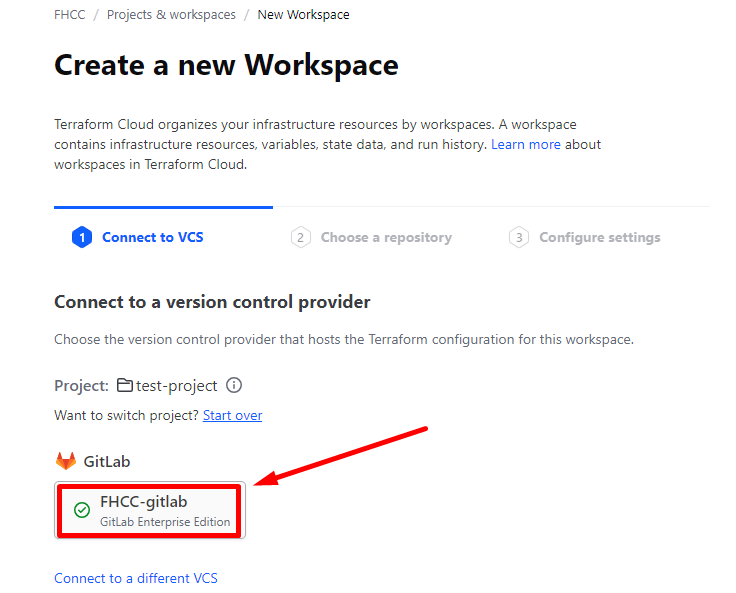
## Step 2: Create a New Workspace

1. Click on **Projects & Workspaces** to view your existing workspaces.
2. Click **New** and select **Workspace** from the dropdown menu.
3. Choose the **VCS workflow**. !Create Workspace

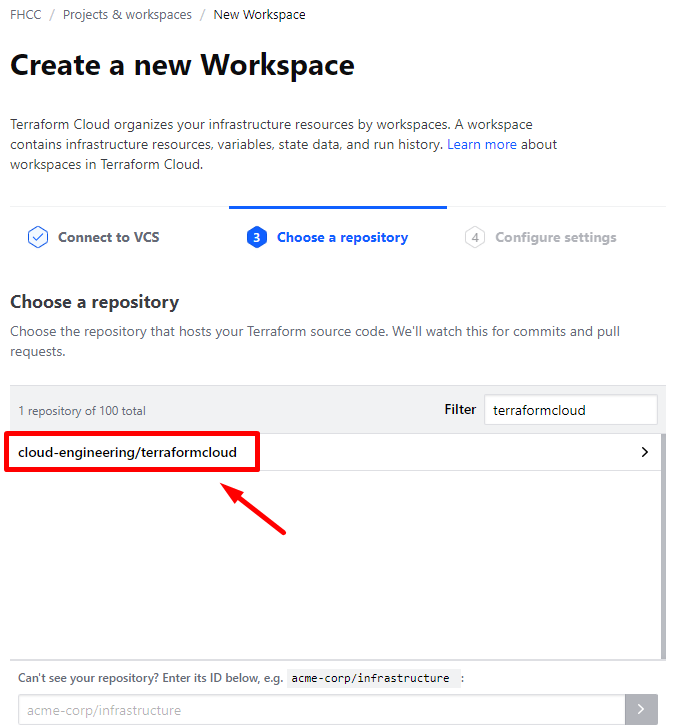


All projects will use Version control workflows.





Select your teams’ projects Repository.

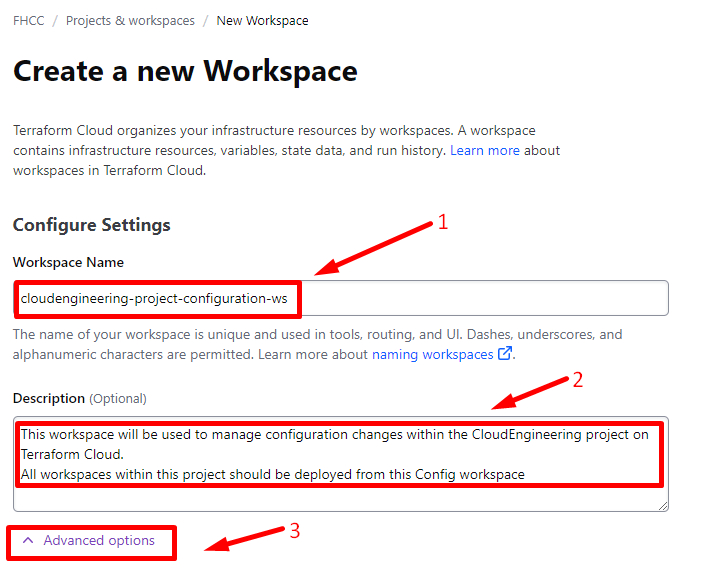


## Step 3: Configure Workspace Settings

1. **Workspace Name**: Give your workspace a descriptive name.

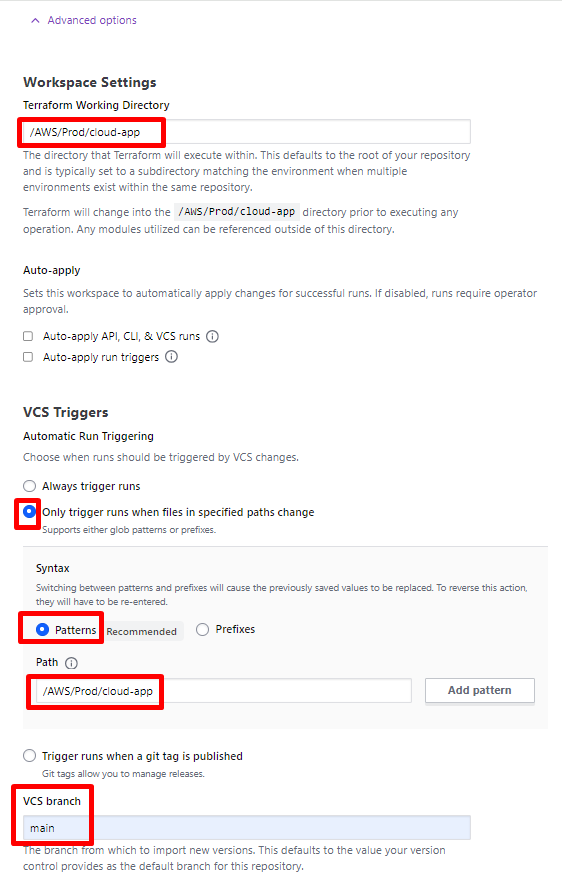
Follow the guide in terraform cloud Naming guide to name the workspace. Make sure to provide a good description of the workspace and **click** on “**Advanced Options**” to define workspace deployment configs.

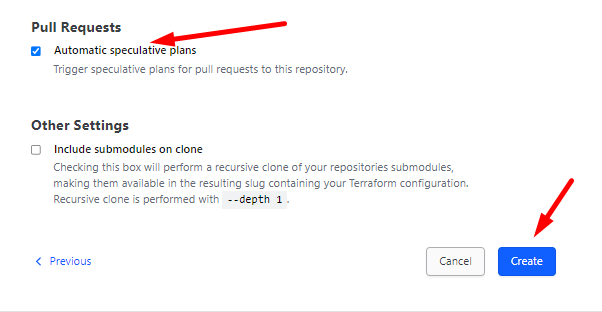
1. **Terraform Working Directory**: Specify the directory where Terraform will execute runs (usually the project directory in your repository).



1. **Apply Method**:
   1. **Auto apply**: Terraform will apply changes from successful plans without manual approval. Suitable for non-interactive environments.
   2. **Manual apply**: Terraform will prompt for approval before applying changes. !Workspace Settings

**NB: Do not use auto-apply**



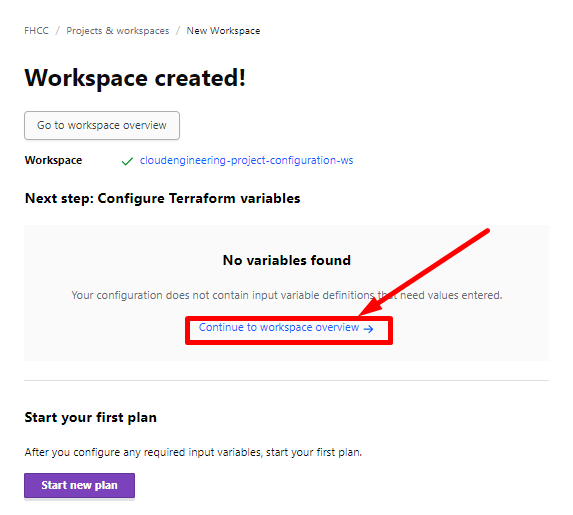


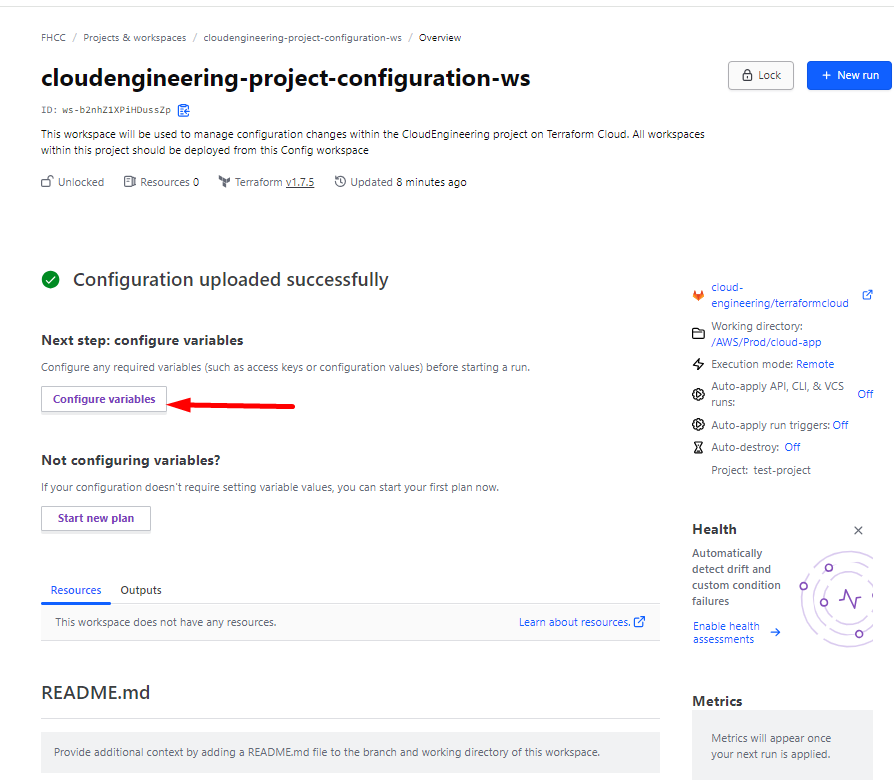
## Step 4: Set Up Variables.

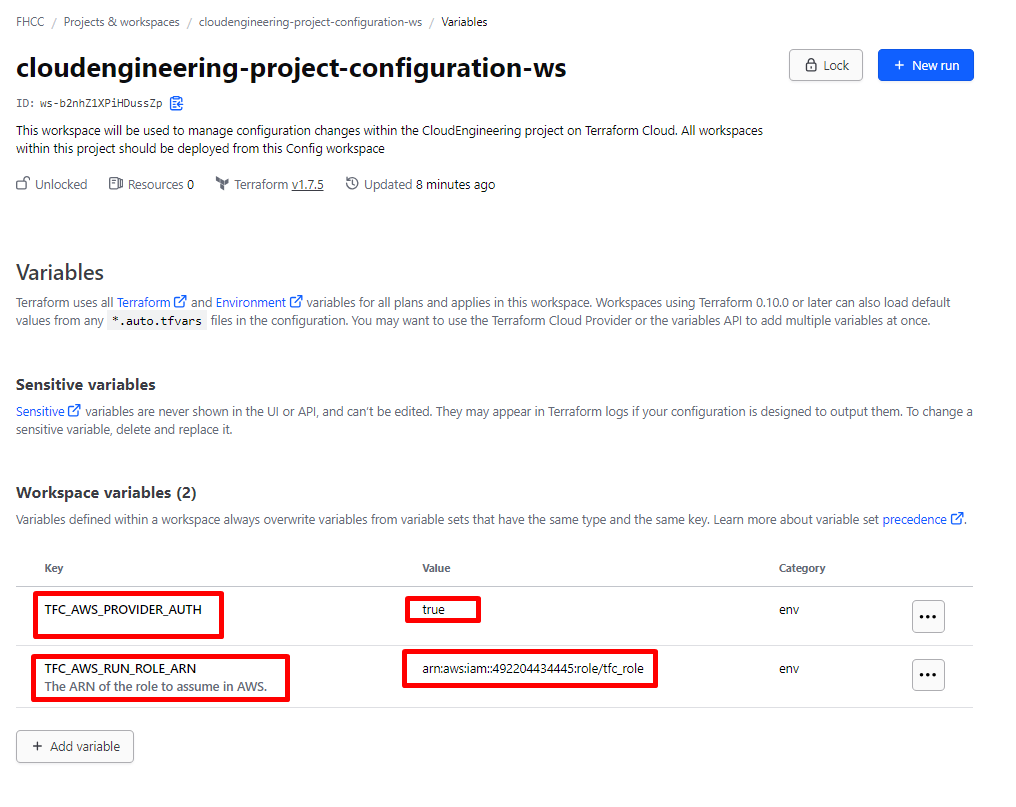
1. Define any necessary environment variables or secrets for your workspace.
2. You **MUST** set environment variables in your Terraform Cloud workspace in order to configure how Terraform Cloud will authenticate to AWS using dynamic credentials:

**NB:** No AWS\_ACCESS\_KEY\_ID and AWS\_SECRET\_ACCESS\_KEY should be used.

* **TFC\_AWS\_PROVIDER\_AUTH** = set to true to use dynamic provider credentials.
* **TFC\_AWS\_RUN\_ROLE\_ARN** = IAM Role ARN to authenticate against.







## Step 5: Triggering Runs

1. Terraform Cloud will automatically trigger runs when changes are pushed to your connected VCS repository. Follow your teams merge request process for approvals to merge to main branch.
2. You can also manually trigger runs from the Terraform Cloud UI.