Lesson 13 Demo 03

Working with Private Registry on Terraform Cloud

Objective: To demonstrate how to work with a private module registry on Terraform Cloud, enabling the use and management of modules within a private organization

Tools required: AWS Account, Terraform, VS Code

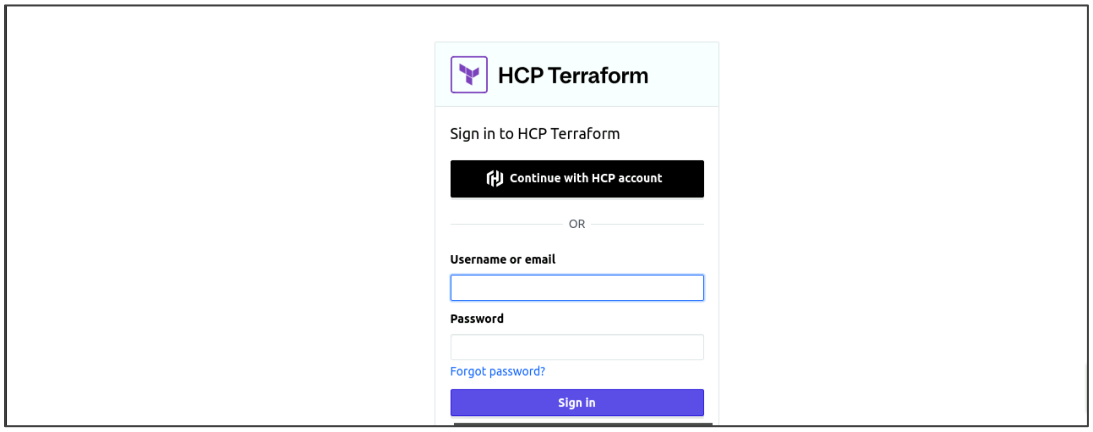
Prerequisites: None

Steps to be followed:

1. Set up a Terraform Cloud account and organization
2. Create and publish a module to the private registry
3. Initialize the Terraform configuration repository

Step 1: Set up Terraform Cloud account and organization

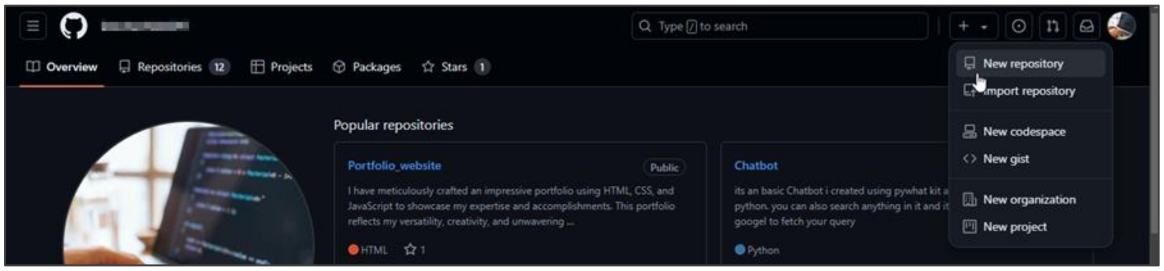
1.1 Go to Terraform Cloud to create an account and organization:



Note: Refer to demo 01 of lesson 13 for creating a Terraform Cloud account and organization

Step 2: Create and publish a module to the private registry

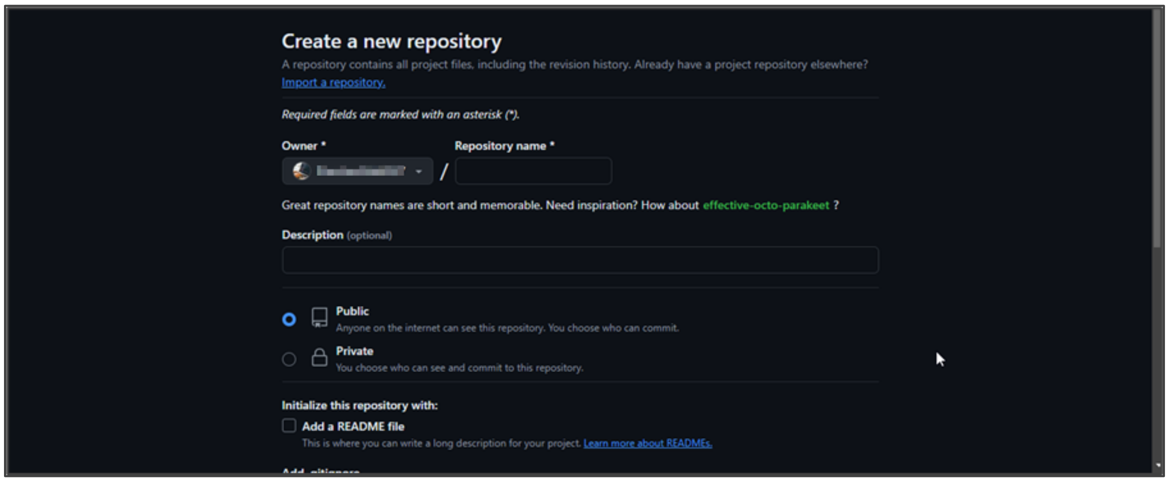
2.1 Create a new repository on GitHub



I have meticulously crafted an impressive portfolio using HTML, CSS, and JavaScript to showcase my expertise and accomplishments. This portfolio reflects my versatility, creativity, and unwavering ...

its an basic Chatbot i created using pywhat kit a python. you can also search anything in it and it googel to fetch your query

2.2 Give a name for your repository


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * [redacted] / Repository name *

Great repository names are short and memorable. Need inspiration? How about effective-octo-parakeet ?

Description (optional)

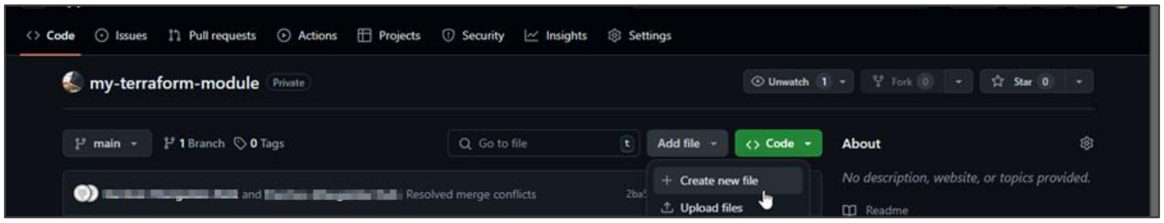
[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file This is where you can write a long description for your project. Learn more about READMEs.

2.3 Create three files **main.tf, variables.tf,** and **output.tf** in your repository

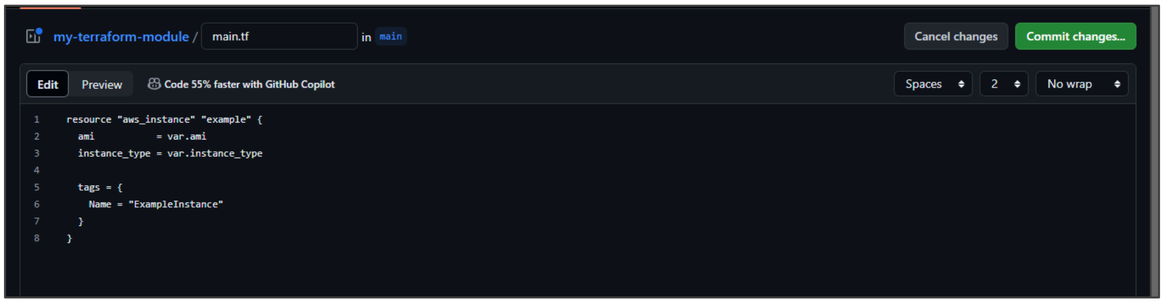


[redacted] and [redacted] Resolved merge conflicts

No description, website, or topics provided.

2.4 Add the following script in the **main.tf** file and commit the changes:

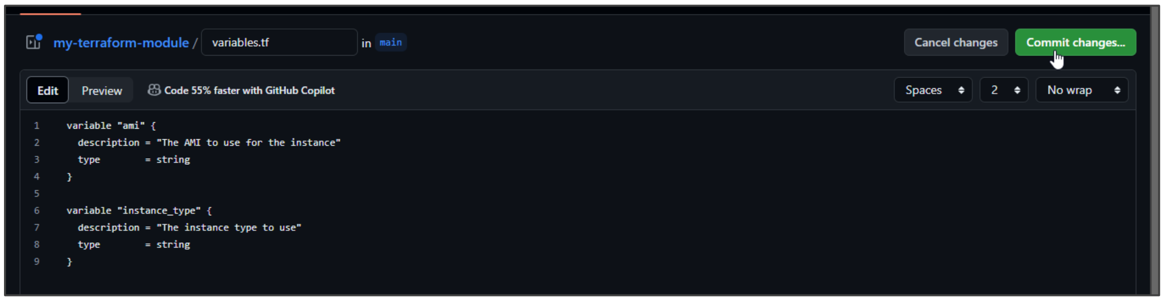
```
resource "aws_instance" "example" {
  ami           = var.ami
  instance_type = var.instance_type
  tags = {
    Name = "ExampleInstance"
  }
}
```


my-terraform-module / main.tf in main
Cancel changes
Commit changes...
Edit
Preview
Code 55% faster with GitHub Copilot
Spaces
2
No wrap

```
resource "aws_instance" "example" {
  ami           = var.ami
  instance_type = var.instance_type

  tags = {
    Name = "ExampleInstance"
  }
}
```

2.5 Add the following script to the **variables.tf** file and commit:

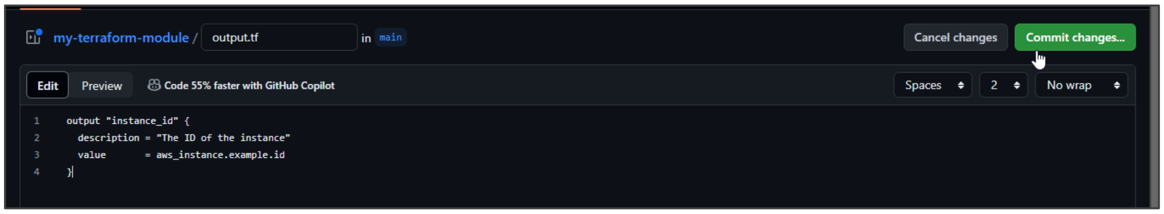
```
variable "ami" {
  description = "The AMI to use for the instance"
  type        = string
}
variable "instance_type" {
  description = "The instance type to use"
  type        = string
}
```


my-terraform-module / variables.tf in main
Cancel changes
Commit changes...
Edit
Preview
Code 55% faster with GitHub Copilot
Spaces
2
No wrap

```
variable "ami" {
  description = "The AMI to use for the instance"
  type        = string
}

variable "instance_type" {
  description = "The instance type to use"
  type        = string
}
```

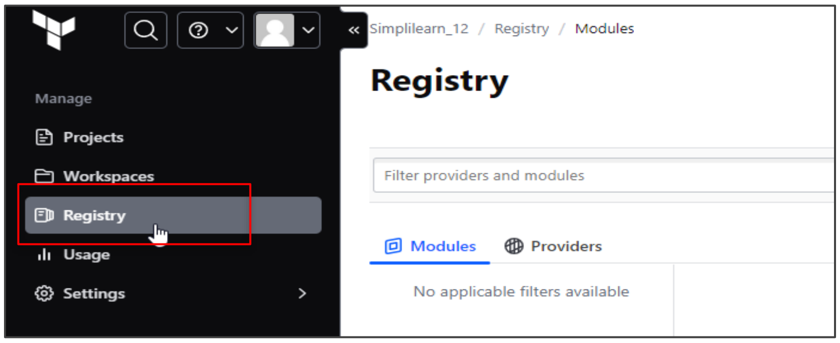
2.6 Add the following script to the **output.tf** file and commit:

```
output "instance_id" {
  description = "The ID of the instance"
  value       = aws_instance.example.id
}
```

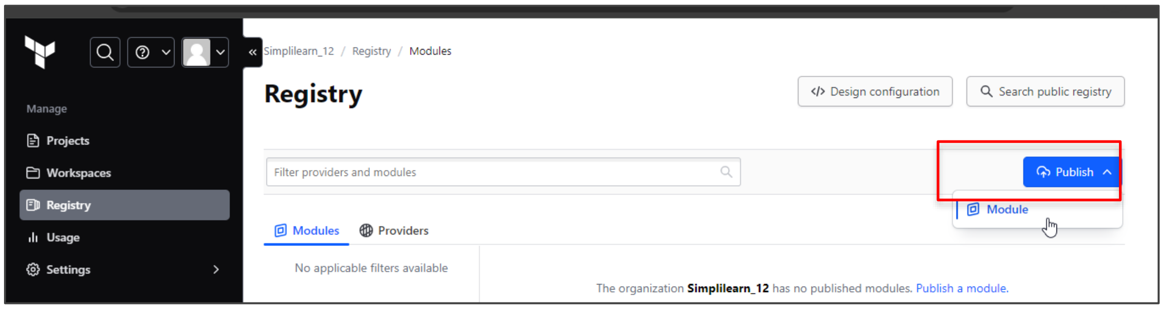

my-terraform-module / output.tf in main
Cancel changes
Commit changes...
Edit
Preview
Code 55% faster with GitHub Copilot
Spaces
2
No wrap

```
output "instance_id" {
  description = "The ID of the instance"
  value       = aws_instance.example.id
}
```

2.7 Go to Terraform Cloud, navigate to your organization, and go to the **Registry** tab


Simplilearn_12 / Registry / Modules
Registry
Manage
Projects
Workspaces
Registry
Usage
Settings
Filter providers and modules
Modules
Providers
No applicable filters available

2.8 Click on **Publish** to publish your module from your VCS.



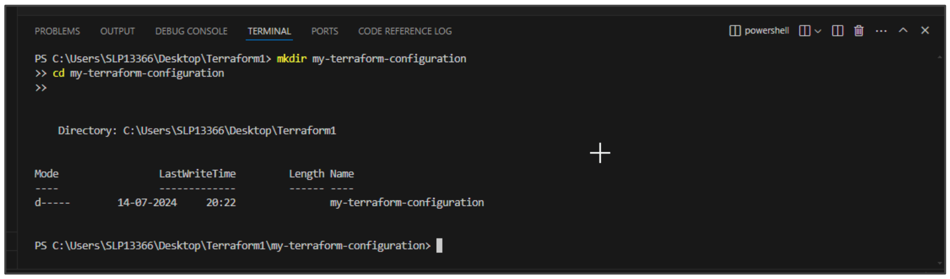
The organization **Simplilearn_12** has no published modules. Publish a module.

Step 3: Initialize the Terraform configuration repository

3.1 Run the following command to create a new directory for your Terraform configuration and navigate to that directory:

mkdir my-terraform-configuration

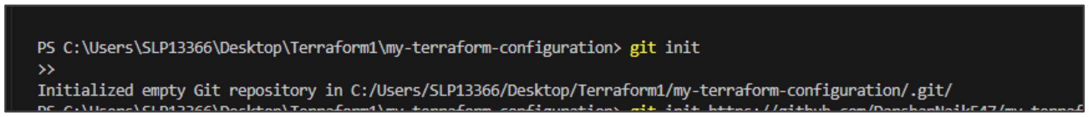
cd my-terraform-configuration



```
PS C:\Users\SLP13366\Desktop\Terraform1> mkdir my-terraform-configuration
>> cd my-terraform-configuration
>>


    Directory: C:\Users\SLP13366\Desktop\Terraform1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        14-07-2024     20:22                my-terraform-configuration

PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration>
```

3.2 Execute the following command to initialize a Git repository:

git init

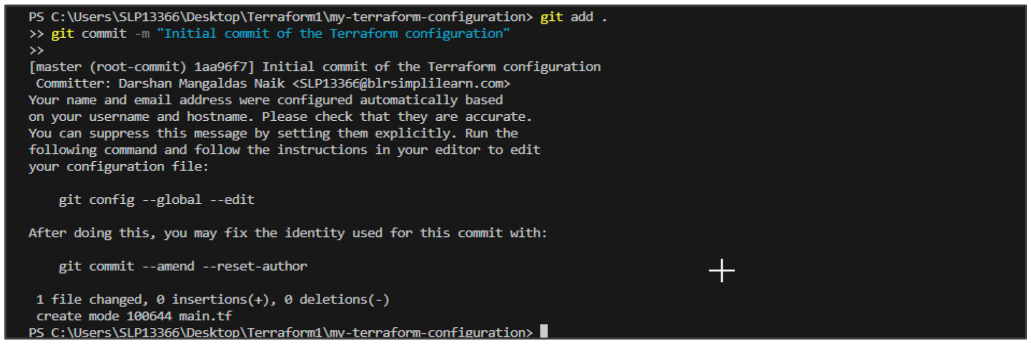


```
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration> git init
>>
Initialized empty Git repository in C:/Users/SLP13366/Desktop/Terraform1/my-terraform-configuration/.git/
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration> git init https://github.com/DarshanNaik547/my-terraf
```

3.3 Run the following command to add and commit the configuration files:

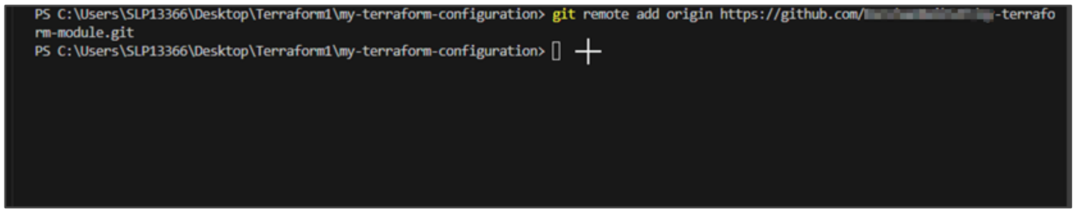
git add .

git commit -m “Initial commit of the Terraform configuration



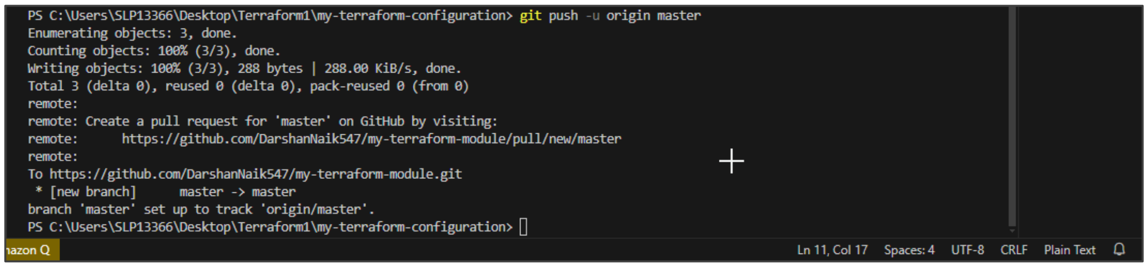
```
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration> git add .
>> git commit -m "Initial commit of the Terraform configuration"
>>
[master (root-commit) 1aa96f7] Initial commit of the Terraform configuration
 Committer: Darshan Mangaldas Naik <SLP13366@blrsimplilearn.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 main.tf
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration>
```

3.4 Execute the following command to link the local repository to the remote repository:
git remote add origin <HTTPS_GITHUB_URL>



```
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration> git remote add origin https://github.com/[redacted]-terrafo
rm-module.git
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration> []
```

3.5 Run the following command to push the changes to the remote repository:
git push -u origin master



```
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration> git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 288 bytes | 288.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/DarshanNaik547/my-terraform-module/pull/new/master
remote:
To https://github.com/DarshanNaik547/my-terraform-module.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\SLP13366\Desktop\Terraform1\my-terraform-configuration> []
```

By following these steps, you will successfully set up a private module registry on Terraform Cloud, publish a module, and use it in a Terraform configuration.