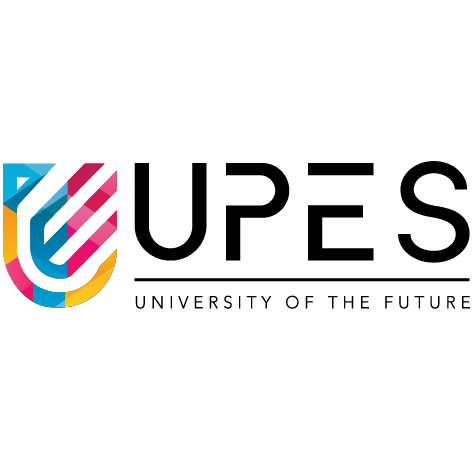
**School of Computer Science**

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**

**DEHRADUN, UTTARAKHAND**



**System Provisioning and Configuration Management Lab**

**Lab File**

**(2024)**

**for**

**6th Semester**

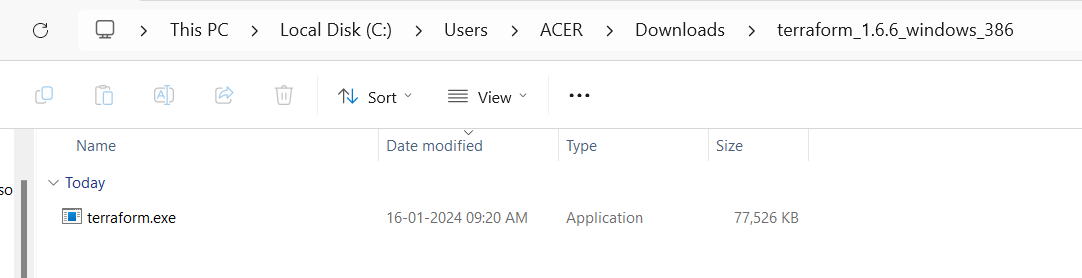
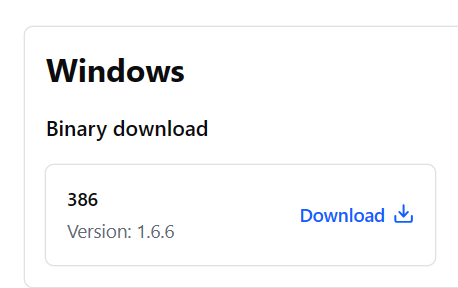
|  |  |
| --- | --- |
| Submitted To:  Dr. Hitesh Kumar Sharma | Submitted By:  Deepanshi Gupta  B. Tech. CSE DevOps  [6th Semester]  Sap id- 500090982  Batch - 1 DevOps |

**LAB EXERCISE 1**

INSTALL TERRAFORM ON WINDOWS

Download Terraform File for Windows

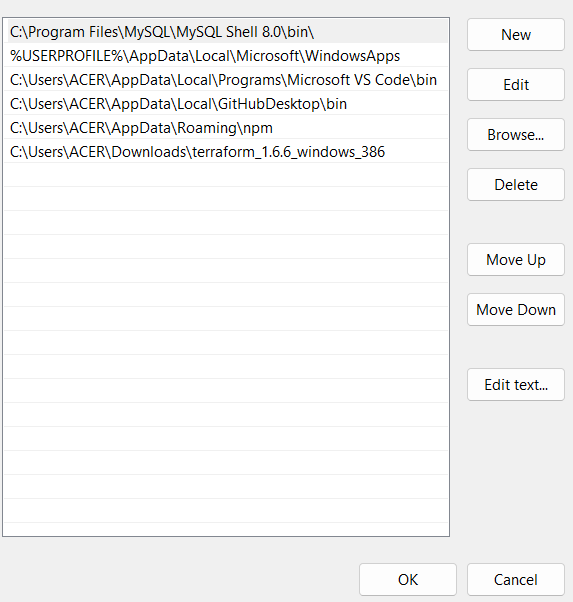
1. Browse to the [Download Terraform](https://developer.hashicorp.com/terraform/downloads) page.
2. Select the Windows tab under the **Operating System** heading. The latest version is preselected.



1. Choose the binary to download. Select 386 for 32-bit systems or [AMD64](https://phoenixnap.com/glossary/amd64) for 64-bit systems. Choose the download location for the zip file if the download does not start automatically.
2. Unzip the downloaded file. For example, use the *C:\terraform* path. Remember this location so you can add the path to the environment variables.

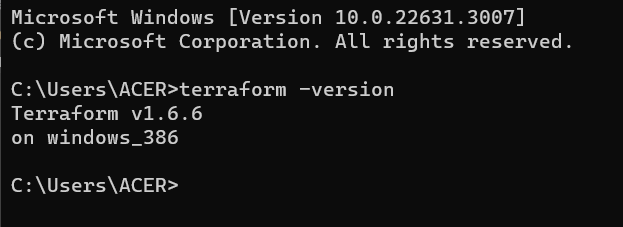
Add Terraform Path to System Environment Variables

1. Open the start menu, start typing *environment* and click **Edit system environment variables**. The System Properties window opens.
2. Click the **Environment Variables...** button.
3. Select the **Path variable** in the System variables section to add terraform for all accounts. Alternatively, select **Path** in the **User variables** section to add terraform for the currently logged-in user only. Click **Edit** once you select a Path**.**
4. Click **New** in the edit window and enter the location of the Terraform folder.
5. Click **OK** on all windows to apply the changes.



# Verify Windows Terraform Installation

1. Open a new command-prompt window.
2. Enter the command to check the Terraform version: terraform -version

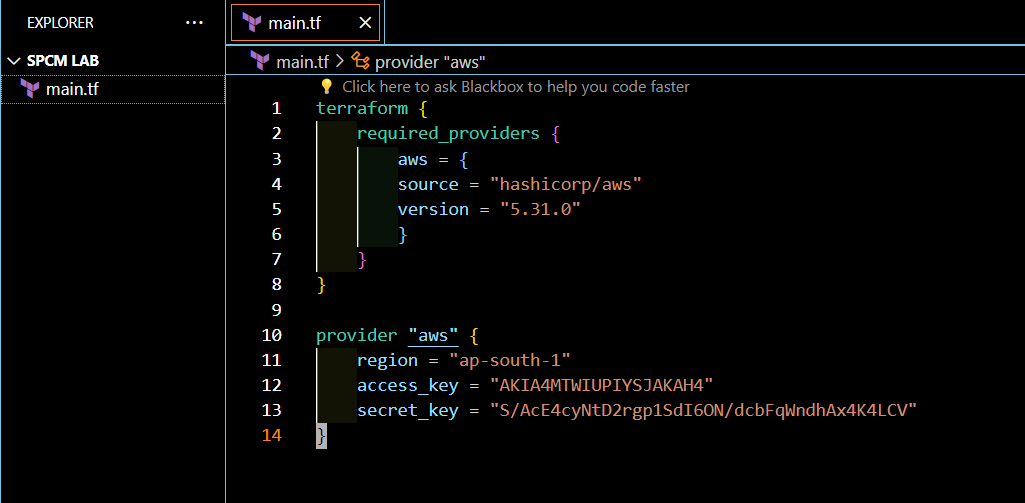


**LAB EXERCISE 2-3**

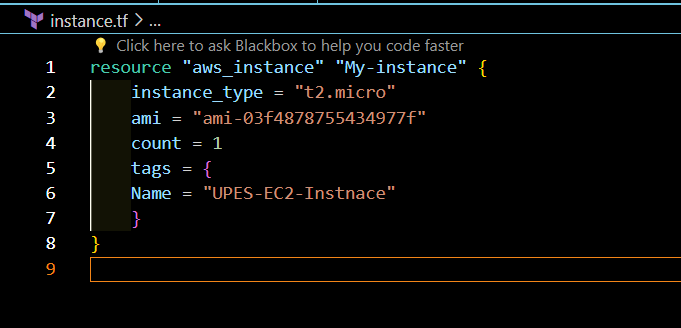
TERRAFORM AWS PROVIDER AND IAM USER SETTINGS

Steps:

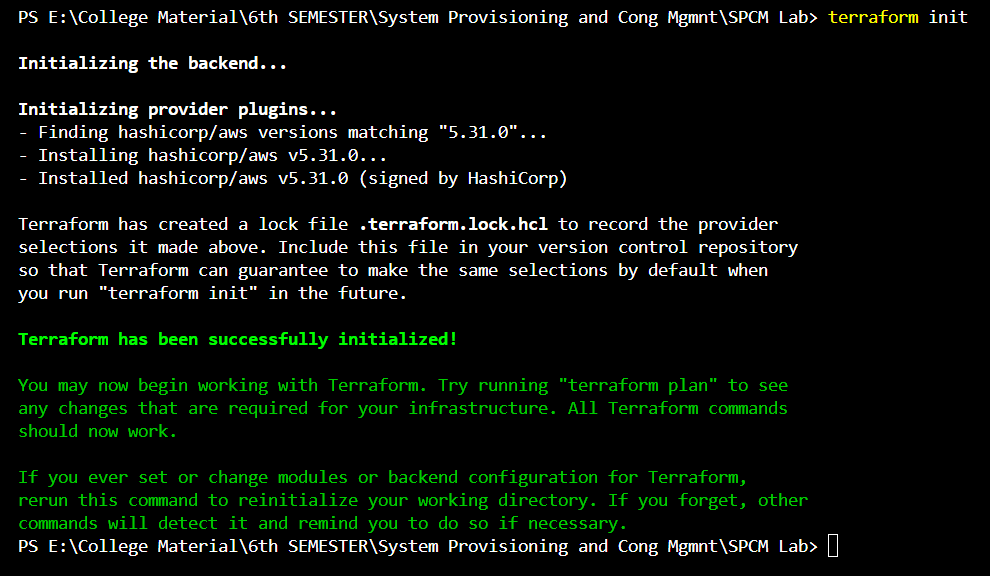
1. Create a new directory.
2. Create a new Terraform Configuration file – **main.tf** with the following content.



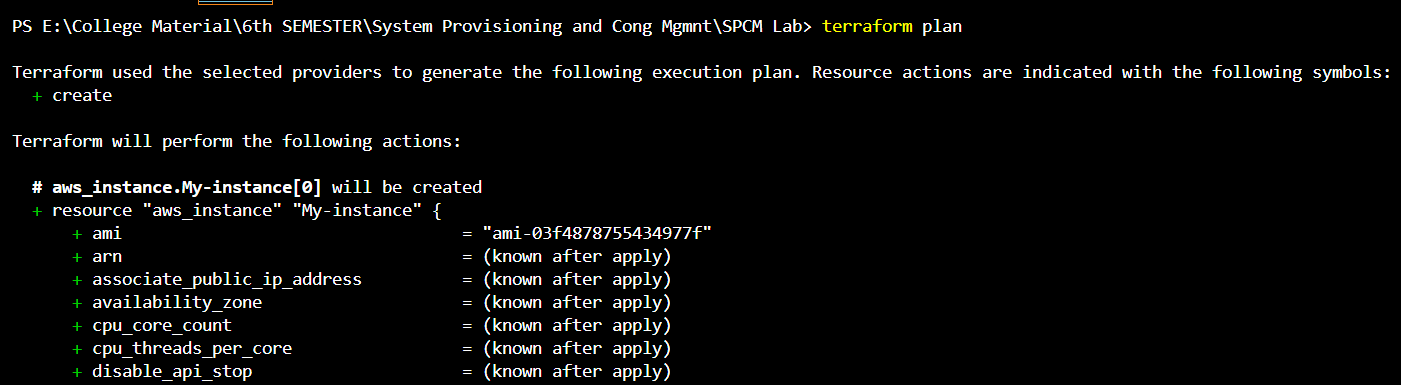
1. Create another Terraform Configuration File for EC2 instance – **instance.tf** with the following content.



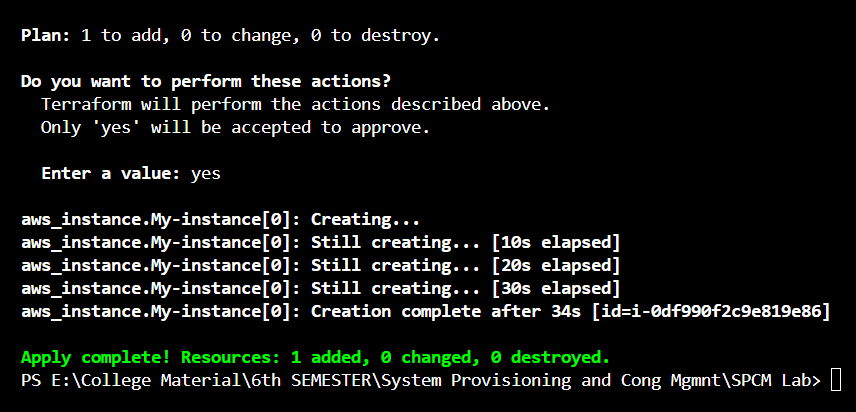
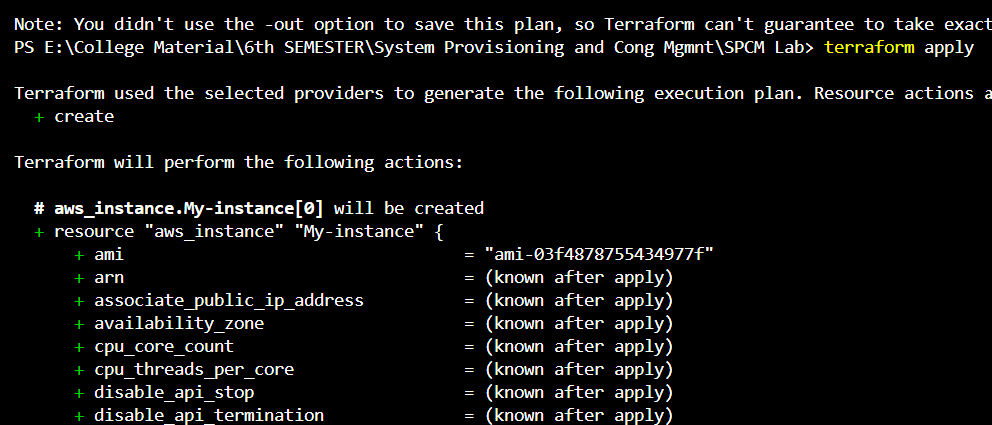
1. Initialize the terraform working directory using – **terraform init** command.



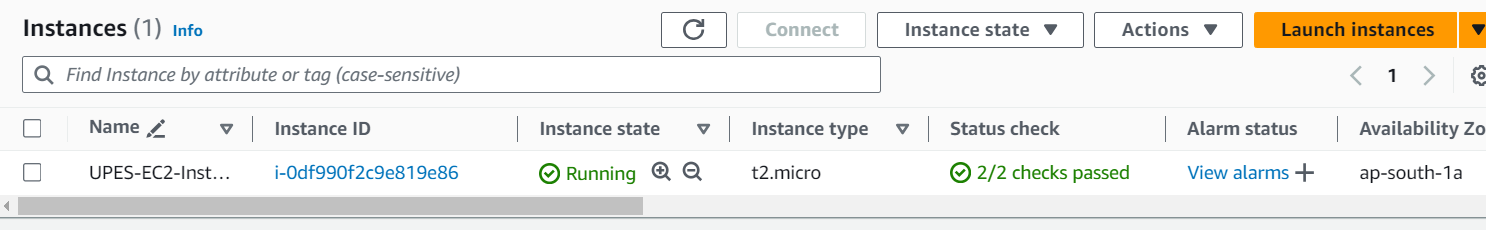
1. Review the Plan using – **terraform plan** command.



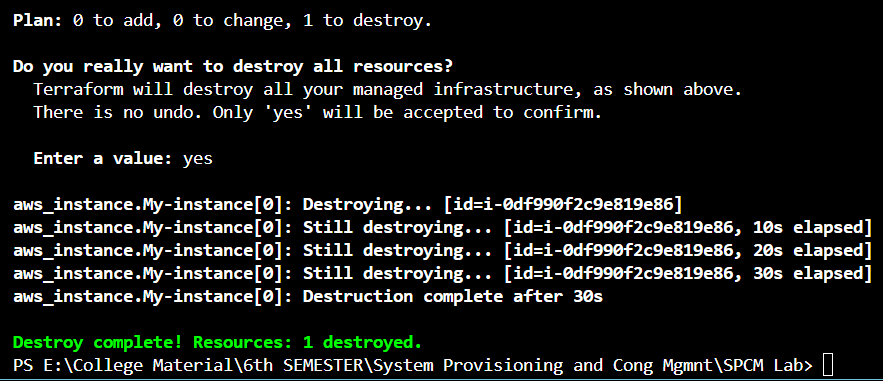
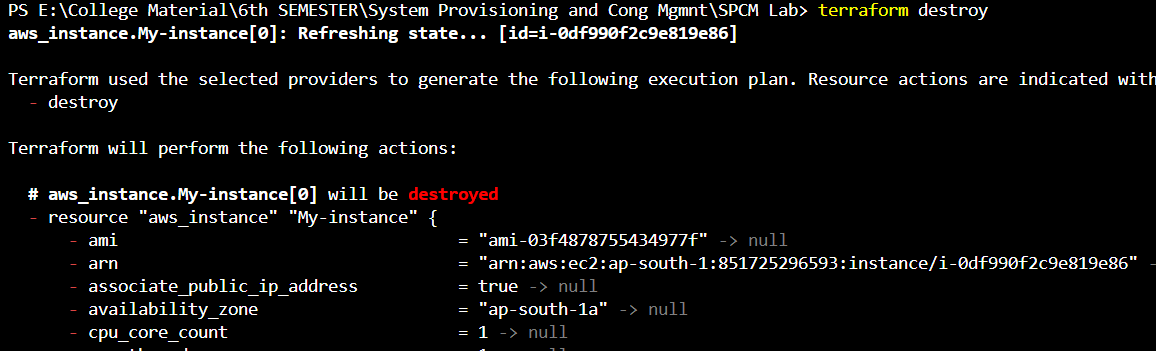
1. Apply the changes to create the AWS resources using - **terraform apply** command, and type yes when prompted.



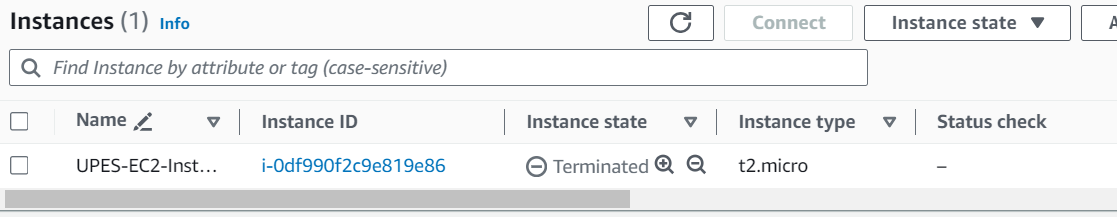
1. **Verify Resources** i.e. after the terraform apply command successfully executes, go to your AWS Management Console and navigate to the EC2 dashboard to verify whether the EC2 instance is successfully created.



1. **Clean up** the resources using – **terraform destroy** command**,** type yes when prompted.



1. Now again, go to your AWS Management Console to check whether the instance has successful terminated.

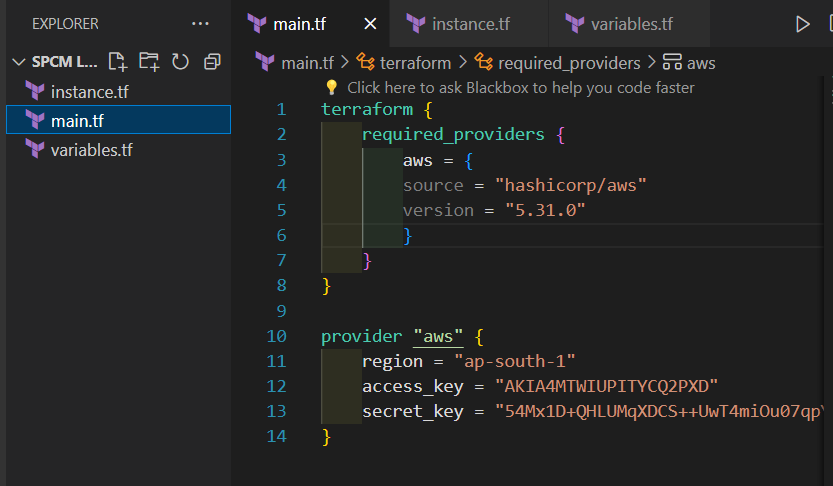


**LAB EXERCISE 4**

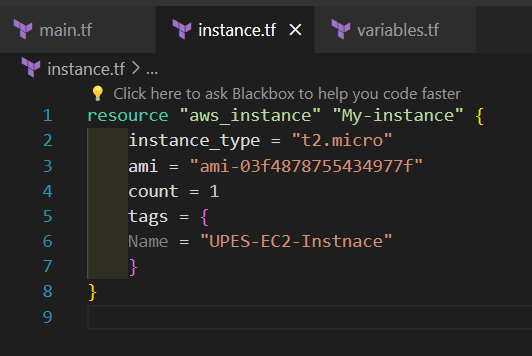
TERRAFORM VARIABLES

Steps:

1. Create a terraform directory for your project.
2. Create Terraform Configuration file – **main.tf**, that includes the following content.



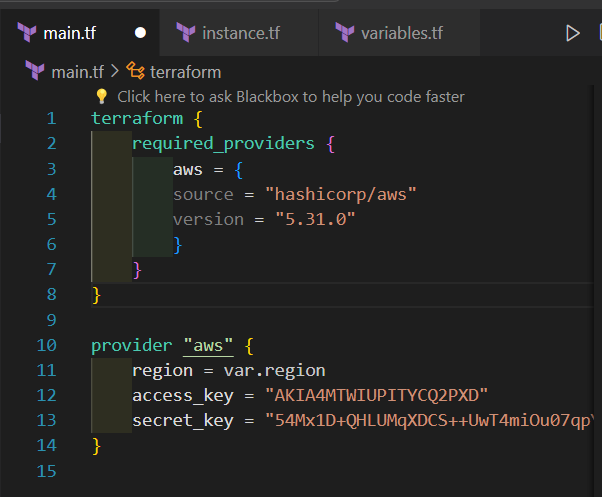
1. Now create another Configuration file for EC2 instance – **instance.tf** that includes the following content.

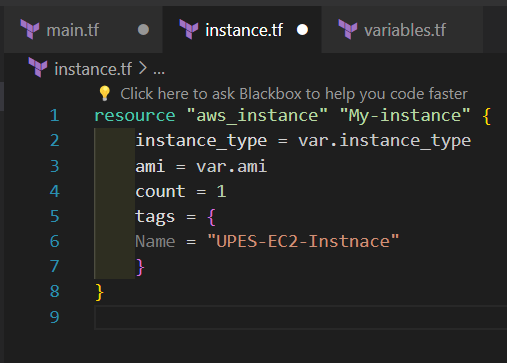


1. Create a new file named – **variables.tf** that defines the variables, for region, ami, and instance\_type.



1. Now modify the **main.tf** and **instance.tf** files to use the variables.

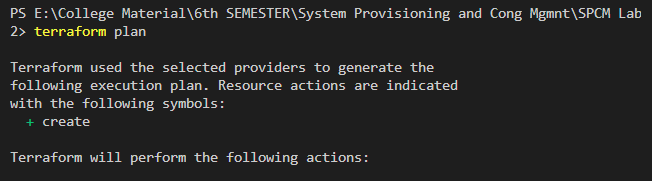




1. Now initialize the terraform working directory using – **terraform init** command.

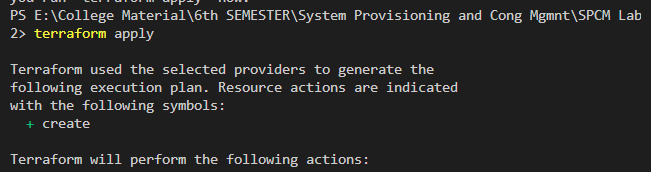


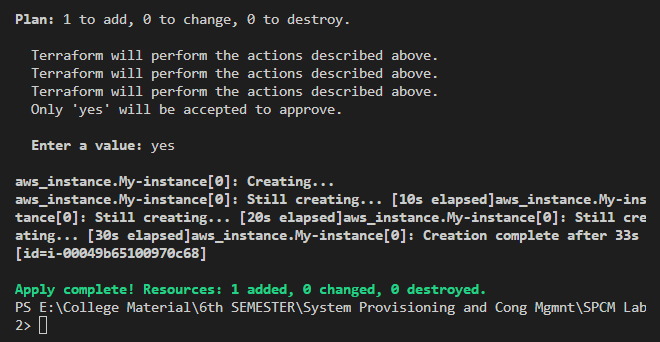
1. Review the plan to check what the terraform will do using - **terraform plan** command.





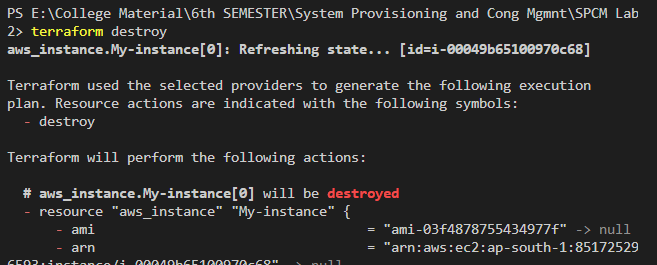
1. Apply the changes using – terraform apply command. Click Yes whenever prompted.

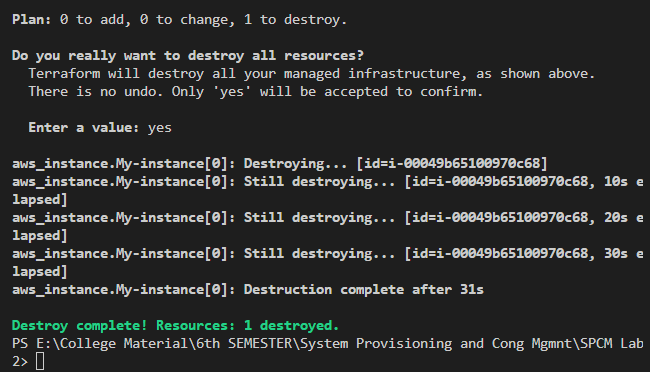


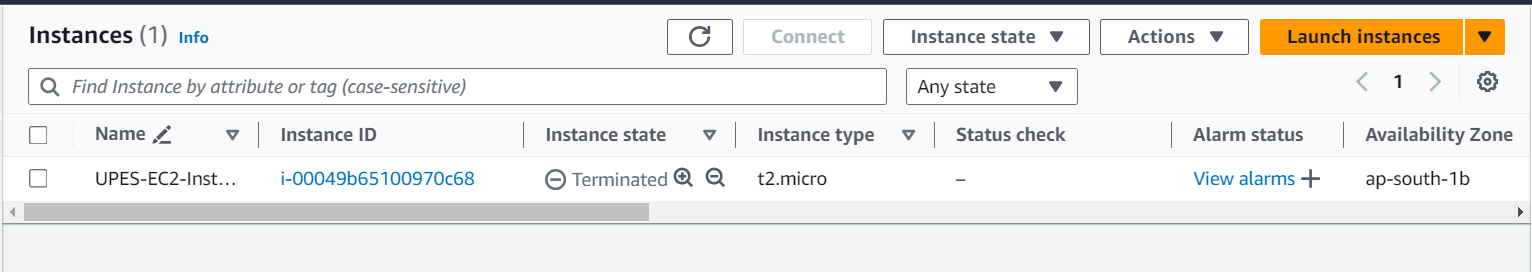




1. Clean up the resources using – terraform destroy command.





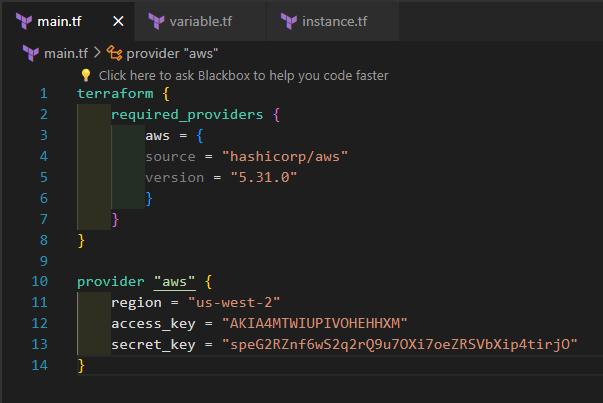


**LAB EXERCISE 5**

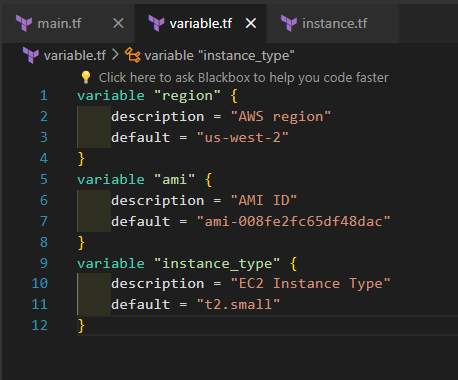
TERRAFORM VARIABLES WITH COMMAND LINE ARGUMENTS

STEPS:

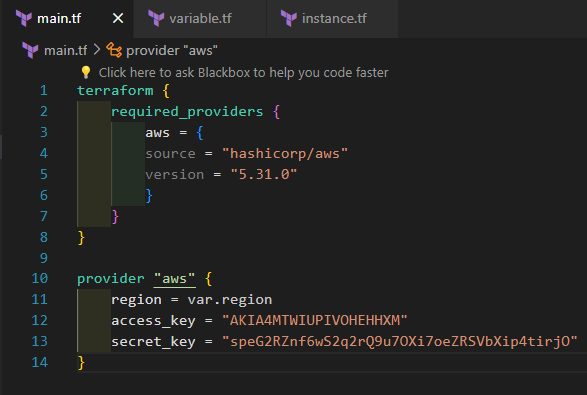
1. Create Terraform directory.
2. Create terraform configuration files – main.tf, instance.tf, variables.tf as follows:

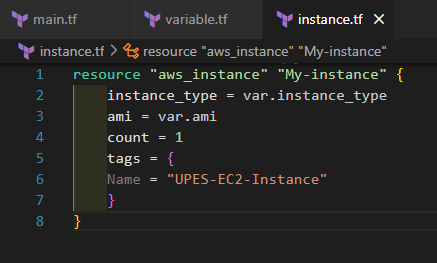






1. Modify the main.tf and instance.tf files as follows:

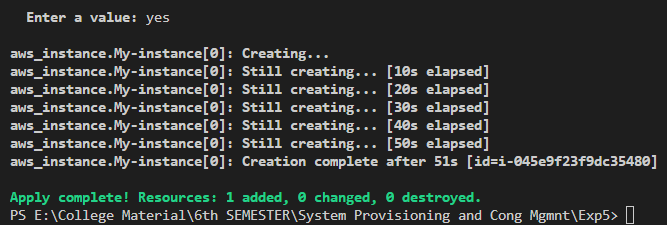
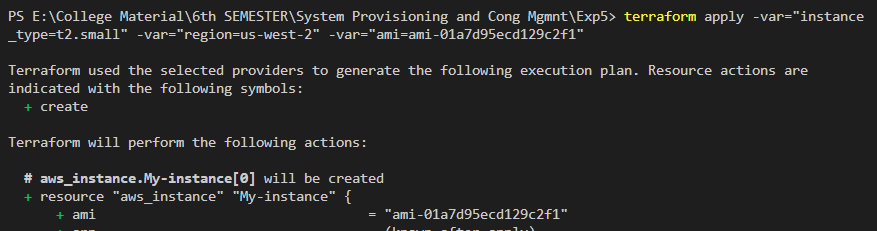




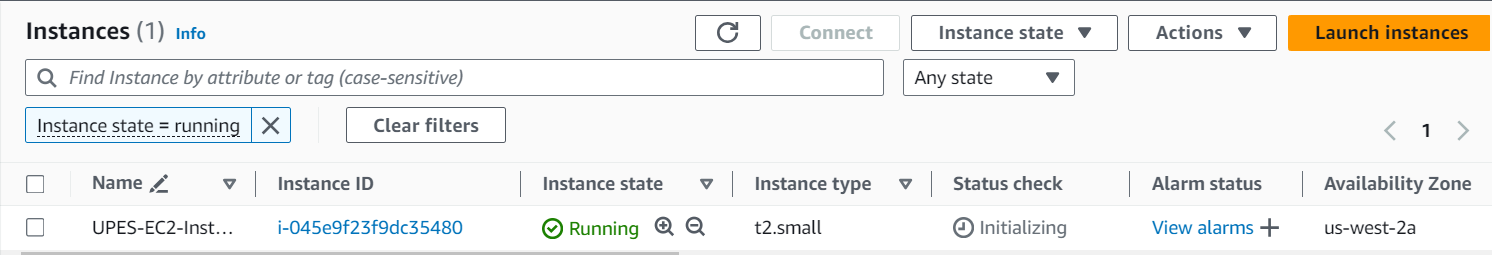
1. Initializing the terraform project directory using – terraform init.



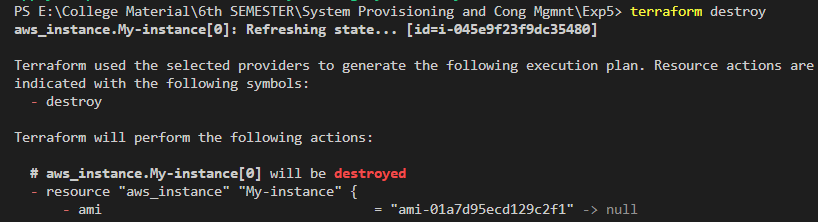
1. Run the terraform apply command with command line arguments to set the variables values:

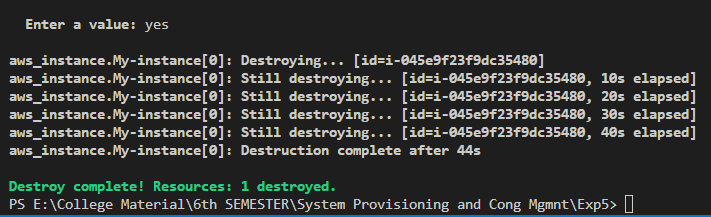


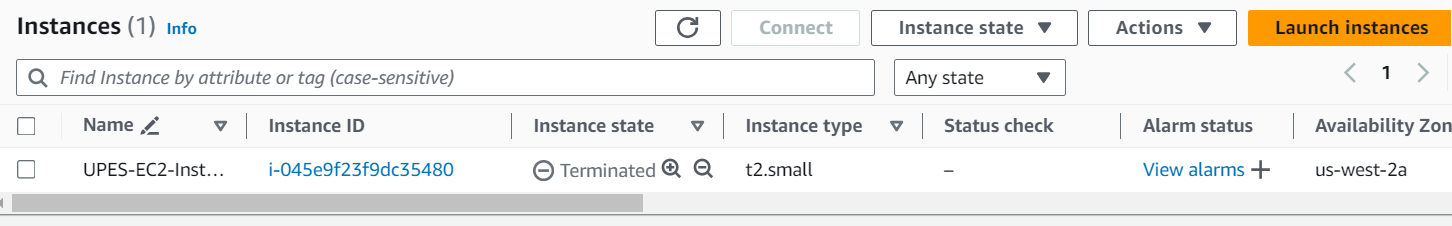
1. Observe how the command line arguments dynamically set the variables values during the apply process.



1. Clean up the resources using – terraform destroy.





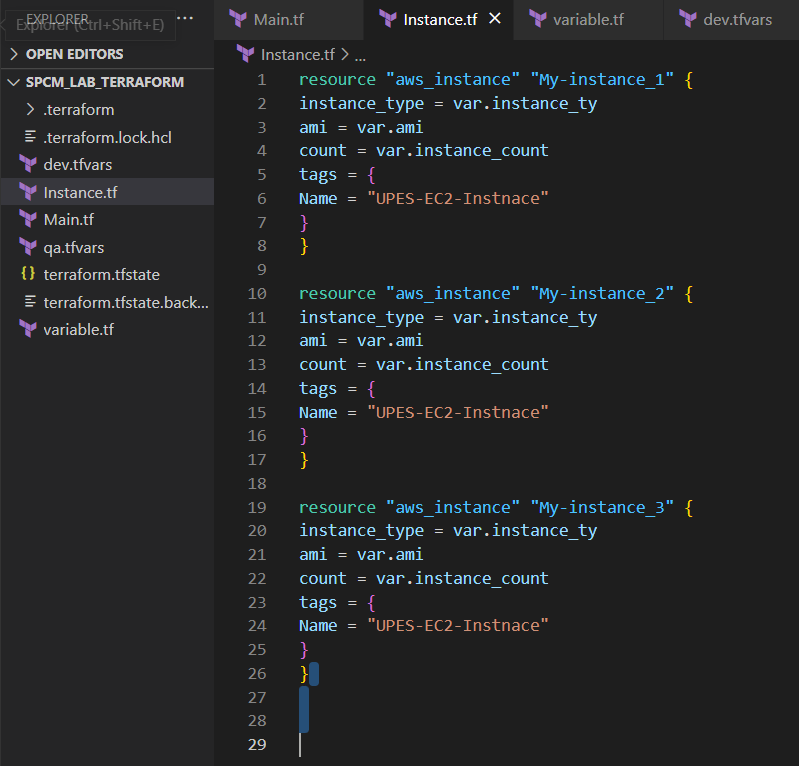


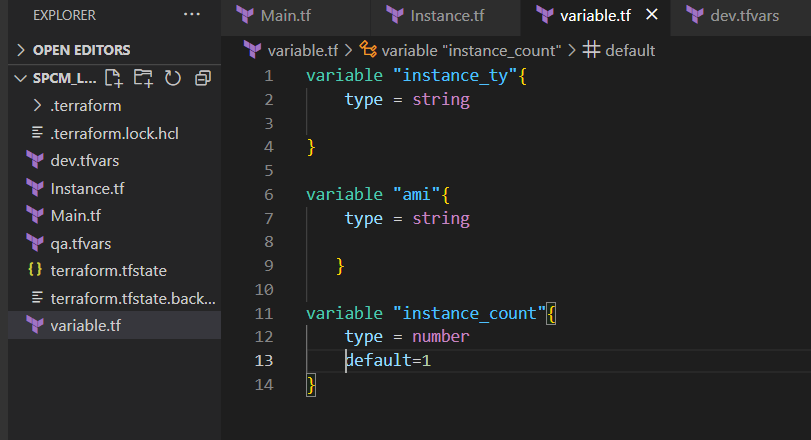
**LAB EXERCISE 5**

TERRAFORM VARIABLES WITH COMMAND LINE ARGUMENTS

STEPS:

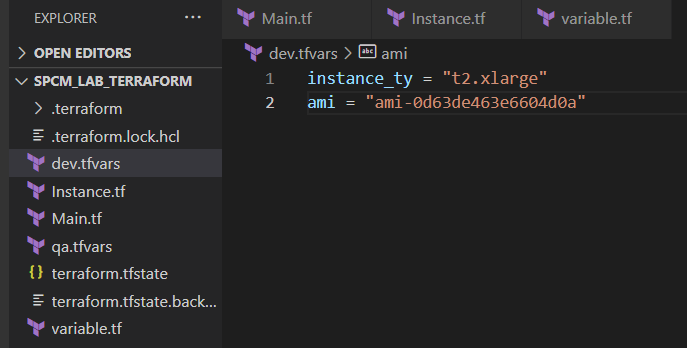
1. Create Terraform directory.
2. Create terraform configuration files – main.tf, instance.tf, variables.tf as follows:



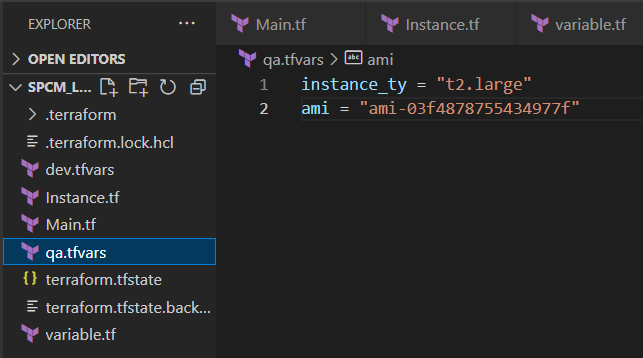


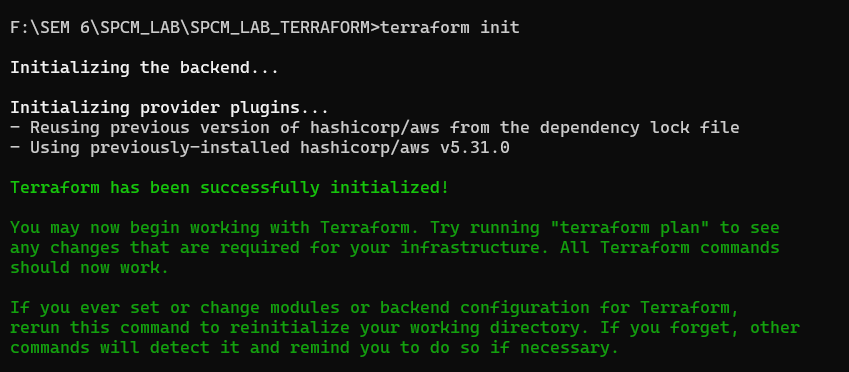
1. Create multiple tfvars files:

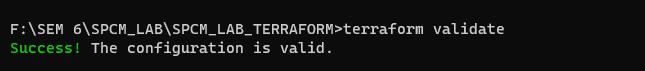
* dev.tfvars



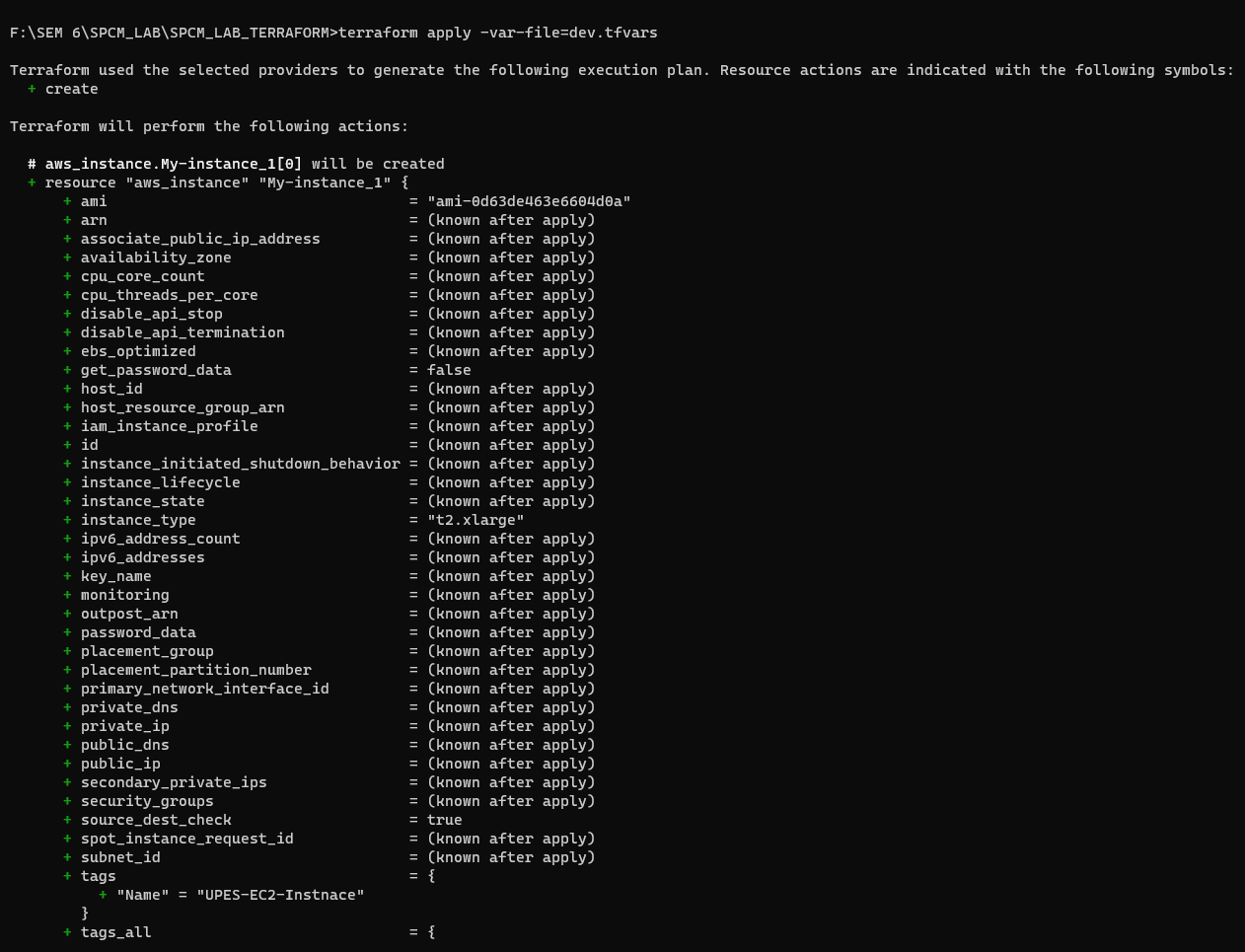
* qa.tfvars

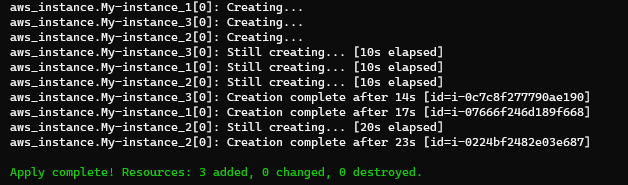


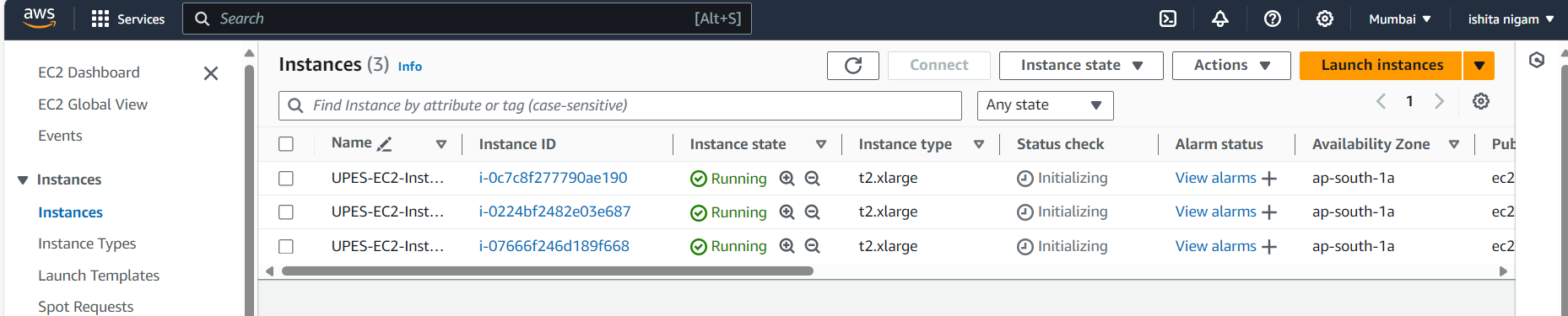
1. Initialize the terraform repository using terraform init.



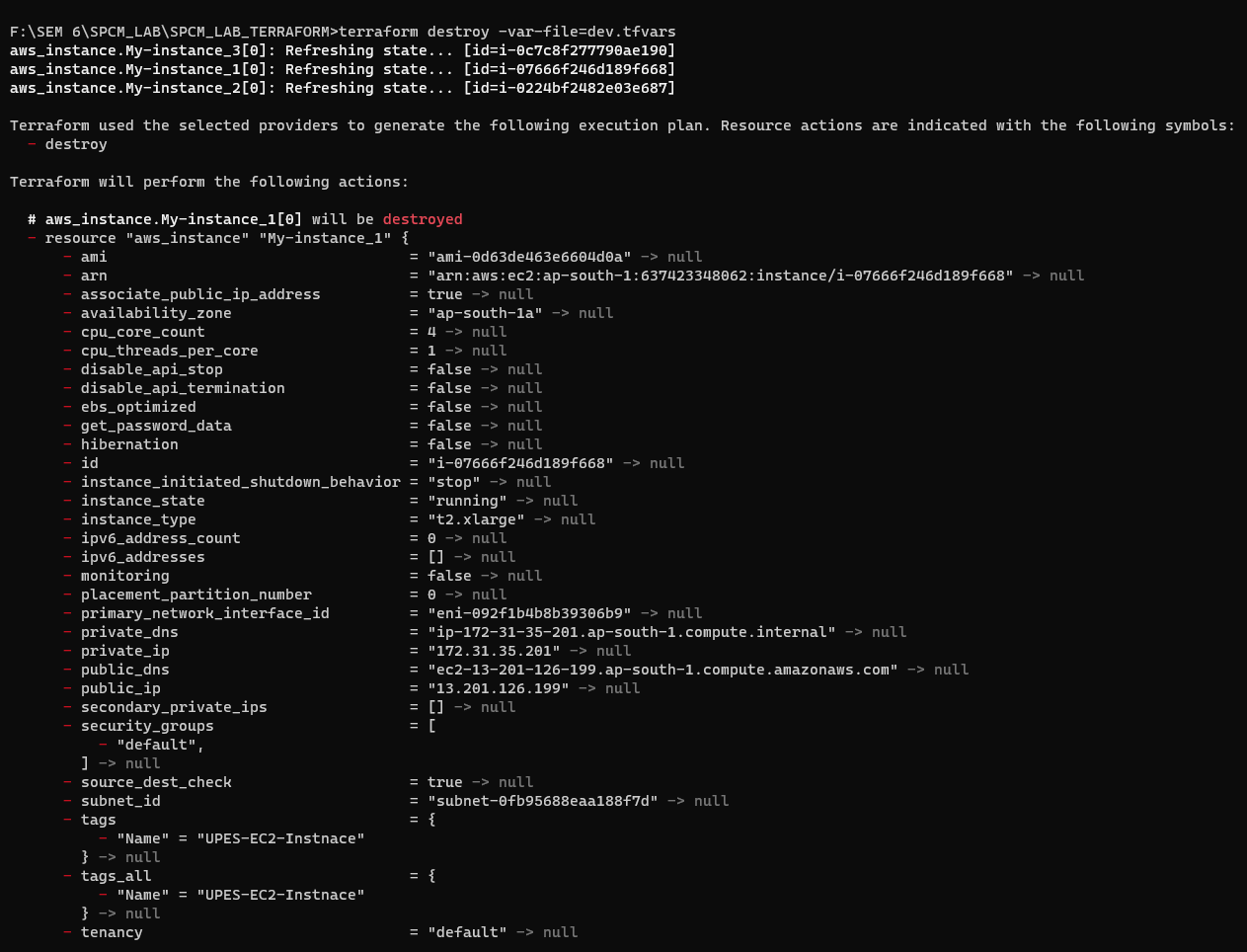
1. Apply for dev environment

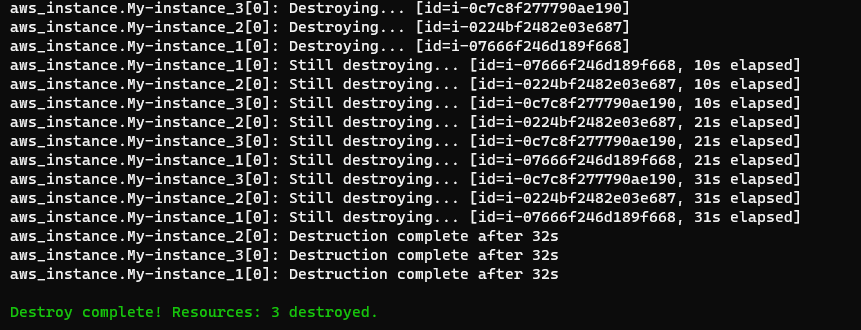


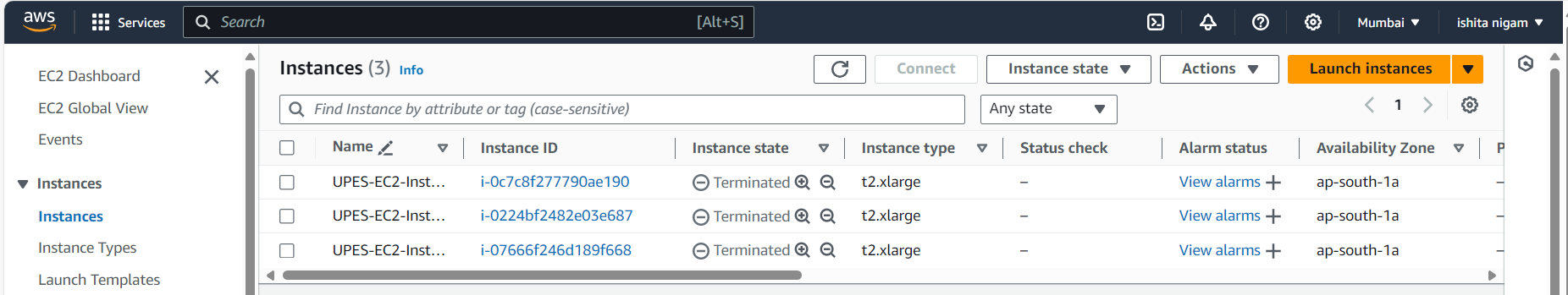




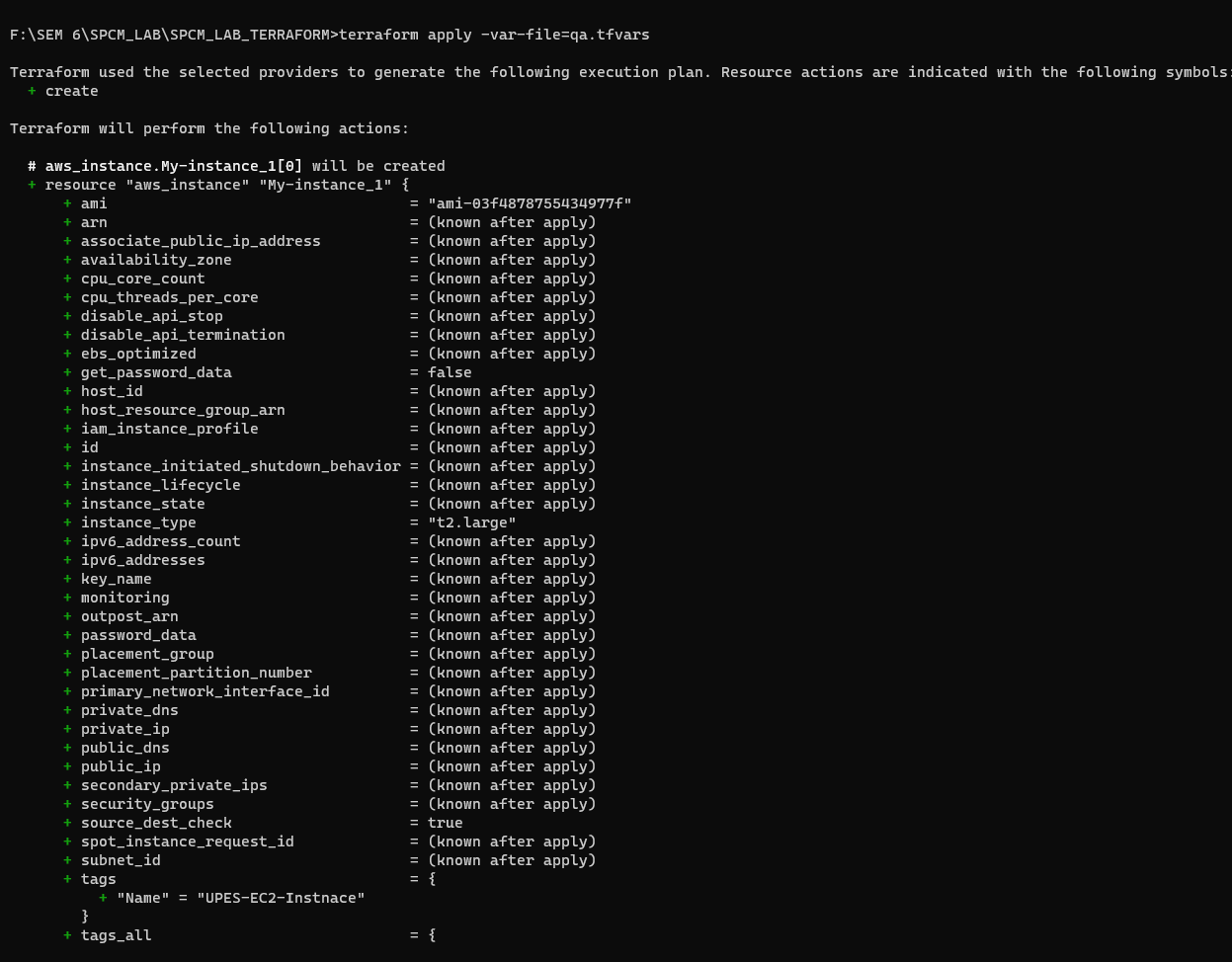
1. Destroy for dev environment.

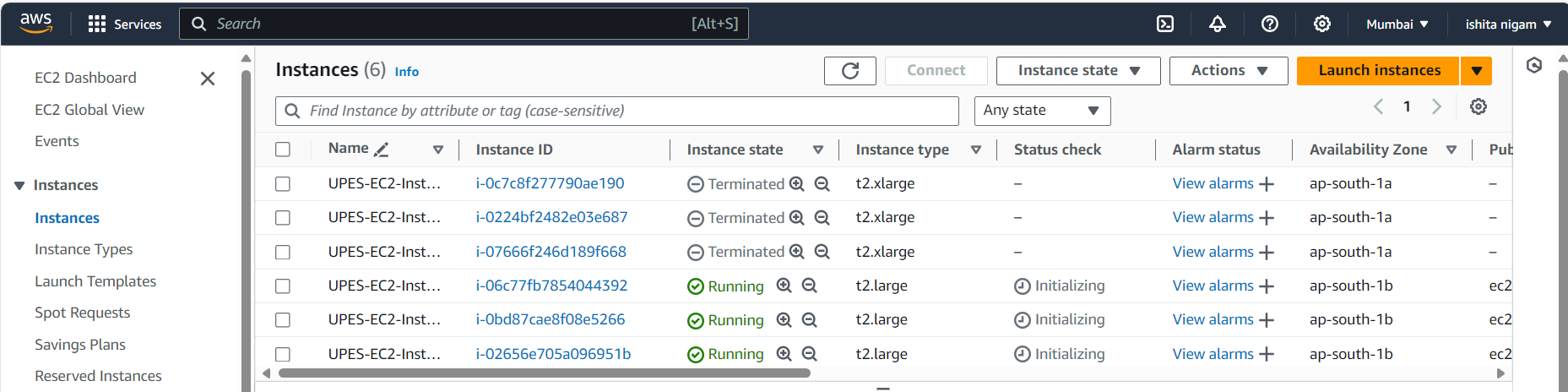




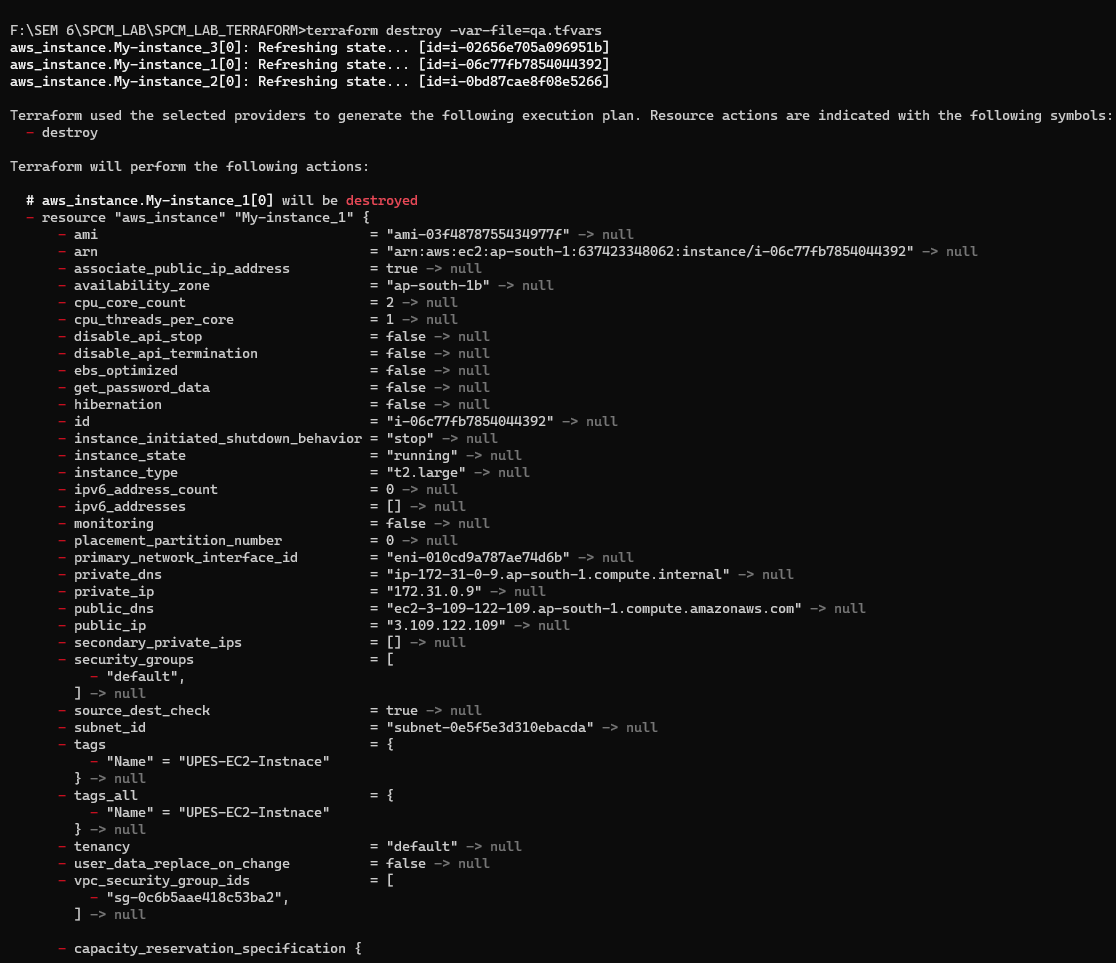


1. Apply for QA environment.

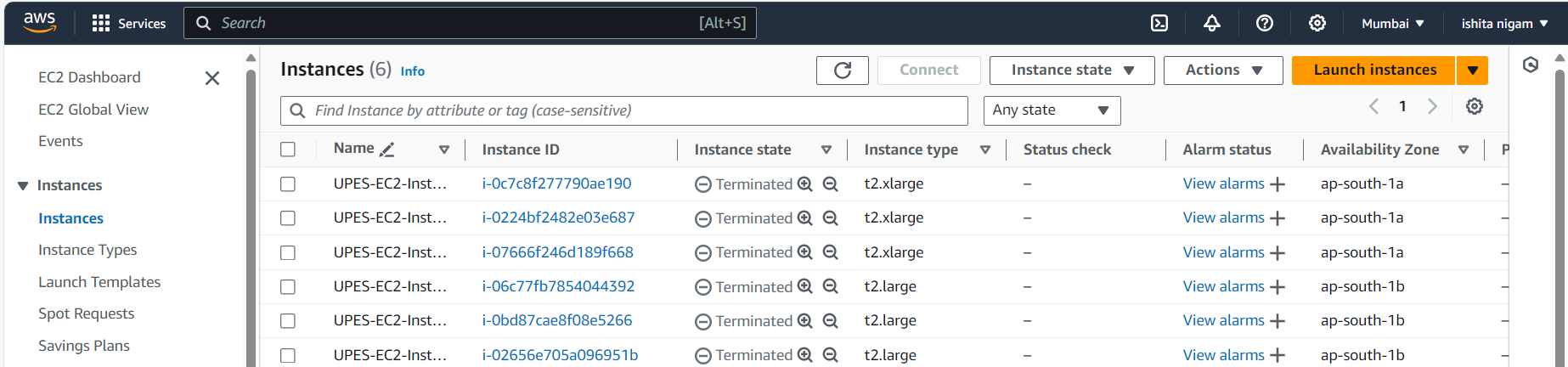




1. Destroy for QA environment.







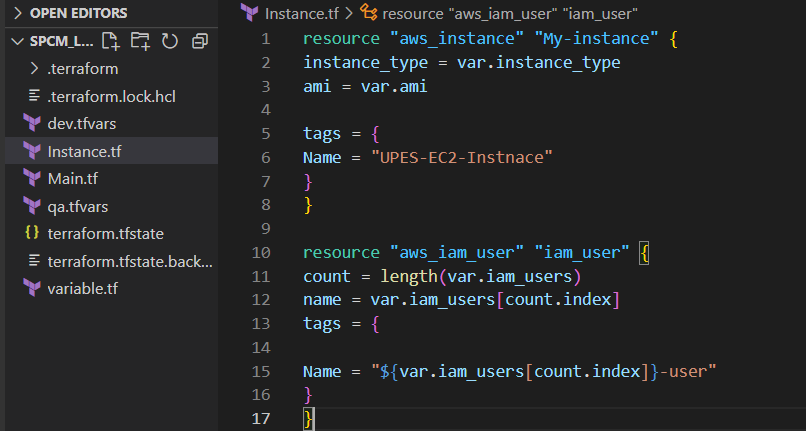
**LAB EXERCISE 7**

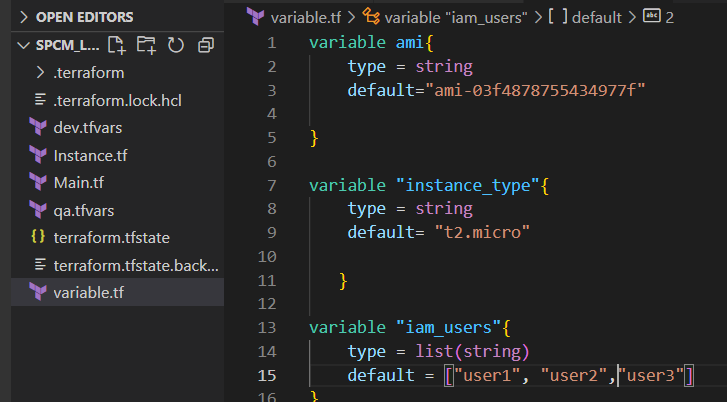
CREATING MULTIPLE IAM USERS IN TERRAFORM

STEPS:

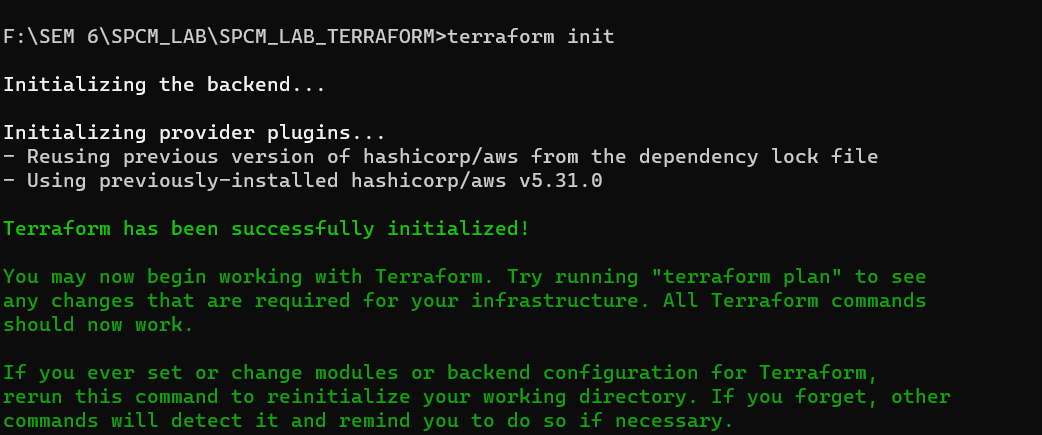
1. Create Terraform directory.
2. Create terraform configuration files – main.tf, instance.tf, variables.tf as follows:

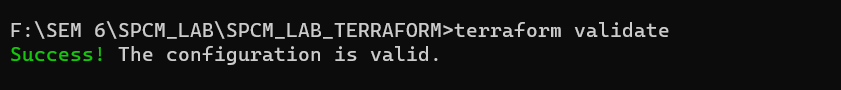




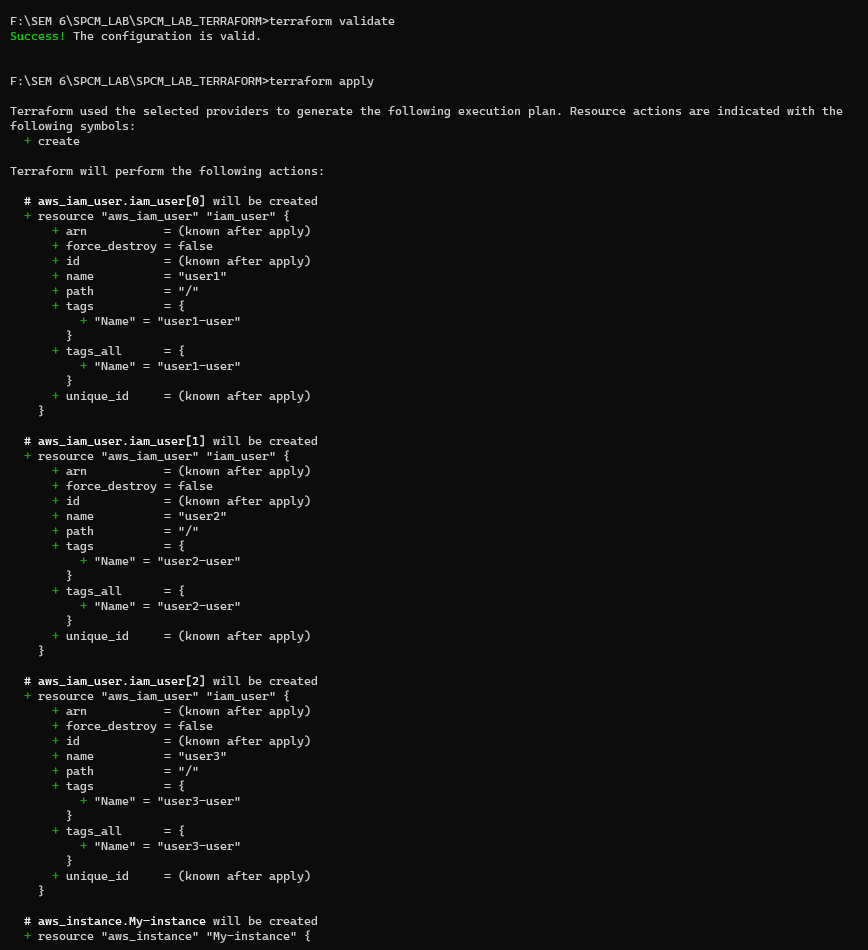


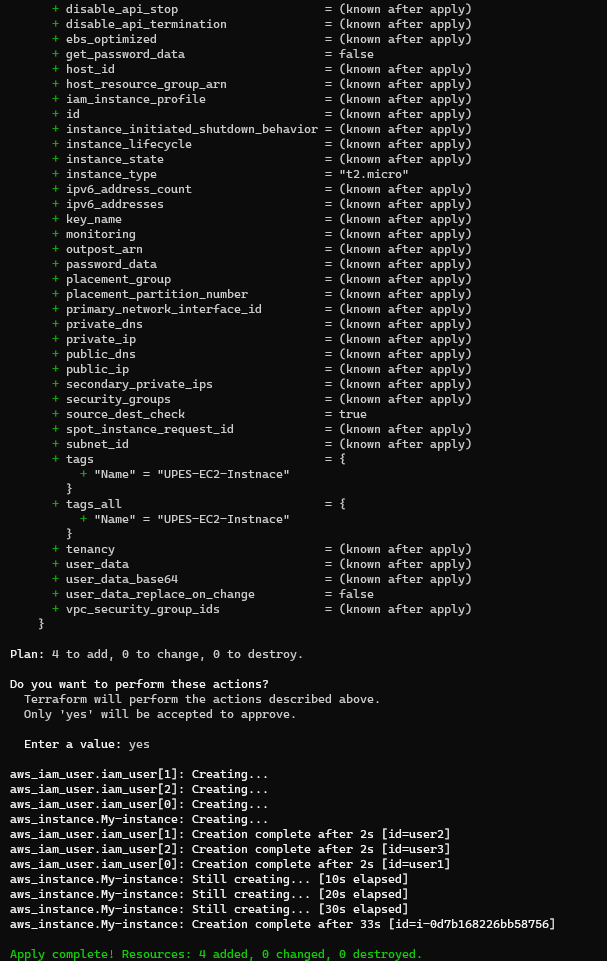
1. Initialize the terraform directory using terraform init.





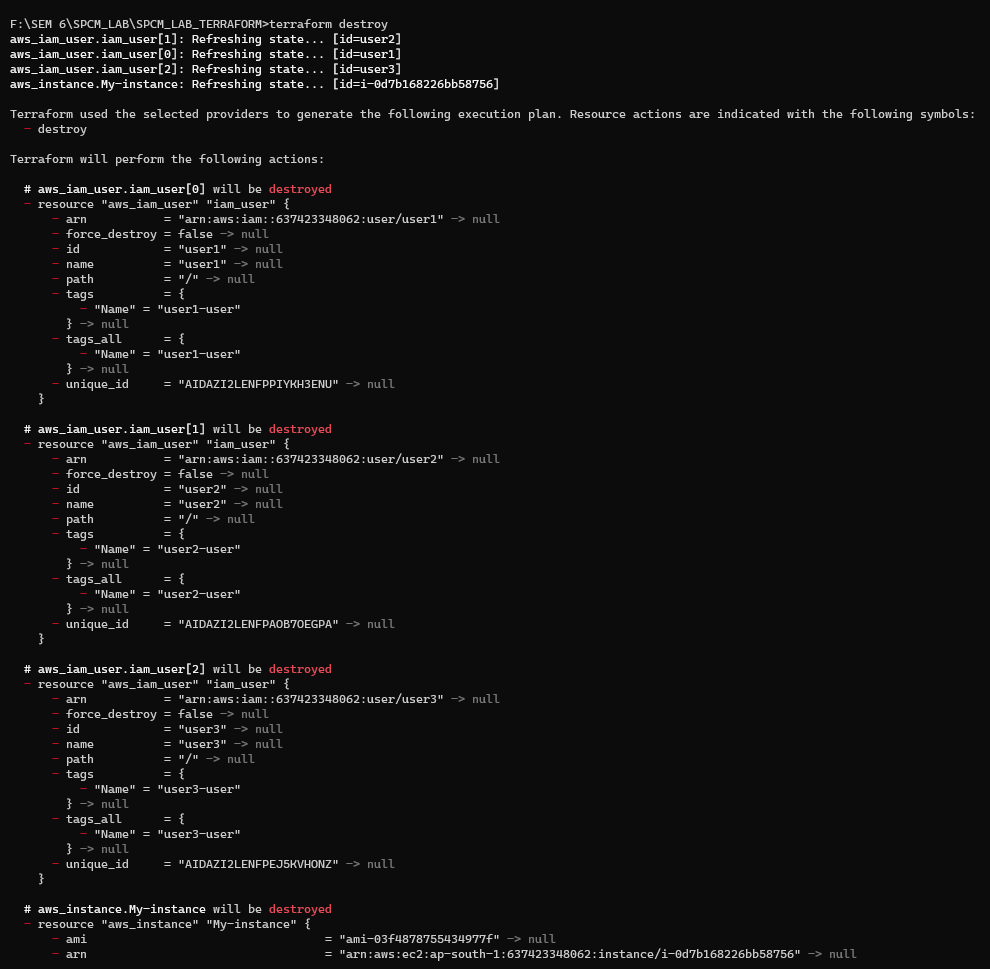
1. Terraform apply



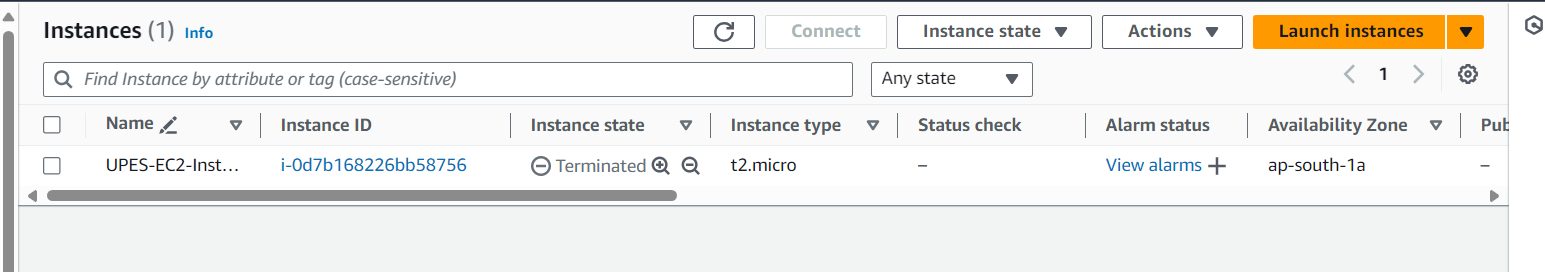




1. Terraform Destroy







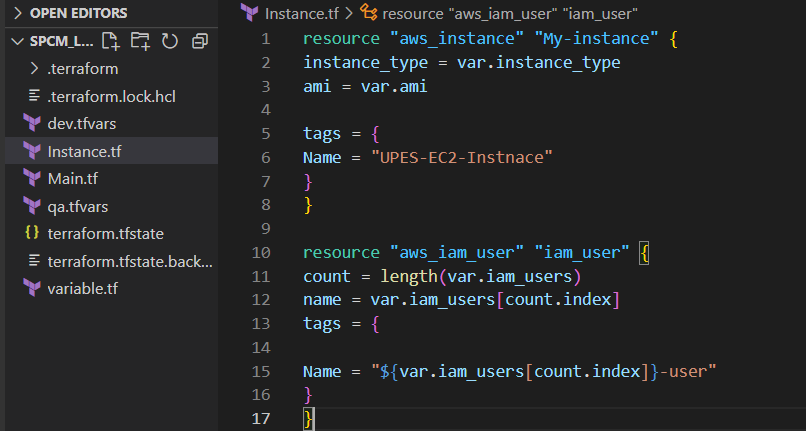
**LAB EXERCISE 8**

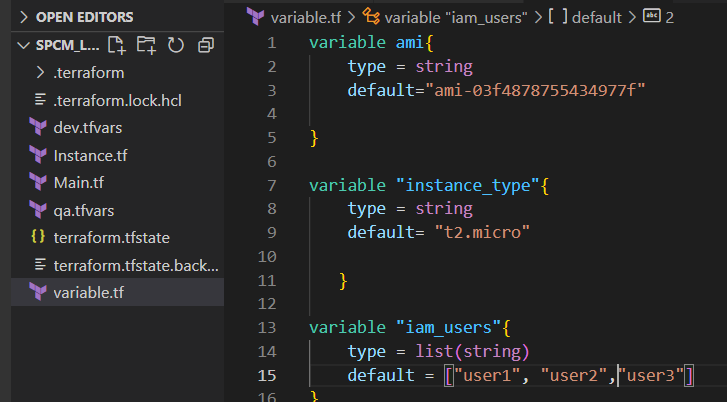
CREATING A VPC IN TERRAFORM

STEPS:

1. Create Terraform directory.
2. Create terraform configuration files – main.tf, instance.tf, variables.tf as follows:



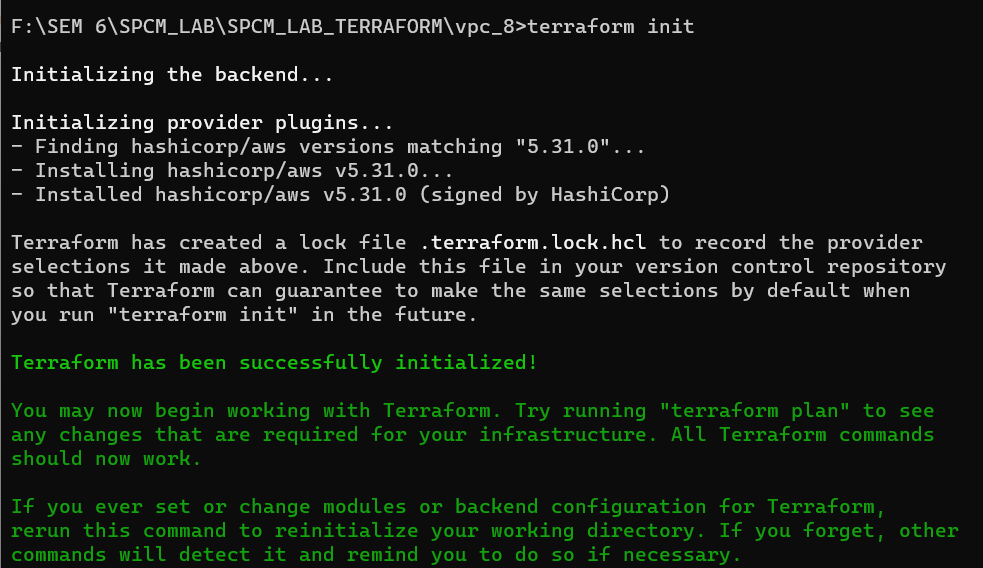


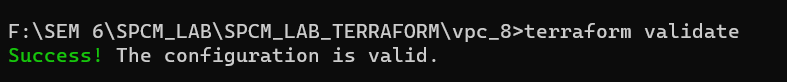


1. Create vpc.tf file.

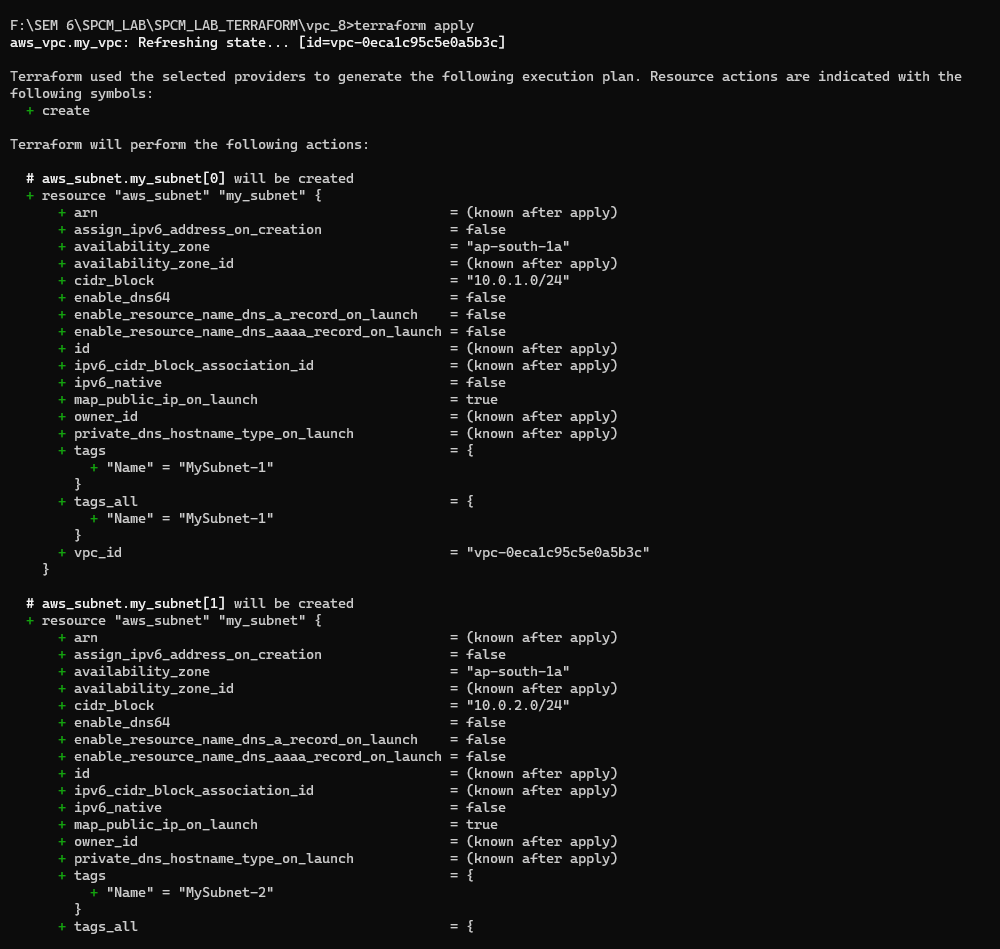


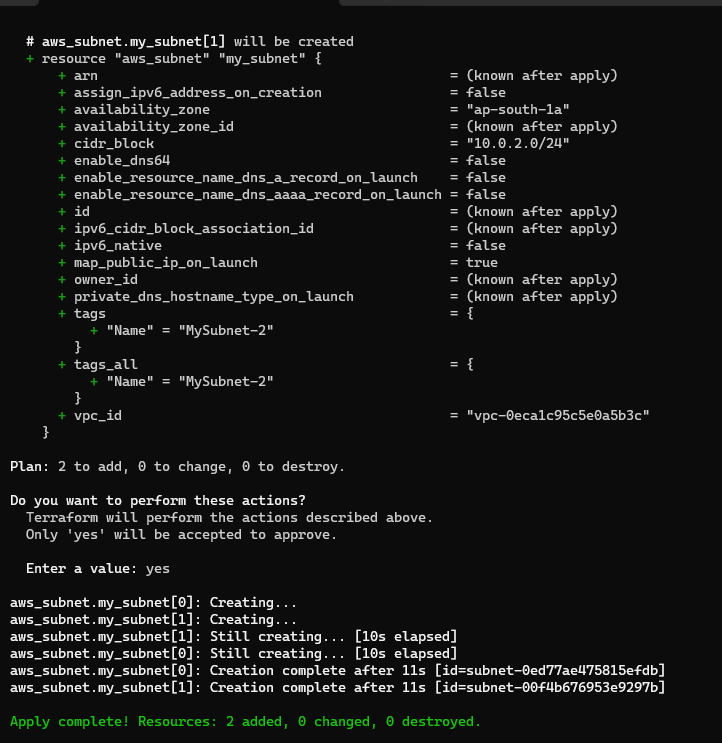
1. Run terraform init to initialize.

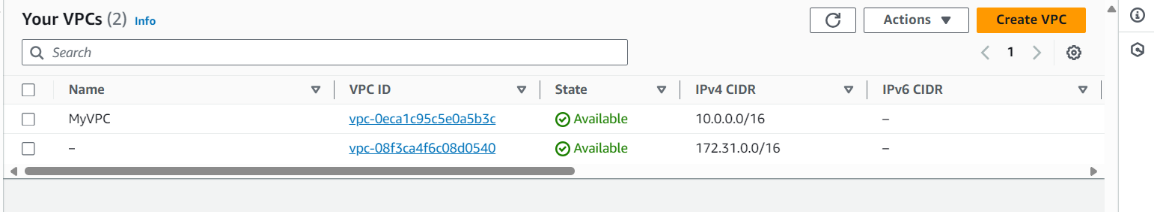


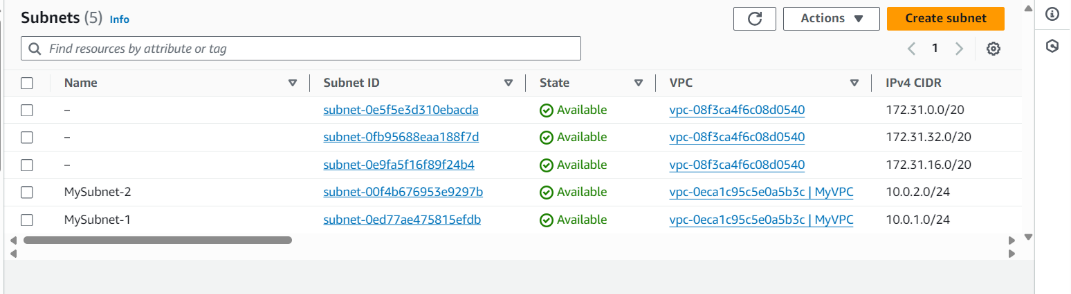


1. Run terraform apply command.

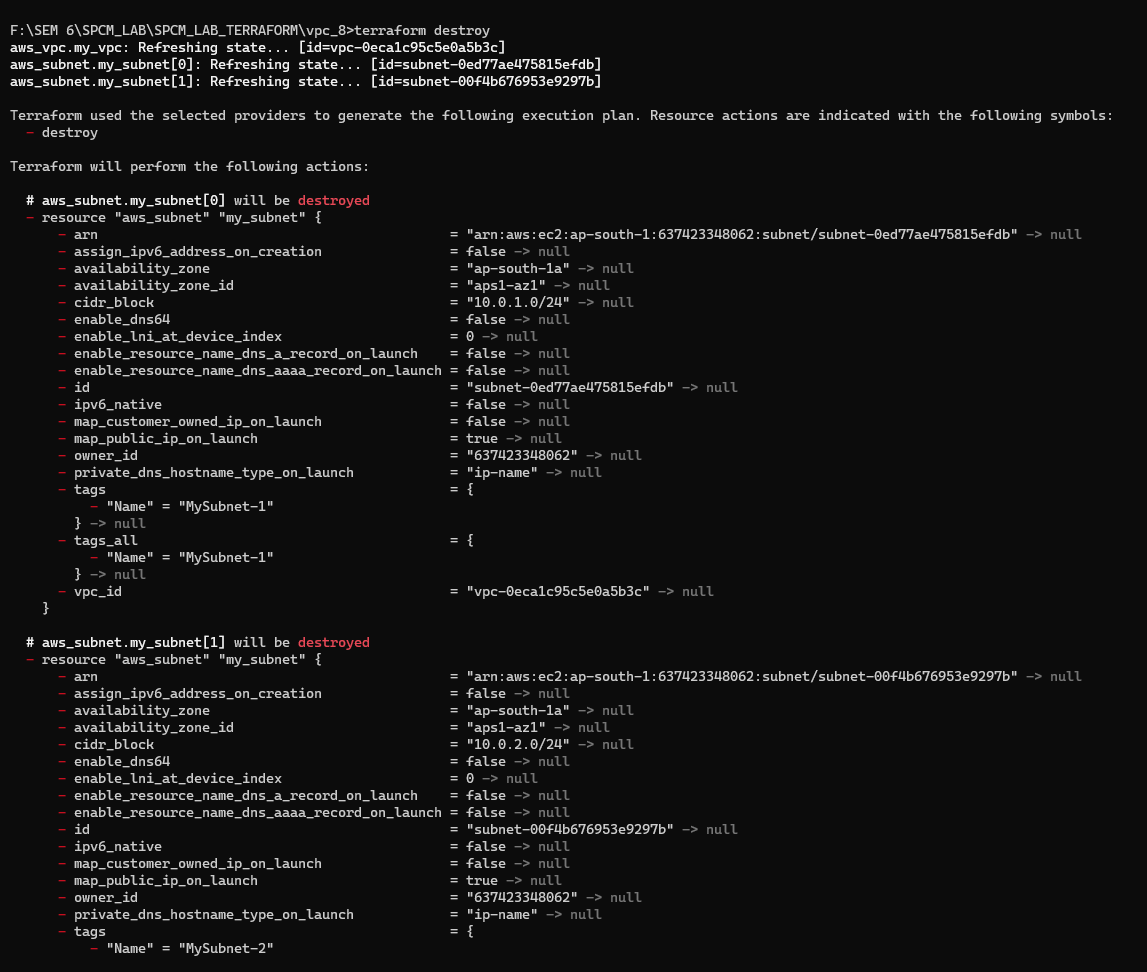


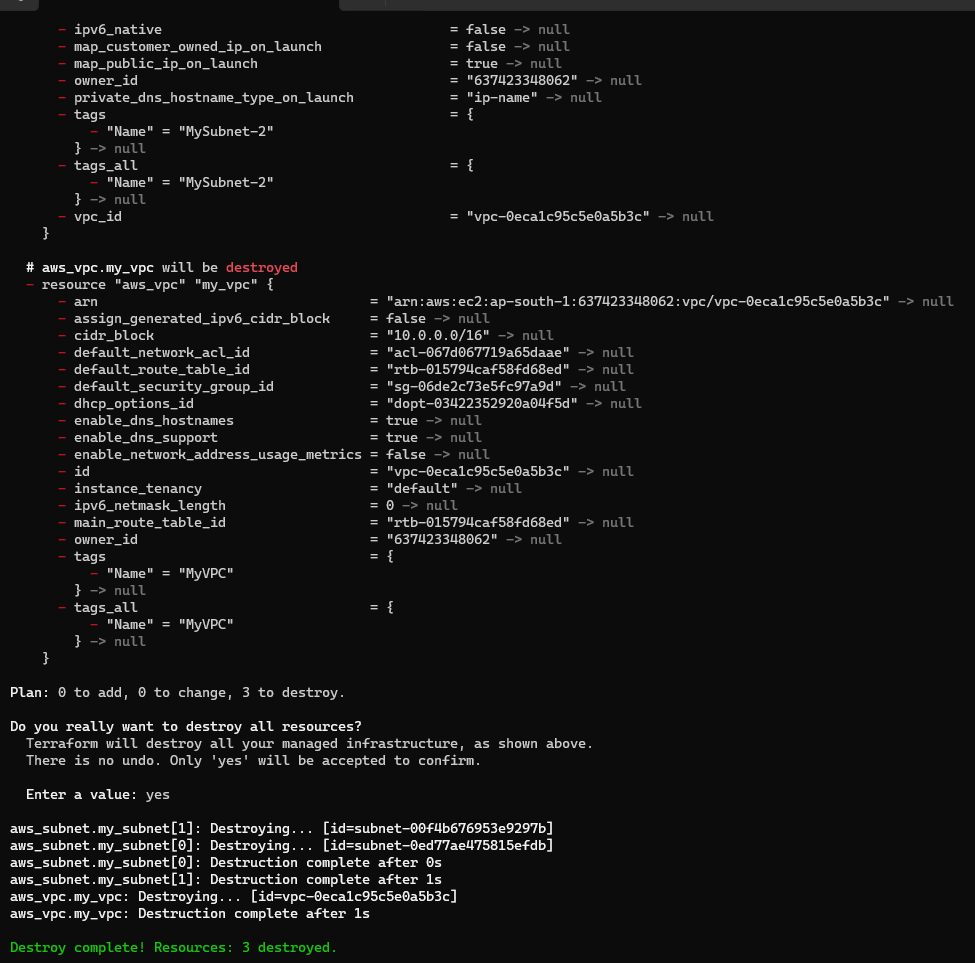




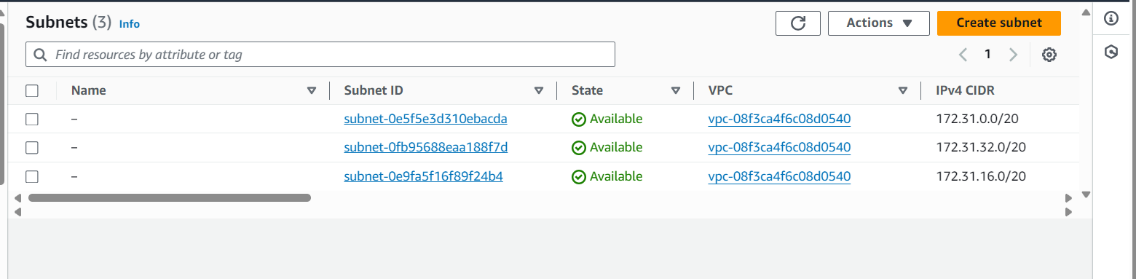


1. Run terraform destroy to destroy the vpc created.

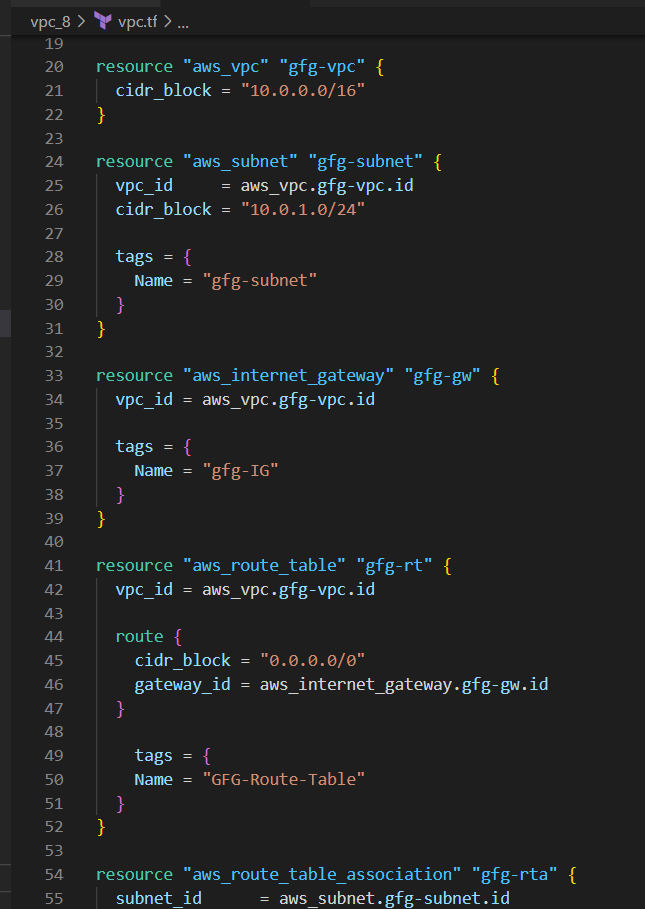


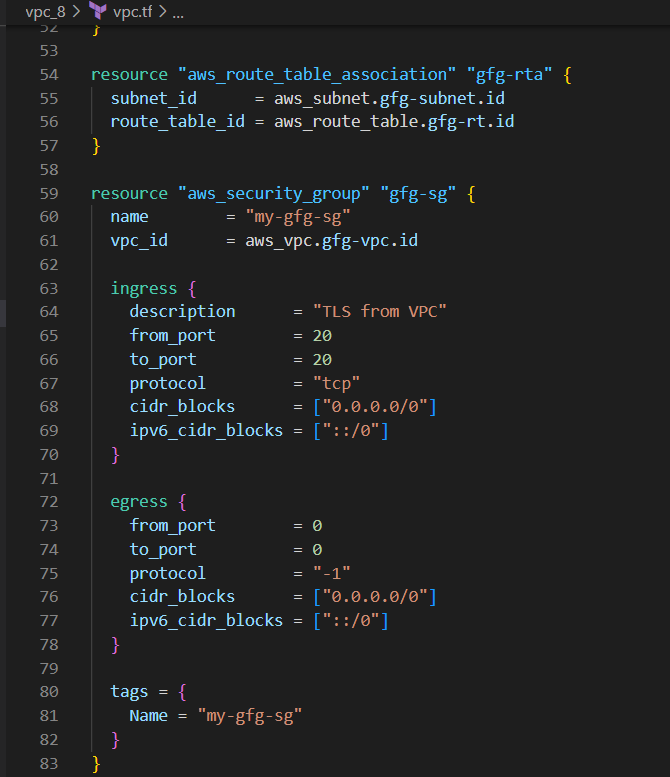




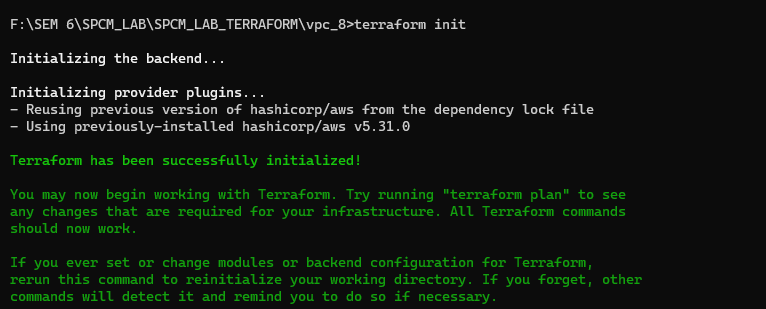


1. Update the vpc.tf file





1. Again initialize using terraform init.



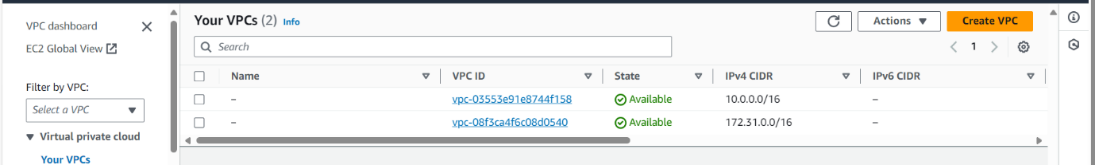
1. Run terraform apply command.



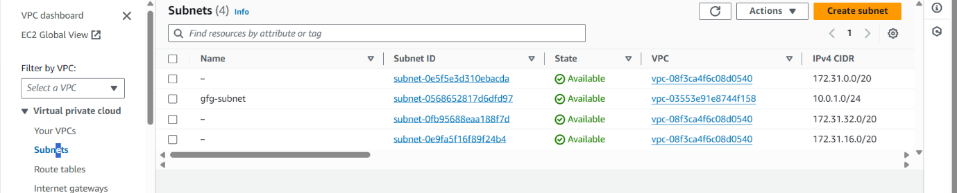




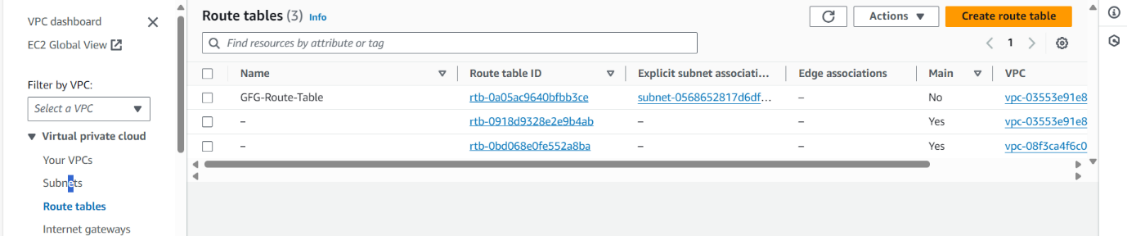
VPC:



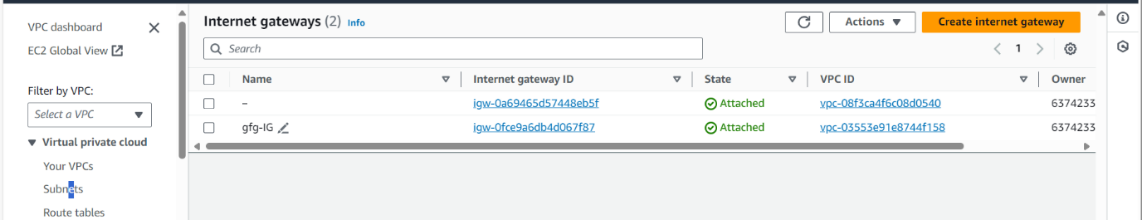
SUBNET:



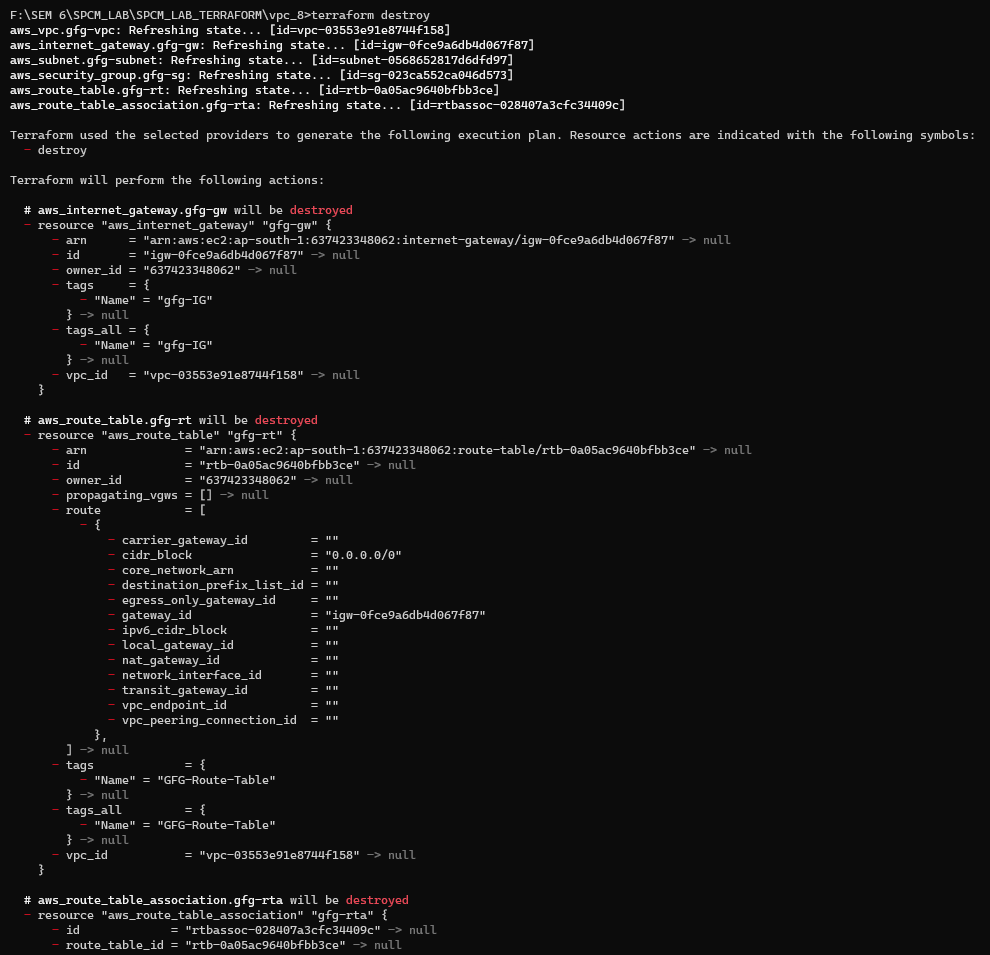
ROUTE TABLE:

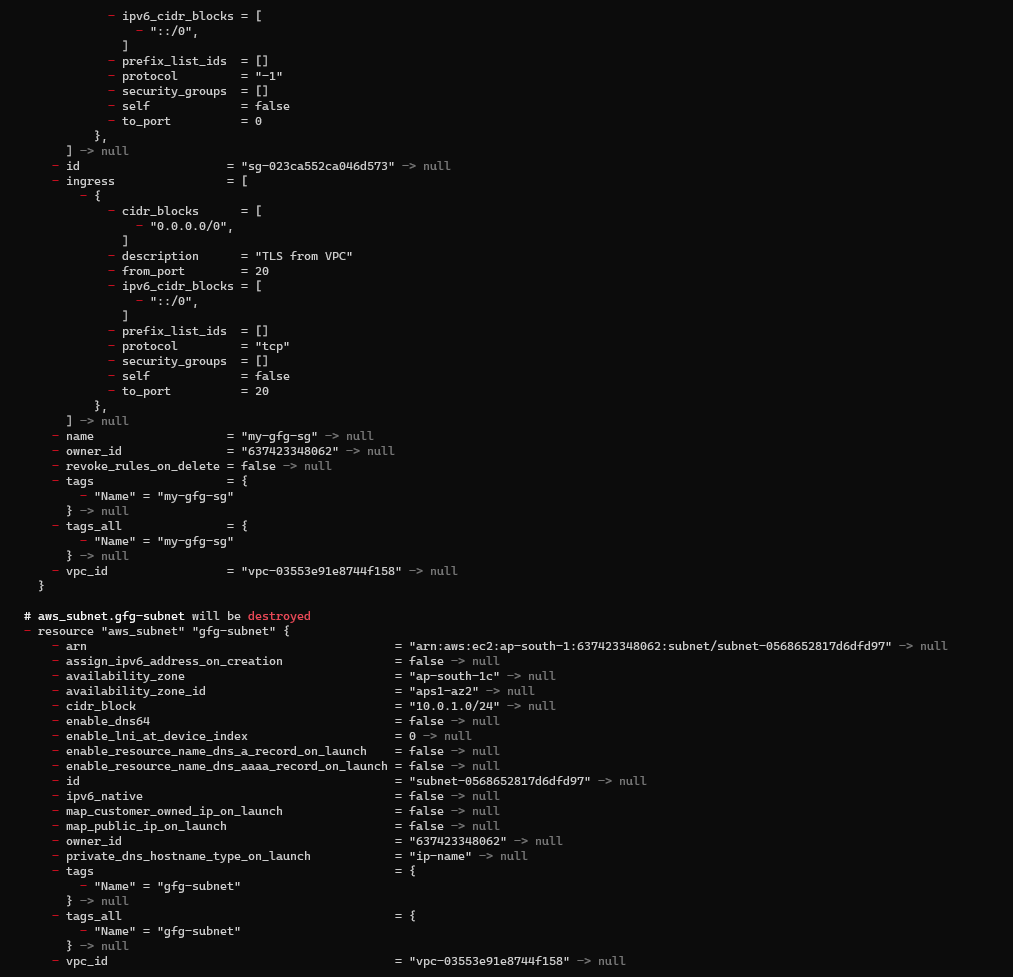


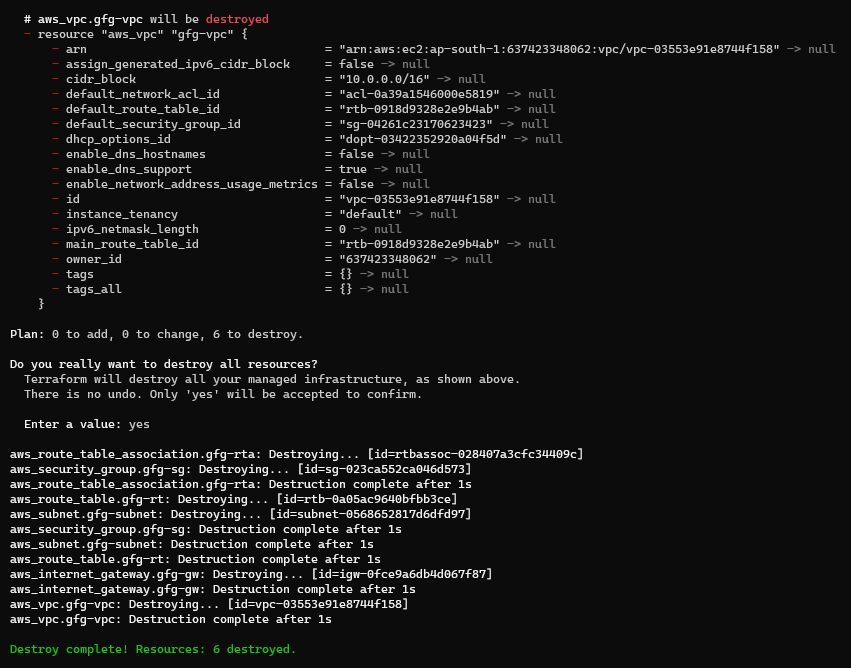
INTERNET GATEWAY



1. Run terraform destroy command.



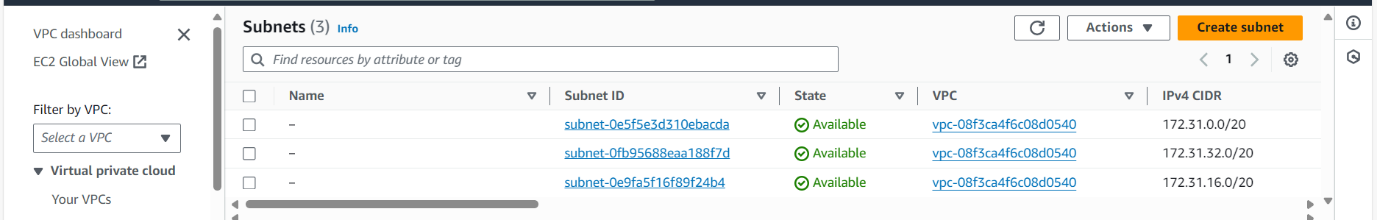




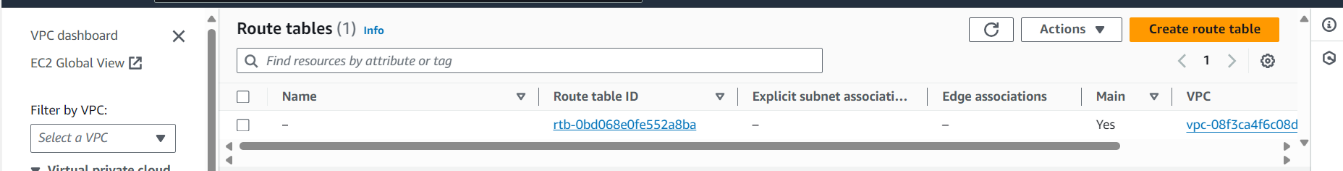
VPC:



SUBNET:



ROUTE TABLE:



INTERNET GATEWAY:

