# School of Computer Science

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES DEHRADUN, UTTARAKHAND**



System Provisioning and Configuration Management Lab

# Lab File (2023-2024)

**for**

**6th Semester**

**Submitted To:**

Dr. Hitesh Kumar Sharma

**Submitted By:**

Siddhanth verma

B. Tech. CSE [DevOps] 500094460

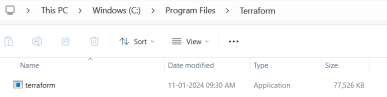
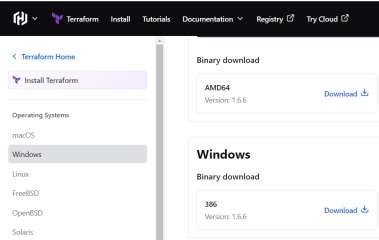
R2142210766

B2- Devops (Non-Hons)

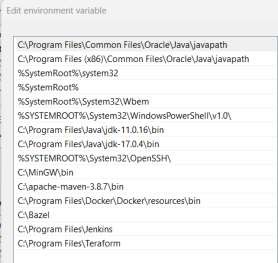
# EXPERIMENT – 1

## Aim: Install Terraform on Windows

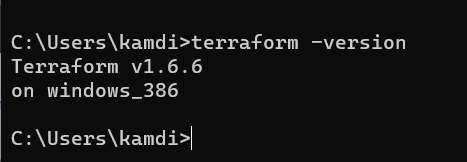
1. Download Terraform File for Windows



1. Add Terraform path to system Environment-variables



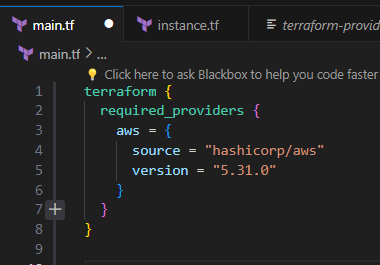
1. Verify Windows Terraform Insallation.
   1. Open a new command-prompt windows.
   2. Enter the command to check the terraform version: terraform -version



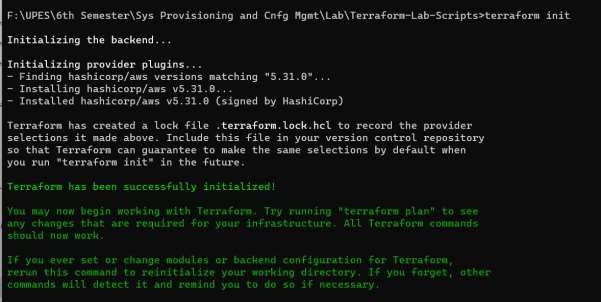
# EXPERIMENT – 2

## Aim: Terraform AWS provider and IAM user setting.

1. Create a new directory and Create terraform Configuration File (main.tf)



1. Initialize Terraform



# EXPERIMENT – 3

## Aim: Provisioning an EC2 Instance on AWS.

1. Create a Terraform Configuration File (main.tf)

A black rectangle with white text

Description automatically generatedA screenshot of a computer program

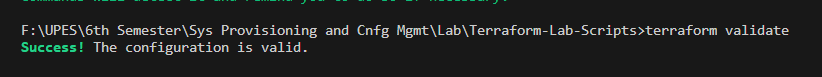
Description automatically generated

1. Initialize Terraform.

A screenshot of a computer

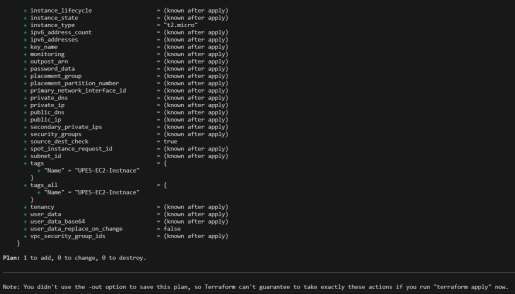
Description automatically generated

1. Validate the Script.

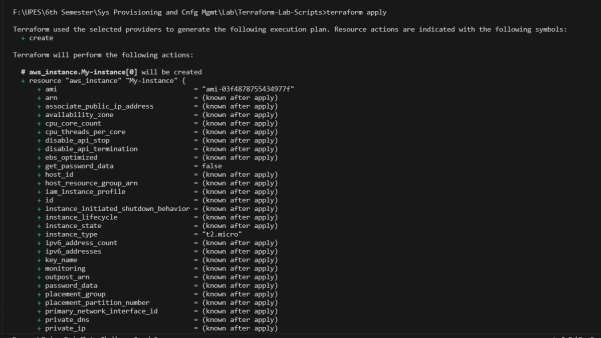


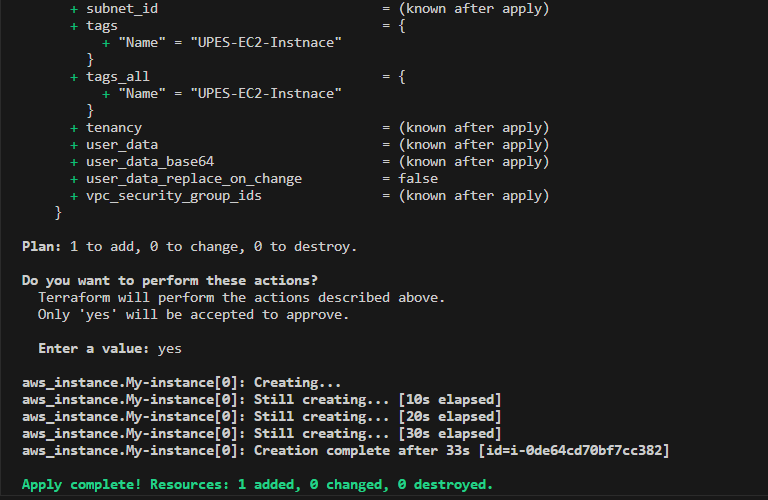
1. Create a Terraform Configuration File for EC2 instance (instance.tf).



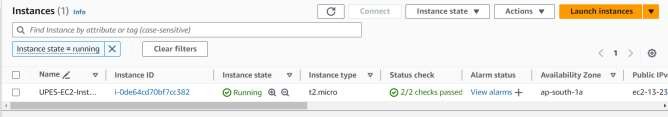
1. Review Plan using Command “Terraform plan”
2. A screenshot of a computer

   Description automatically generated
3. Apply it using command “Terraform apply”

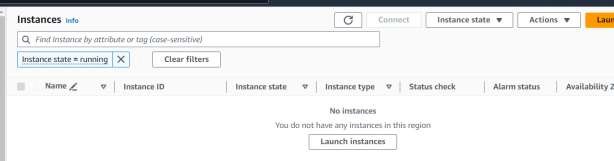
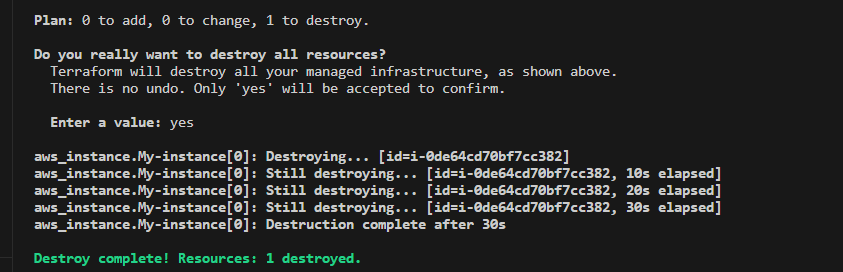




1. Verify Resources on AWS Management Console.



1. Cleanup Resources using command “Terraform destroy”



# EXPERIMENT – 4

## Aim: Terraform Variables.

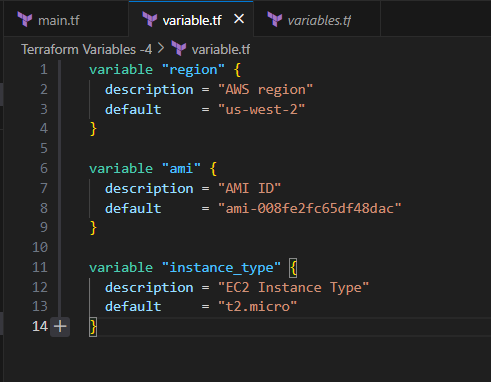
1. Create a Terraform Configuration File (main.tf)

A black rectangle with white text

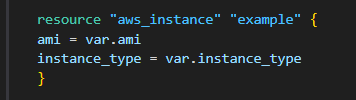
Description automatically generated A screenshot of a computer program

Description automatically generatedS

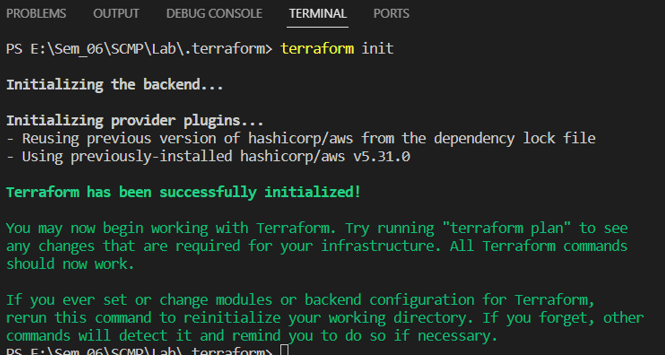
1. Create new file name as “variables.tf”’



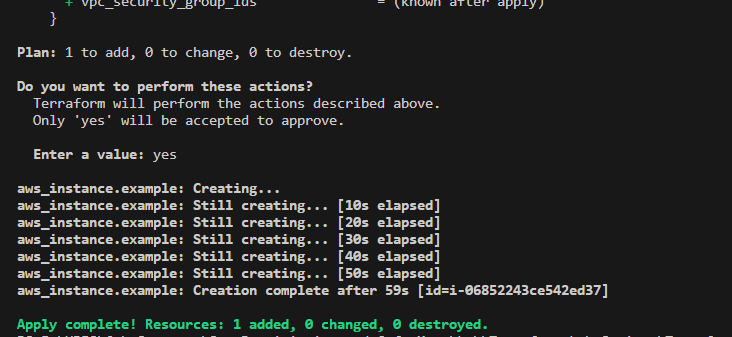
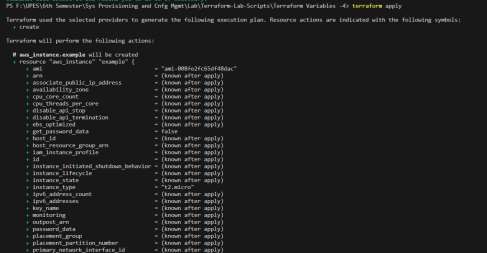
1. Use Variables in “main.tf” and update main.tf file.



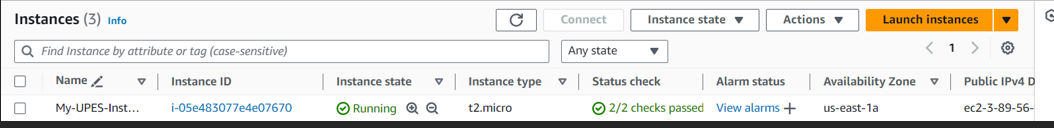
1. Initialize Terraform using command “terraform init”



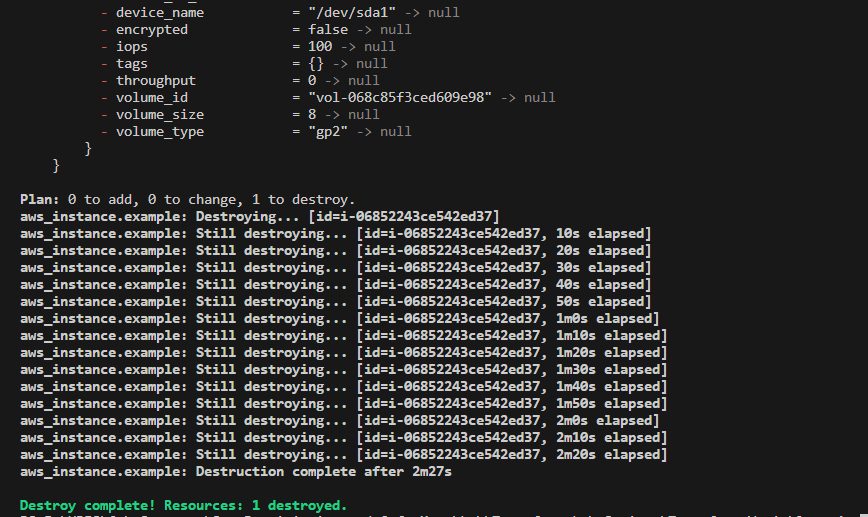
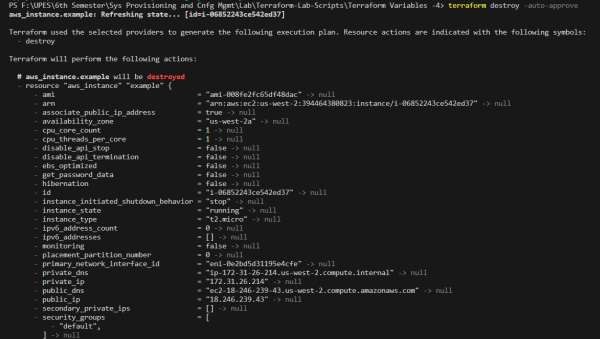
1. Apply it using command “Terraform apply”

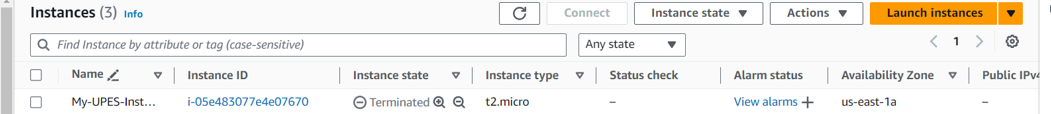


1. Verify Resources on AWS Management Console.



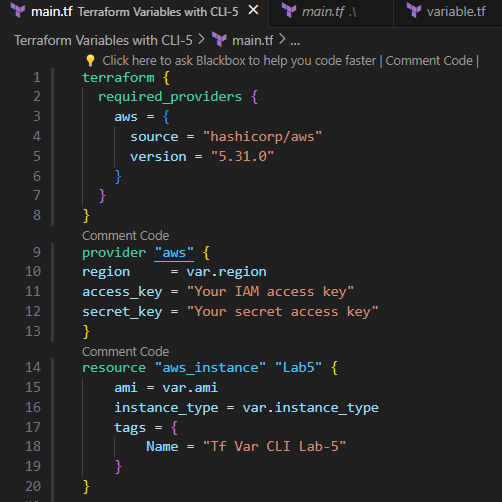
1. Cleanup Resources using command “Terraform destroy”





# EXPERIMENT – 5

## Aim: Terraform Variables with command Line Arguments

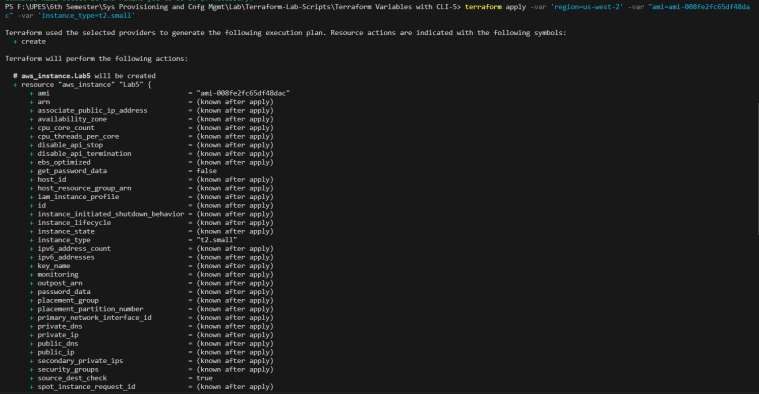
1. Create a Terraform Configuration File (main.tf)
2. 

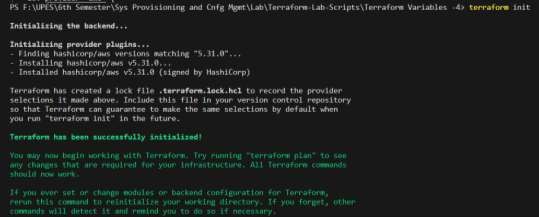
2.Create new file name as “variables.tf”’

A screenshot of a computer program

Description automatically generated

1. Initialize Terraform using command “terraform init”



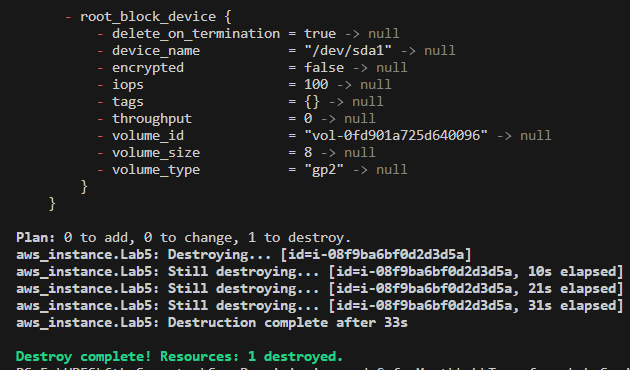


1. Apply command with command line arguments to set variables values using command “terraform apply -var 'region=us-west-2' -var "ami=ami-008fe2fc65df48dac" -var 'instance\_type=t2.small'”
2. Verify Resources on AWS Management Console.

A screenshot of a computer

Description automatically generated

1. Cleanup Resources using command “terraform destroy”



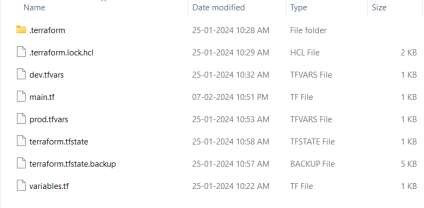
A screenshot of a computer

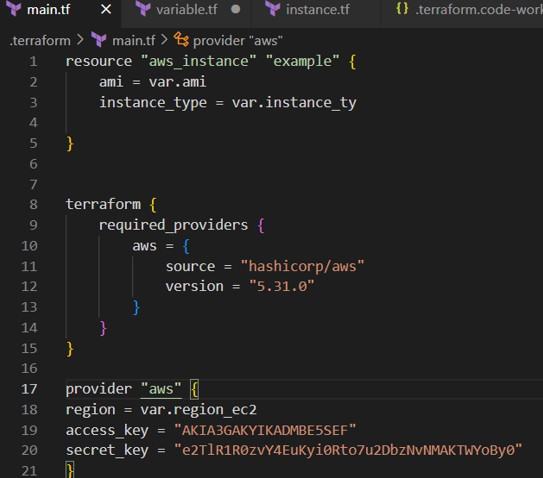
Description automatically generated

# EXPERIMENT – 6

## Aim: Terraform Multiple tfvars Files.

1. Create a new directory and Create terraform Configuration File (main.tf)





A black rectangle with white text

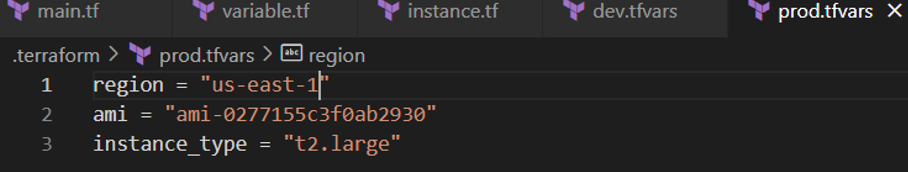
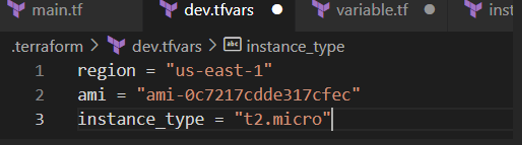
Description automatically generated

1. Create a file named as “variable.tf”

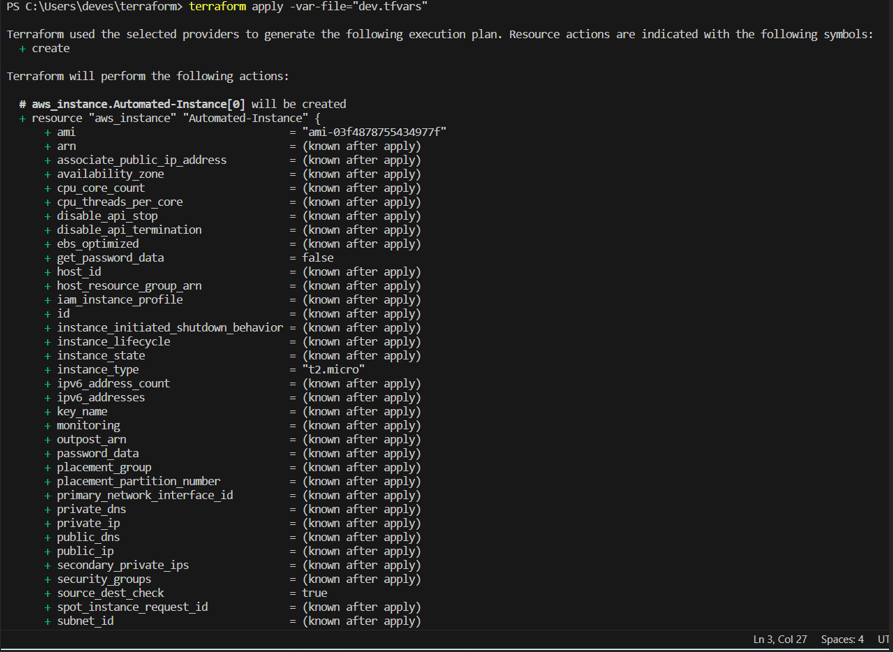
A screenshot of a computer program

Description automatically generated

1. Create Multiple tfvars Files:
   1. dev.tfvars
   2. prod.tfvars



1. Initialize Terraform for Dev Environment and apply it using command “Terraform apply”

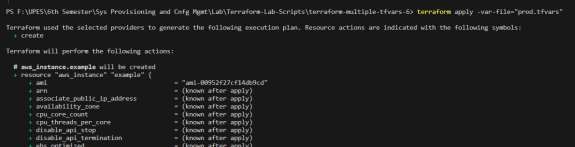


A screenshot of a computer

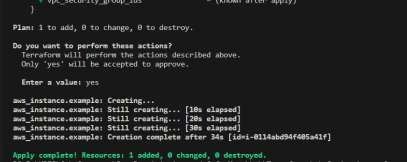
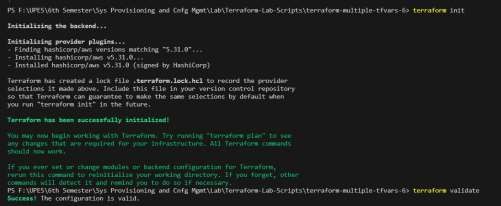
Description automatically generated

1. Verify Resources on AWS Management Console for Dev Environment





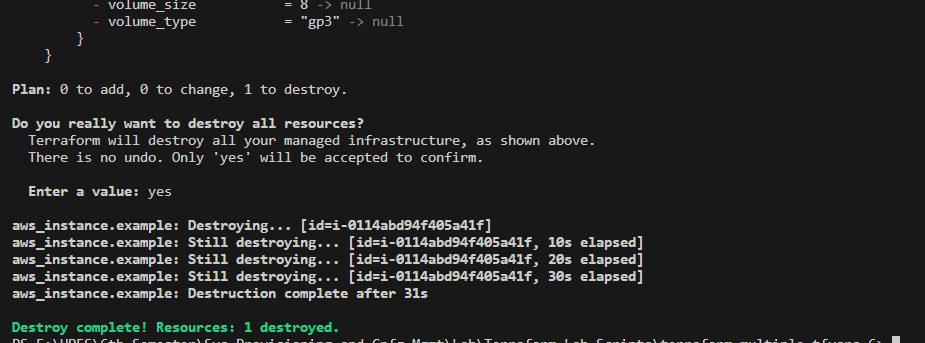
1. Initialize Terraform for Prod Environment and apply it using command “Terraform apply”



1. Verify Resources on AWS Management Console for Prod Environment



1. Cleanup Resources for Dev and Prod Environment using command “Terraform destroy”



# EXPERIMENT – 7

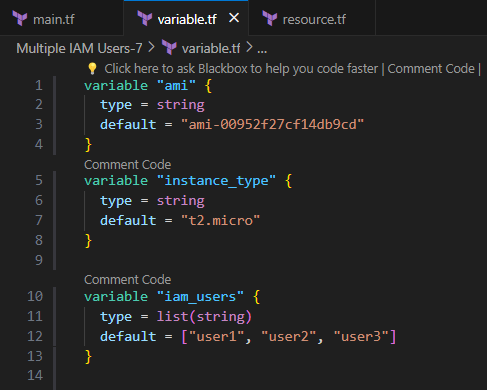
## Aim: Creating Multiple IAM Users in Terraform.

1. Create a Terraform Configuration File (main.tf)

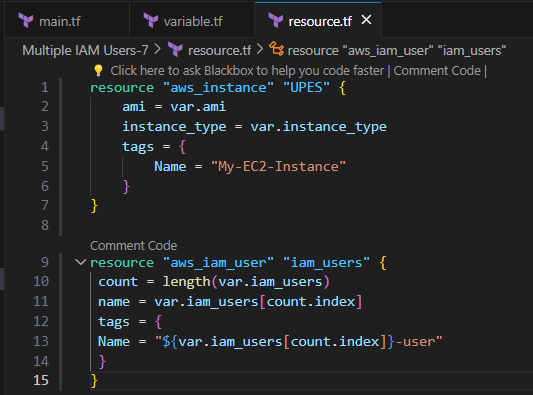
A screenshot of a computer screen

Description automatically generated

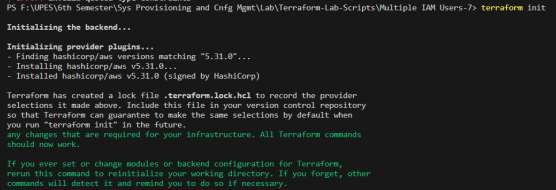
1. Create new file name as “variables.tf”’



1. Create new file name as “resource.tf”’and define a list variable IAM users containing the names of the IAM users that we want to create.

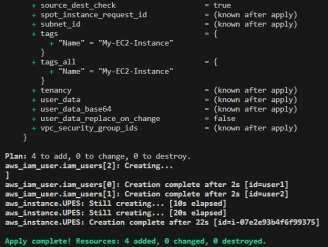


1. Initialize Terraform using command “terraform init”



1. Apply it using command “Terraform apply”





1. Verify Resources on AWS Management Console.

**EC2**



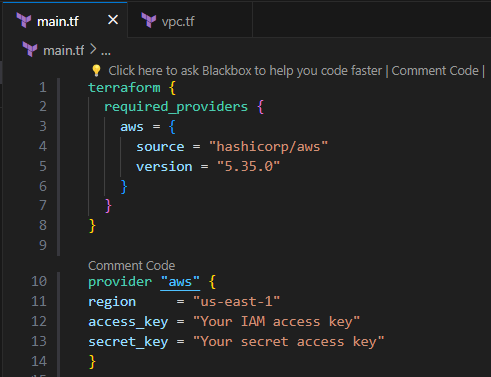
**User**

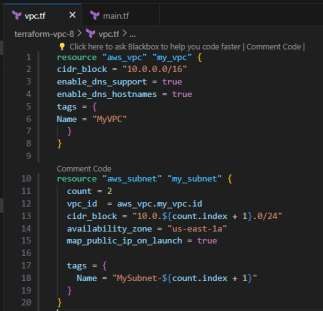


# EXPERIMENT – 8

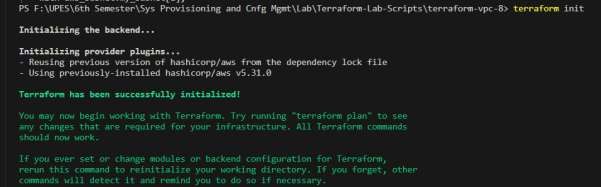
## Aim: Creating a VPC in Terraform

1. A black rectangle with white text

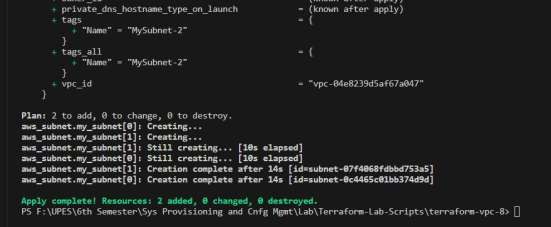
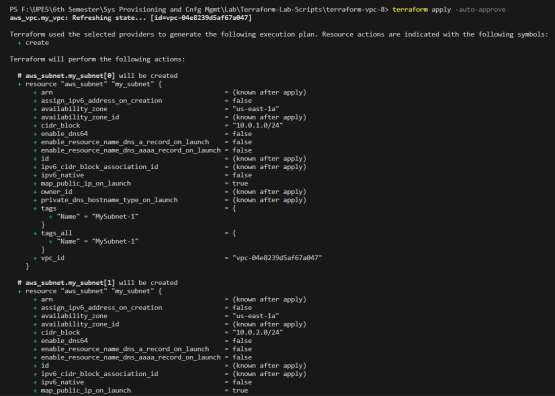
   Description automatically generatedCreate a Terraform Configuration File (main.tf)
2. 
3. Create new file name as “vpc.tf”’



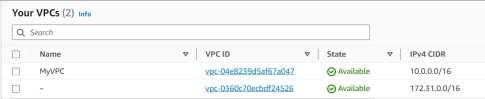
1. Initialize Terraform using command “terraform init”



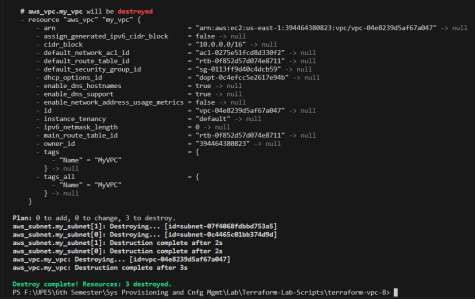
1. Apply it using command “Terraform apply -auto-approve”



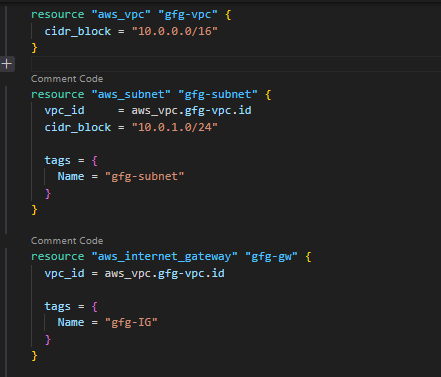
1. Verify Resources on AWS Management Console.

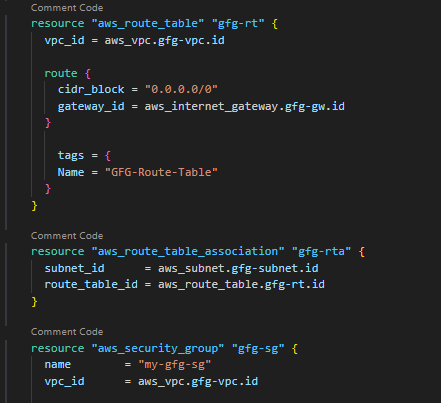


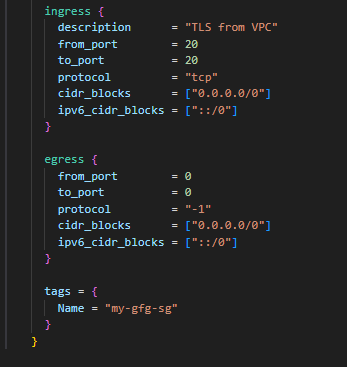
1. Cleanup Resources using command “Terraform destroy”



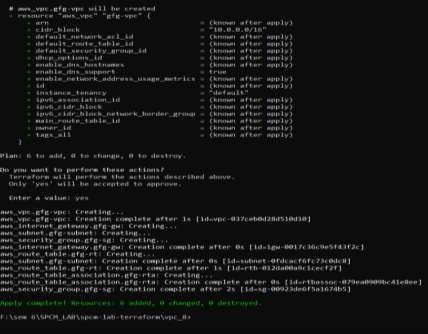
1. Now Update the vpc.tf file and repeat the previous Steps.



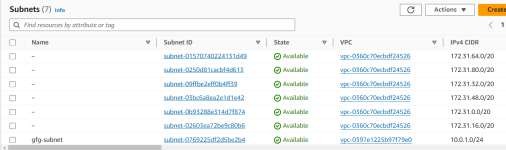




1. Apply it using command “Terraform apply”



1. Again verify resources on AWS Management Console.



1. Cleanup Resources using command “Terraform destroy”



# EXPERIMENT – 9

## Aim: Creating Multiple EC2 Instances with for each in Terraform.

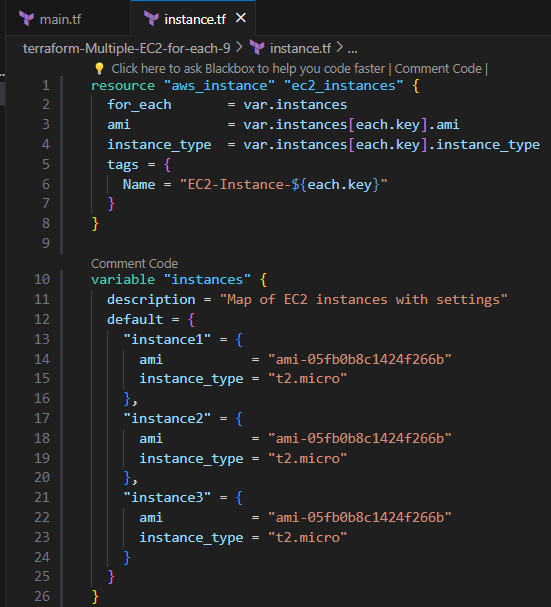
1. Create a Terraform Configuration File (main.tf)

A black rectangle with white text

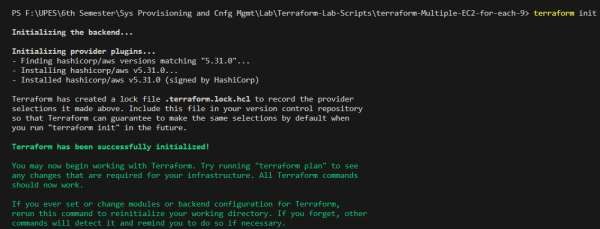
Description automatically generated A screen shot of a computer

Description automatically generated

1. Create new file name as “instance.tf”



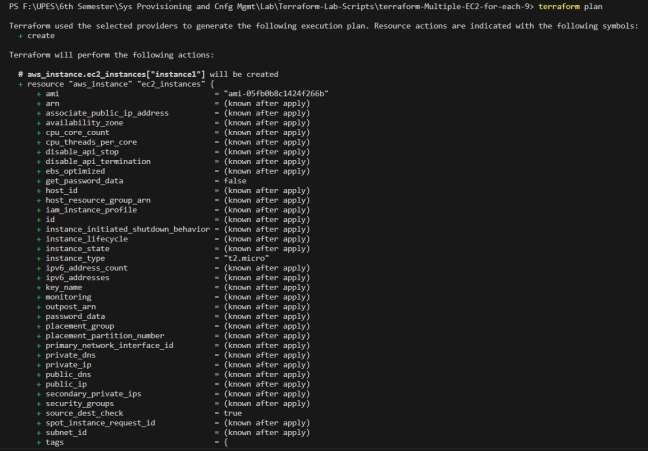
1. Initialize Terraform using command “terraform init”

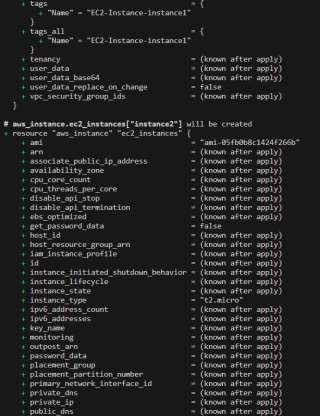


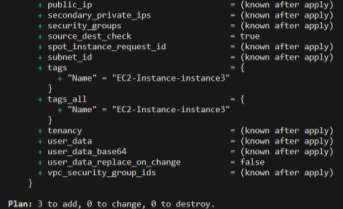
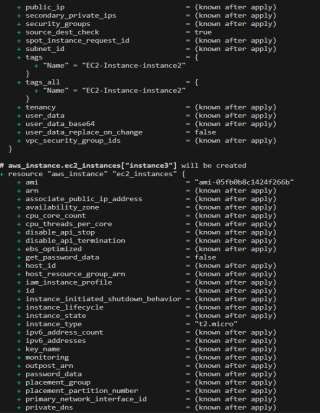
1. Validate it using command “terraform validate”



1. Check the Plan using command “terraform plan”







# EXPERIMENT – 10

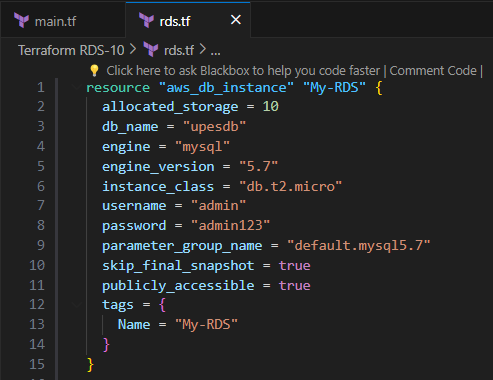
## Aim: Creating an AWS RDS Instance in Terraform.

1. Create a Terraform Configuration File (main.tf)

A screen shot of a computer

Description automatically generated

1. Create a Terraform RDS File (rds.tf)



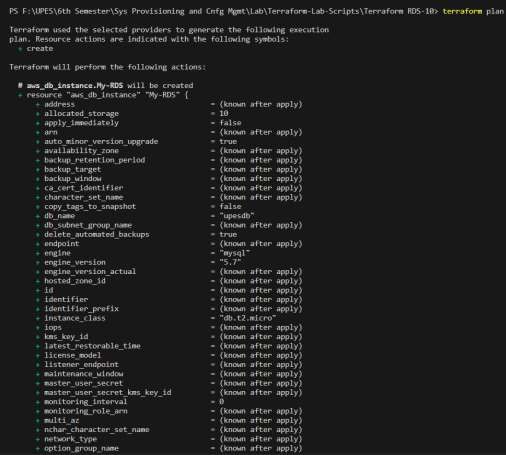
1. Initialize Terraform using command “terraform init”

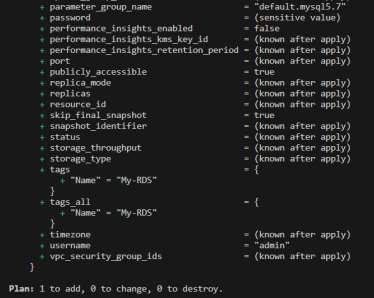


1. Validate it using command “terraform validate”

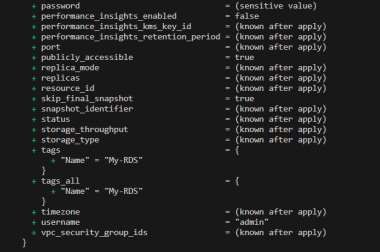
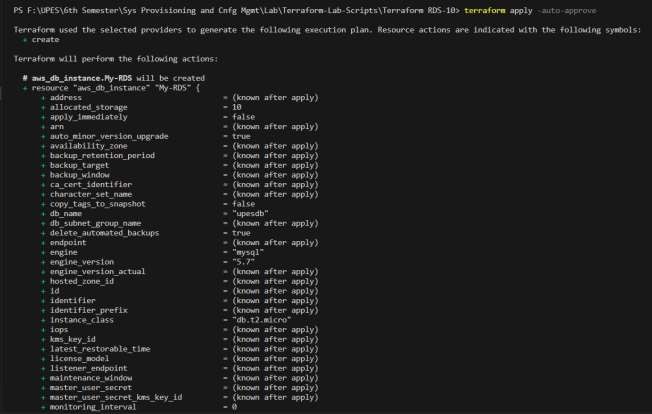


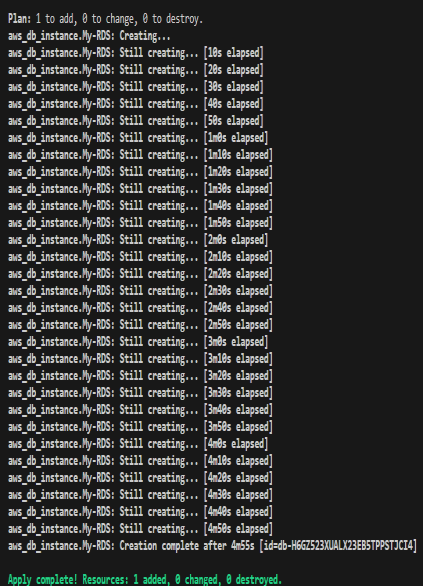
1. Check the Plan using command “terraform plan”

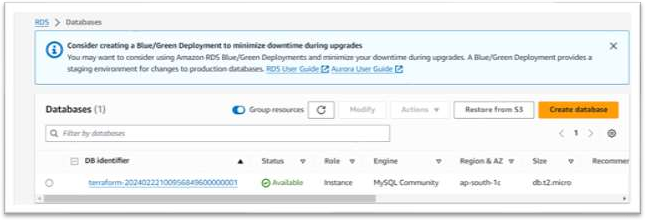




1. Apply it using command “Terraform apply -auto-approve”



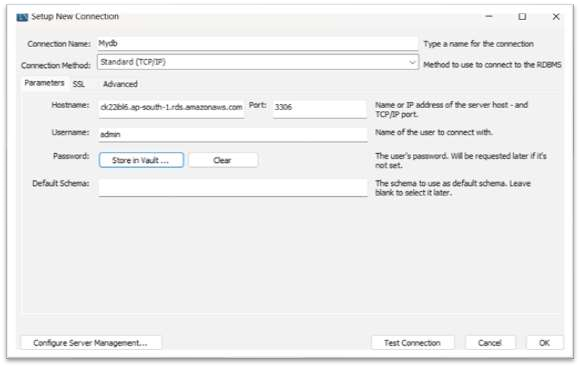


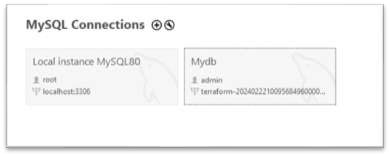


Connect with MySQL Workbench with proper Configuration and save it.

A screenshot of a computer

Description automatically generated





1. Cleanup Resources using command “Terraform destroy”

