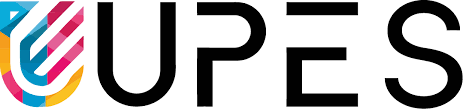
**School of Computer Science**

**UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**

**DEHRADUN, UTTARAKHAND**



**System Provisioning and**

**Configuration Management**

|  |  |
| --- | --- |
| Submitted To:  Dr. Hitesh Kumar Sharma | Submitted By:  Mandala Praneeth reddy  Batch – 2(DevOps)  500094008 |

*Lab Exercise 2– Terraform AWS Provider and IAM User Setting*

***Prerequisites: Terraform Installed: Make sure you have Terraform installed on your machine. Follow the official installation guide if needed.***

***AWS Credentials: Ensure you have AWS credentials (Access Key ID and Secret Access Key) configured. You can set them up using the AWS CLI or by setting environment variables.***

***Exercise Steps:***

***Step 1: Create a New Directory:***

***Create a new directory for your Terraform configuration:***

***A screenshot of a computer

Description automatically generated***

***Step 2: Create Terraform Configuration File (main.tf):***

***Create a file named main.tf with the following content:***

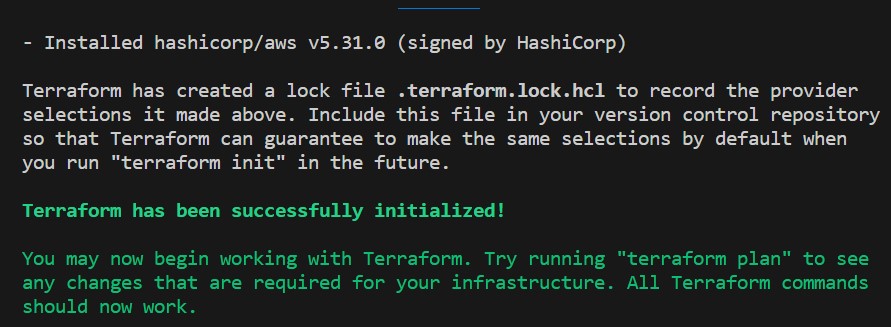
***A screenshot of a computer program

Description automatically generated***

***This script defines an AWS provider and provisions an EC2 instance.***

***Step 3: Initialize Terraform:***

***Run the following command to initialize your Terraform working directory:***

******

*Lab Exercise 3–Provisioning an EC2 Instance on AWS*

# *Prerequisites: Terraform Installed: Make sure you have Terraform installed on your machine. Follow the official installation guide if needed.*

***AWS Credentials: Ensure you have AWS credentials (Access Key ID and Secret Access Key) configured. You can set them up using the AWS CLI or by setting environment variables.***

# *Exercise Steps:*

***Step 1: Create a New Directory:***

***Create a new directory for your Terraform configuration:***

******

# *Step 2: Create Terraform Configuration File (main.tf):*

***Create a file named main.tf with the following content:***

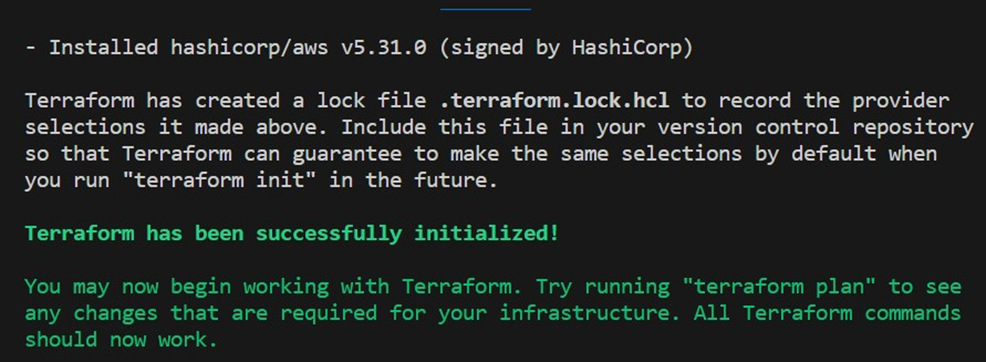
***A screenshot of a computer program

Description automatically generated***

***This script defines an AWS provider and provisions an EC2 instance.***

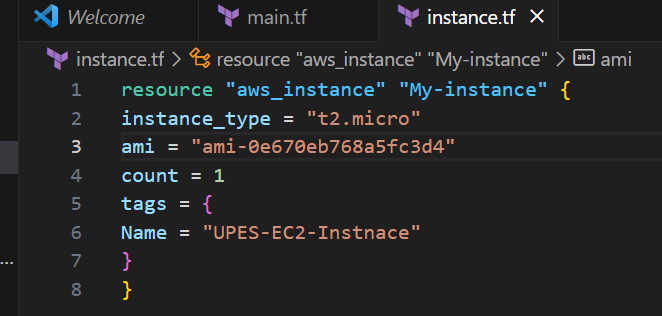
# *Step 3: Initialize Terraform:*

***Run the following command to initialize your Terraform working directory:***

******

# *Step 4: Create Terraform Configuration File for EC2 instance (instance.tf):*

***Create a file named instnace.tf with the following content:***

******

# *Step 5: Review Plan:*

***Run the following command to see what Terraform will do:***

***Review the plan to ensure it aligns with your expectations.***

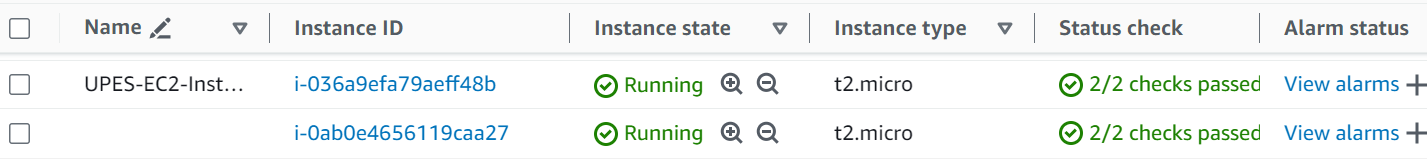
***Apply the changes to create the AWS resources:***

***A screenshot of a computer program

Description automatically generated***

***A screen shot of a computer

Description automatically generated***

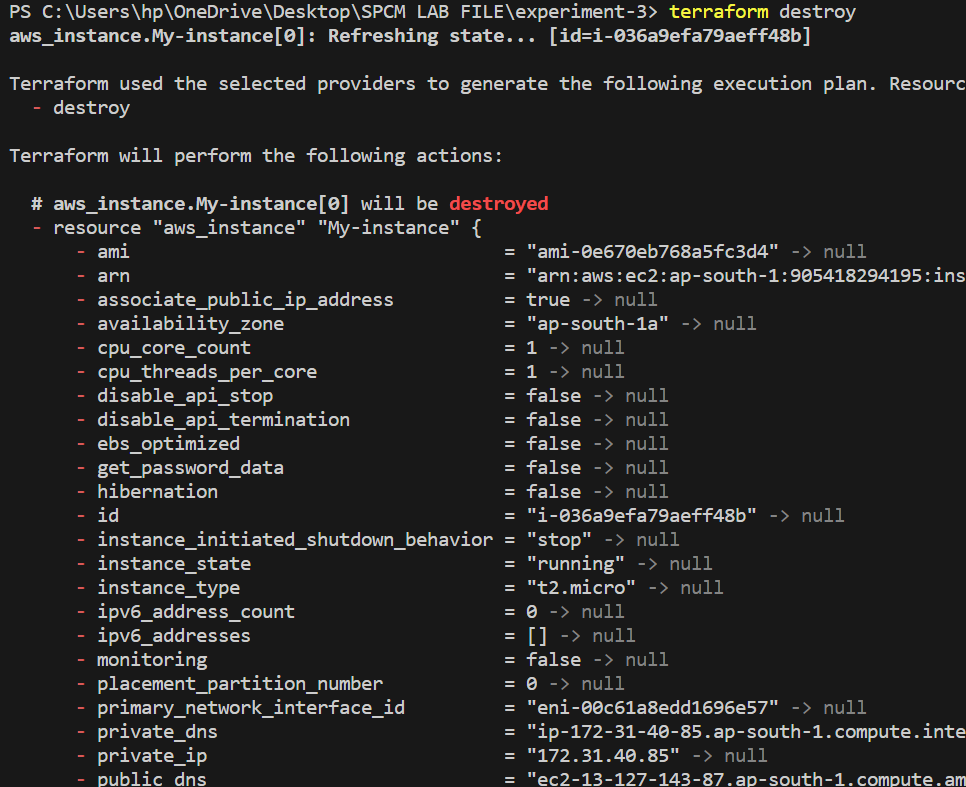
******

# *Step 7: Verify Resources:*

***After the terraform apply command completes, log in to your AWS Management Console and navigate to the EC2 dashboard. Verify that the EC2 instance has been created.***

# *Step 8: Cleanup Resources:*

***When you are done experimenting, run the following command to destroy the created resources:***

******

***A computer screen shot of a program

Description automatically generated***

******

***Lab Exercise 4– Terraform Variables Objective:***

***Learn how to define and use variables in Terraform configuration.***

***Prerequisites:***

***• Install Terraform on your machine.***

***Steps:***

1. ***Create a Terraform Directory:*** 
   * ***Create a new directory for your Terraform project.***

***A screenshot of a computer

Description automatically generated***

1. ***Create a Terraform Configuration File:*** 
   * ***Create a file named main.tf within your project directory.***

***A screen shot of a computer program

Description automatically generatedA black screen with orange letters

Description automatically generated***

1. ***Define Variables:*** 
   * ***Open a new file named variables.tf. Define variables for region, ami, and instance\_type.***

***# variables.tf***

***A computer screen with text and images

Description automatically generated***

***Use Variables in main.tf:***

* + ***Modify main.tf to use the variables. # main.tf***

***A screen shot of a computer program

Description automatically generatedA black screen with orange letters

Description automatically generated***

1. ***Initialize,Apply&Validate:*** 
   * ***Run the following Terraform commands to initialize and apply the configuration.***

***A screenshot of a computer program

Description automatically generated***

***A screenshot of a computer

Description automatically generatedA computer screen shot of a program

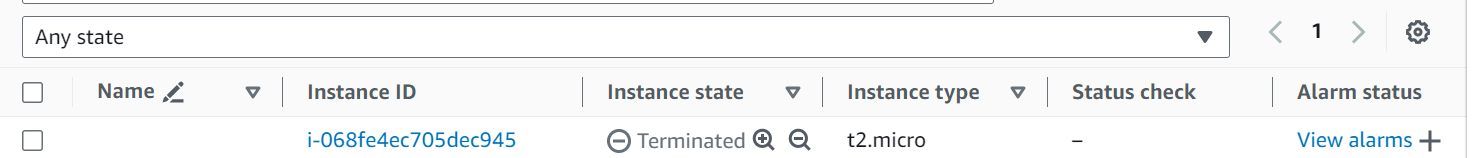
Description automatically generated***

1. ***Clean Up:***

***A screenshot of a computer program

Description automatically generatedA computer screen with white text and numbers

Description automatically generated***

******

***Lab Exercise 5– Terraform Variables with Command Line Arguments***

***Use Command Line Arguments:***

***• Open a terminal and navigate to your Terraform project directory.***

***• Run the terraform init command:***

***Run the terraform apply command with command line arguments to set variable values:***

***A screen shot of a computer code

Description automatically generated A screenshot of a computer program

Description automatically generated***

***A screen shot of a computer program

Description automatically generated***

***A screenshot of a computer

Description automatically generated***

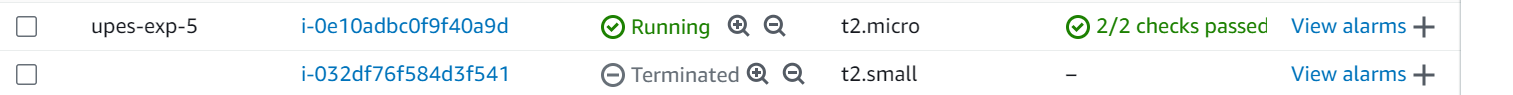
***Clean Up: After testing, you can clean up resources:***

***A screenshot of a computer program

Description automatically generated***

***A screenshot of a computer

Description automatically generated***

******

***Lab Exercise 6– Terraform Multiple tfvars Files***

***Objective:***

***Learn how to use multiple tfvars files in Terraform for different***

***environments.***

***Prerequisites:***

***• Terraform installed on your machine.***

***• Basic knowledge of Terraform configuration and variables.***

***Steps:***

***1. Create a Terraform Directory:***

***• Create Terraform Configuration Files:***

***A screenshot of a computer

Description automatically generated***

***• Create a file named main.tf:***

***# main.tf***

***A computer screen with text and numbers

Description automatically generatedA computer screen with text and numbers

Description automatically generated***

***• Create a file named variables.tf:***

***# variables.tf***

***A screen shot of a computer program

Description automatically generated***

***2. Create Multiple tfvars Files:***

***•Create a file named dev.tfvars:***

***# dev.tfvars***

***A computer screen with text and numbers

Description automatically generated***

***•Create a file named prod.tfvars:***

***# prod.tfvars***

***A computer code with text and numbers

Description automatically generated with medium confidence***

***•In these files, provide values for the variables based on the environments.***

***3. Initialize and Apply for Dev Environment:***

***•Run the following Terraform commands to initialize and apply the***

***configuration***

***for the dev environment:***

***A screenshot of a computer program

Description automatically generated***

***A computer screen shot of white text

Description automatically generated***

***4. Initialize and Apply for Prod Environment:***

***•Run the following Terraform commands to initialize and apply the***

***configuration***

***for the prod environment:***

***A screenshot of a computer

Description automatically generated***

***A screen shot of a computer

Description automatically generated***

***5. Test and Verify:***

***•Observe how different tfvars files are used to set variable values for different***

***environments during the apply process.***

***•Access the AWS Management Console or use the AWS CLI to verify the***

***creation of***

***resources in the specified regions and instance types.***

***A screenshot of a computer

Description automatically generated***

***6. Clean Up:***

***•After testing, you can clean up resources:***

***terraform destroy -var-file=dev.tfvars***

***A computer screen shot of a program

Description automatically generated***

***A computer screen with white text

Description automatically generated***

***terraform destroy -var-file=prod.tfvars***

***•Confirm the destruction by typing yes.***

***A computer screen shot of a black screen

Description automatically generated***

***A computer screen shot of a computer program

Description automatically generated***

***Lab Exercise 7– Creating Multiple IAM Users***

***in Terraform***

***Objective:***

***Learn how to use Terraform to create multiple IAM users with unique settings.***

***Prerequisites:***

***• Terraform installed on your machine.***

***• AWS CLI configured with the necessary credentials.***

***Steps:***

1. ***Create a Terraform Directory:***

***A screenshot of a computer

Description automatically generated***

***# main.tf***

***A computer screen shot of a program code

Description automatically generated***

***#iamuser.tf***

***A screen shot of a computer program

Description automatically generated***

***2.Initialize and Apply: - Run the following Terraform commands to initialize and apply the configuration: terraform init***

***A screenshot of a computer program

Description automatically generated***

***terraform apply: -***

***A screenshot of a computer program

Description automatically generated***

***A screenshot of a computer program

Description automatically generated***

***Terraform will prompt you to confirm the creation of IAM users. Type yes and press Enter.***

1. ***Verify Users in AWS Console:***

***• Log in to the AWS Management Console and navigate to the IAM service.***

***• Verify that the IAM users with the specified names and tags have been created.***

***A screenshot of a computer

Description automatically generated***

1. ***Update IAM Users:***

***• If you want to add or remove IAM users, modify the iam\_users list in the main.tf file.***

***• Rerun the terraform apply command to apply the changes: terraform apply***

1. ***Clean Up:***

***• After testing, you can clean up the IAM users: terraform destroy***

***• Confirm the destruction by typing yes.***

***A screenshot of a computer program

Description automatically generated***

***A screenshot of a computer program

Description automatically generated***

***Lab Exercise 8– Creating a VPC in Terraform***

***Objective: -***

***Learn how to use Terraform to create a basic Virtual Private Cloud (VPC) in AWS.***

***Prerequisites: -***

***• Terraform installed on your machine.***

***• AWS CLI configured with the necessary credentials.***

***Steps: 1. Create a Terraform Directory: -***

******

***# main.tf***

***A screen shot of a computer program

Description automatically generated***

***In this configuration, we define an AWS provider, a VPC with a specified CIDR block,***

***and two subnets within the VPC.***

***2. Initialize and Apply: -***

***• Run the following Terraform commands to initialize and apply the configuration: -***

***terraform init***

***A screenshot of a computer

Description automatically generated***

***terraform apply***

***• Terraform will prompt you to confirm the creation of the VPC and subnets. Type yes and press Enter.***

***A computer screen shot of a program

Description automatically generated***

***A computer screen with white text

Description automatically generated***

***A screenshot of a computer program

Description automatically generated***

***3. Verify Resources in AWS Console: -***

***• Log in to the AWS Management Console and navigate to the VPC service.***

***• Verify that the VPC and subnets with the specified names and settings have been***

***A screenshot of a computer

Description automatically generated***

***4. Update VPC Configuration:***

***• If you want to modify the VPC configuration, update the main.tf file with the desired changes.***

***• Rerun the terraform apply command to apply the changes: terraform apply***

***5. Clean Up: After testing, you can clean up the VPC and subnets:***

***terraform destroy***

***A computer screen shot of a program

Description automatically generated***

***A screenshot of a computer screen

Description automatically generated***

***6. Conclusion:***

***This lab exercise demonstrates how to create a basic Virtual Private Cloud (VPC) with subnets in AWS using Terraform. The example includes a simple VPC configuration with two subnets.***

***Experiment with different CIDR blocks, settings, and additional AWS resources to customize your VPC.***

***Lab Exercise 9– Creating Multiple EC2 Instances with for each in Terraform***

***Objective:***

***Learn how to use for each in Terraform to create multiple AWS EC2 instances with specific settings for each instance.***

***Prerequisites: -***

***• Terraform installed on your machine.***

***• AWS CLI configured with the necessary credentials.***

***Steps: -***

1. ***Create a Terraform Directory:***

******

***A screenshot of a computer

Description automatically generated***

***• Create Terraform Configuration Files: -***

***• Create a file named main.tf:***

***# main.tf***

******

***• Replace "your-key-pair-name" and "your-subnet-id" with your actual key pair name and subnet ID. • In this configuration, we define a variable instance as a map containing settings for each EC2 instance. The aws instance resource is then used with for each to create instances based on the map.***

1. ***Initialize and Apply: -***

***• Run the following Terraform commands to initialize and apply the configuration: -***

***A screenshot of a computer screen

Description automatically generated***

***#Terraform apply***

***A screenshot of a computer

Description automatically generated***

***A computer screen shot of a program

Description automatically generated***

***Terraform will prompt you to confirm the creation of EC2 instances. Type yes and press Enter.***

1. ***Verify Instances in AWS Console:***

***• Log in to the AWS Management Console and navigate to the EC2 service.***

***• Verify that the specified EC2 instances with the specified names and settings have been created.***

***A screenshot of a computer

Description automatically generated***

***. 4.***

***Clean Up: -***

***• After testing, you can clean up the EC2 instances:***

***A computer screen shot of a black screen

Description automatically generated***

***A screenshot of a computer program

Description automatically generated***

***Lab Exercise 10– Creating an AWS RDS Instance in Terraform***

***Objective:***

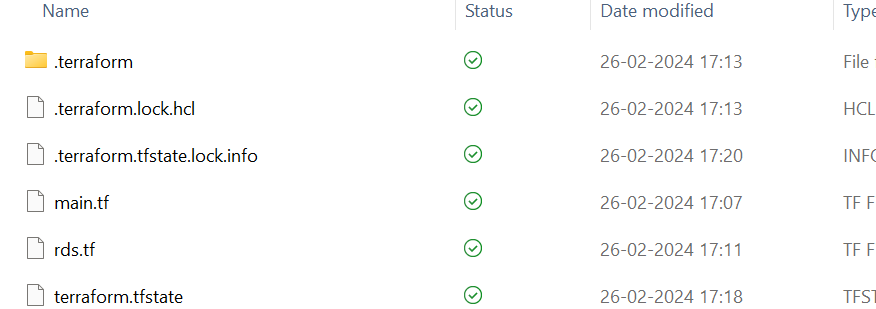
***Learn how to use Terraform to create an AWS RDS instance.***

***Prerequisites:***

***• Terraform installed on your machine.***

***• AWS CLI configured with the necessary credentials.***

***Steps:***

***1. Create a Terraform Directory: ***

***2. Create Terraform Configuration Files:***

***Create a file named main.tf:***

***# main.tf***

***A screen shot of a computer program

Description automatically generated***

***#rds.tf***

***A computer screen shot of text

Description automatically generated***

***3. Initialize and Apply:***

***• Run the following Terraform commands to initialize and apply the configuration:***

***terraform init***

***A screenshot of a computer program

Description automatically generated***

***terraform apply***

***• Terraform will prompt you to confirm the creation of the RDS instance. Type yes***

***and press Enter.***

***A computer screen with white text

Description automatically generated***

***A computer screen shot of a program

Description automatically generatedA screen shot of a computer program

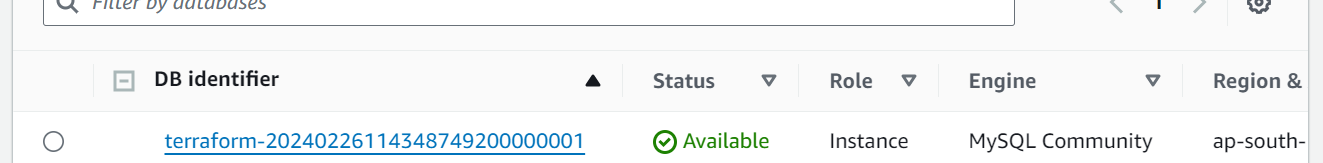
Description automatically generated***

***4. Verify RDS Instance in AWS Console:***

***• Log in to the AWS Management Console and navigate to the RDS service.***

***• Verify that the specified RDS instance with the specified settings has been***

***created.***

******

***6. Clean Up:***

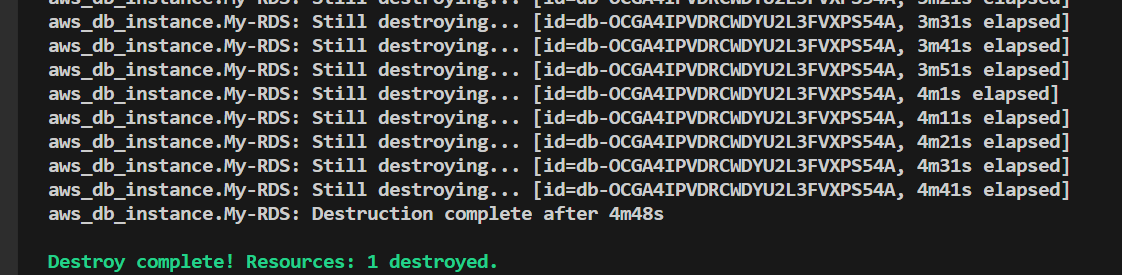
***After testing, you can clean up the RDS instance:***

***terraform destroy***

***Confirm the destruction by typing yes.***

***A screen shot of a computer

Description automatically generated***

******

***7. Conclusion:***

***This lab exercise demonstrates how to use Terraform to create an AWS RDS***

***instance.***

***You learned how to define RDS settings, initialize and apply the Terraform***

***configuration, and verify the creation of the RDS instance in the AWS***

***Management***

***Console. Experiment with different RDS settings in the main.tf file to observe how***