

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:

```
mkdir aws-terraform-demo
```

```
cd aws-terraform-demo
```

```
palakgupta@Palaks-MacBook-Air ~ % cd Documents
palakgupta@Palaks-MacBook-Air Documents % ls
Vaibhav_Resume.pdf      java          photo.jpg          sem 6          wallpaper
dsa.pages               palak        resume.pdf         sem 5          vpn.db
palakgupta@Palaks-MacBook-Air Documents % cd sem 6
cd: string not in pwd: sem
palakgupta@Palaks-MacBook-Air Documents % cd sem6
palakgupta@Palaks-MacBook-Air sem6 % ls
SPCM          ci:cd        practice
palakgupta@Palaks-MacBook-Air sem6 % cd SPCM
palakgupta@Palaks-MacBook-Air SPCM % ls
Lab Exercise 1- Install Terraform on Windows.pages      Terraform
Lab Exercise 2- Terraform AWS Provider and IAM user Settings.docx
palakgupta@Palaks-MacBook-Air SPCM % cd Terraform
palakgupta@Palaks-MacBook-Air Terraform % ls
main.tf
palakgupta@Palaks-MacBook-Air Terraform %
```

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

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

```
terraform {

  required_providers {

    aws = {
```

```
source = "hashicorp/aws"

version = "5.31.0"

}

}

}
```

```
provider "aws" {

region    = "ap-south-1"

access_key = "your IAM access key"

secret_key = "your secret access key"

}
```

```
main.tf > provider "aws"
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.83.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "us-east-1"
12   access_key = "AKIAQMEY6IA6VI75R3U4"
13   secret_key = "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"
14 }
```

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:

```
terraform init
```

```
palakgupta@Palaks-MacBook-Air Terraform % terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.83.0"...
- Installing hashicorp/aws v5.83.0...
- Installed hashicorp/aws v5.83.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

```
palakgupta@Palaks-MacBook-Air Terraform %
```