ANSHIKA SRIVASTAVA

ROLL NUMBER – R2142220907

SAP ID – 500107049

LAB EXERCISE 2

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.

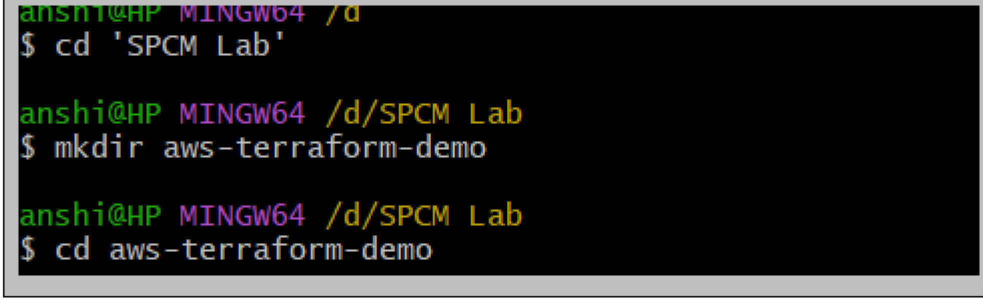
AWS Credentials: Ensure you have AWS credentials (Access Key ID and Secret Access Key) configured. You can set them up using the AWS CLI or by setting environment variables.

Exercise Steps:

Step 1: Create a New Directory:

Create a new directory for your Terraform configuration:

```
mkdir aws-terraform-demo

cd aws-terraform-demo
```



```
anshi@HP MINGW64 /d
$ cd 'SPCM Lab'

anshi@HP MINGW64 /d/SPCM Lab
$ mkdir aws-terraform-demo

anshi@HP MINGW64 /d/SPCM Lab
$ cd aws-terraform-demo
```

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

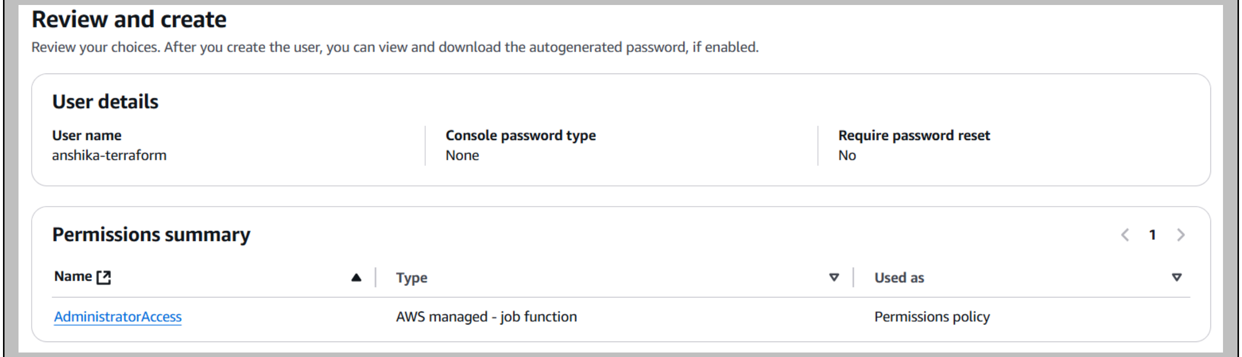
Create a file named main.tf with the following content:

```
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"
}
```

Creating IAM user first and giving 'administrator access'



Review and create

Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

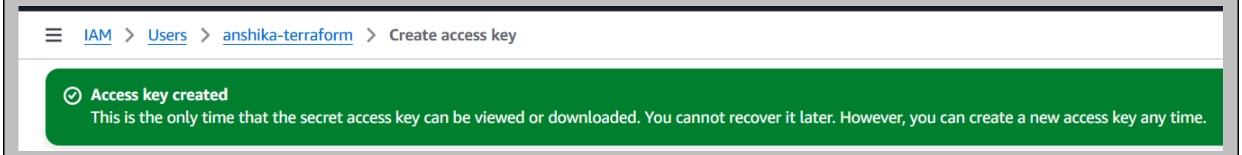
User details

User name	Console password type	Require password reset
anshika-terraform	None	No

Permissions summary

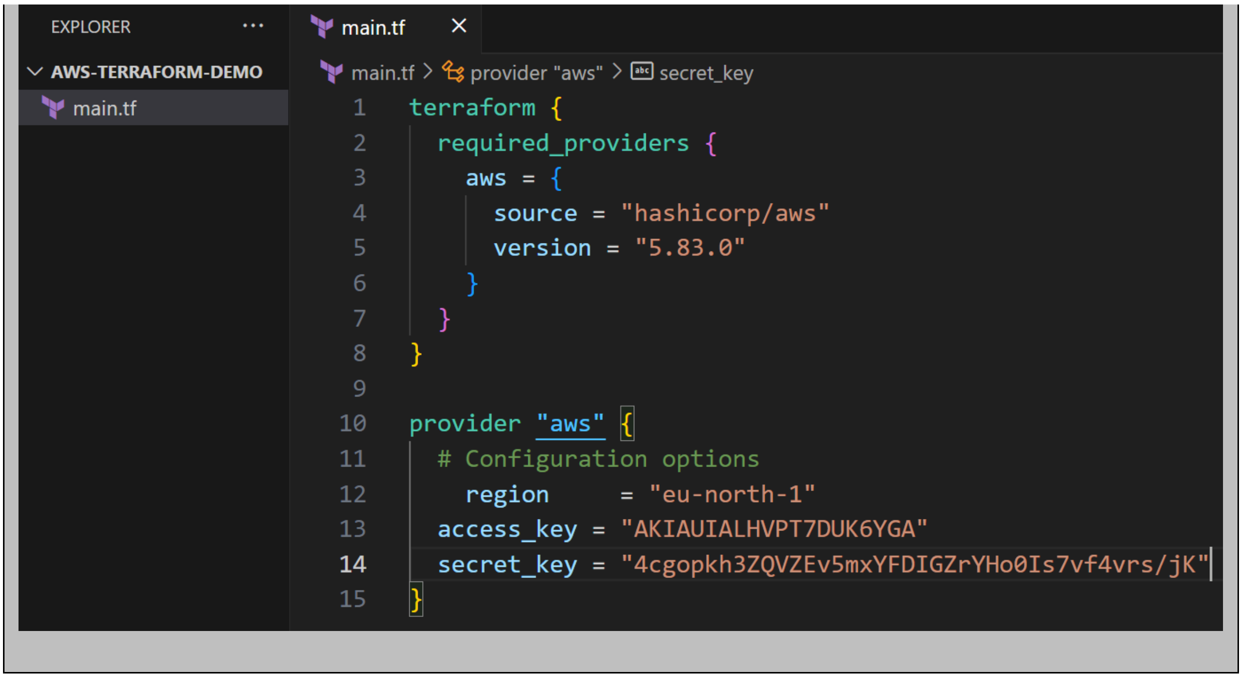
Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Creating access key



IAM > Users > anshika-terraform > Create access key

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.



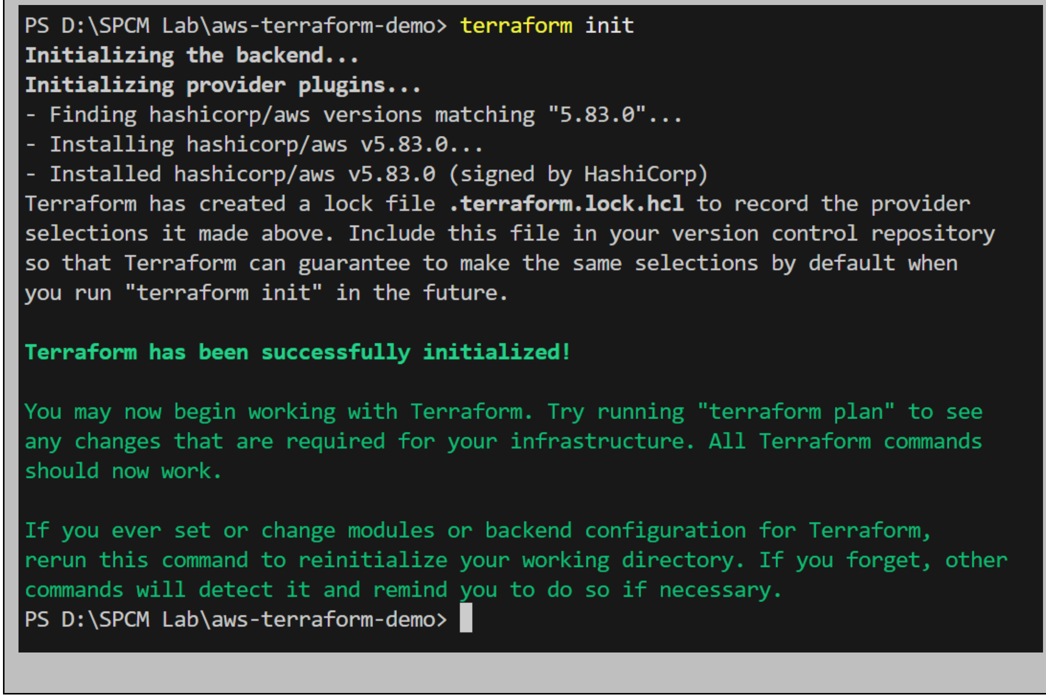
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.83.0"
    }
  }
}

provider "aws" {
  # Configuration options
    region      = "eu-north-1"
  access_key = "AKIAUIALHVPT7DUK6YGA"
  secret_key = "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"
}
```

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



```
PS D:\SPCM Lab\aws-terraform-demo> 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.
PS D:\SPCM Lab\aws-terraform-demo>
```