

Lab Exercise 3– Terraform 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:

“Terraform-Demo”

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

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

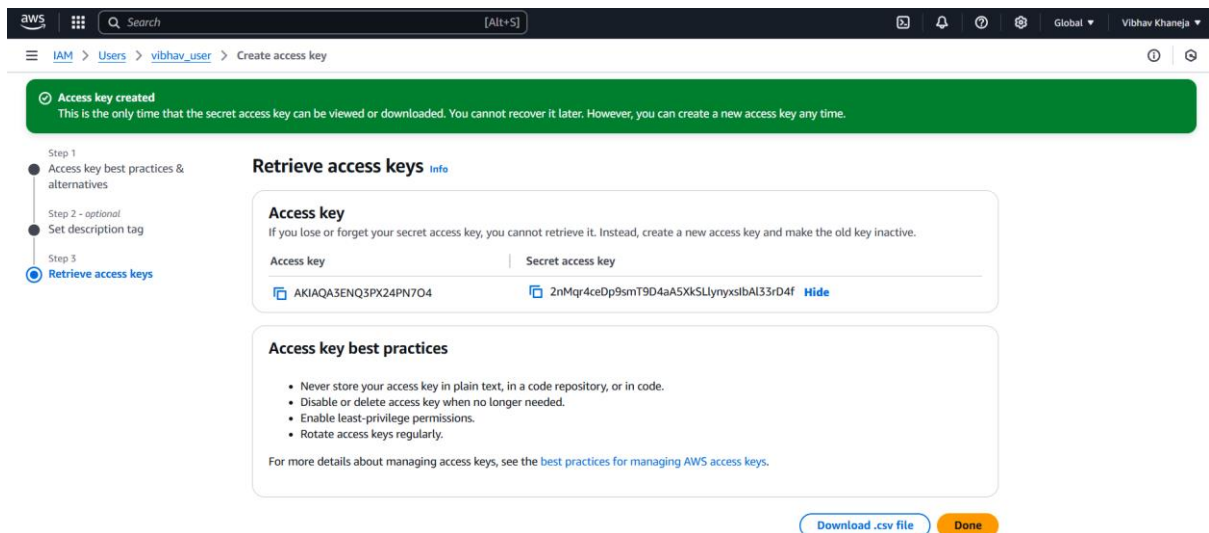
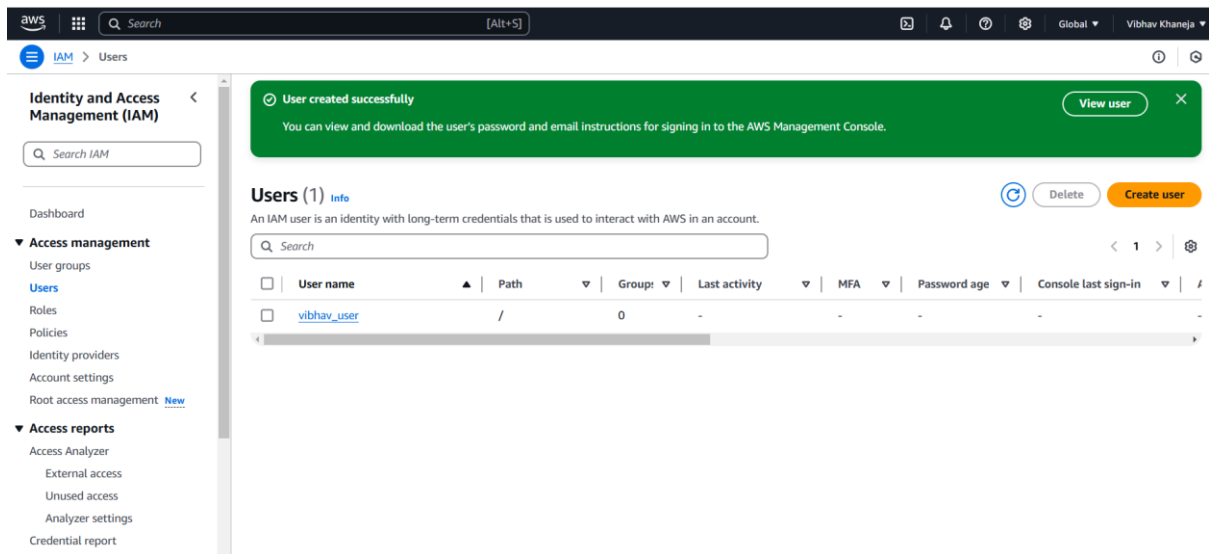
The screenshot shows the AWS IAM console 'Create user' wizard at the 'Review and create' step. The breadcrumb navigation at the top is 'IAM > Users > Create user'. On the left, a progress indicator shows three steps: 'Specify user details', 'Set permissions', and 'Review and create' (which is the active step). The main content area is titled 'Review and create' and includes a sub-header 'Review your choices. After you create the user, you can view and download the autogenerated password, if enabled.' Below this, there are three sections: 'User details', 'Permissions summary', and 'Tags - optional'. The 'User details' section shows 'User name' as 'vibhav_user', 'Console password type' as 'None', and 'Require password reset' as 'No'. The 'Permissions summary' section shows a table with one entry: 'AdministratorAccess' (Name), 'AWS managed - job function' (Type), and 'Permissions policy' (Used as). The 'Tags - optional' section states 'No tags associated with the resource.' and includes an 'Add new tag' button. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create user'.

User details		
User name	Console password type	Require password reset
vibhav_user	None	No

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

Tags - optional
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.
No tags associated with the resource.
[Add new tag](#)
You can add up to 50 more tags.

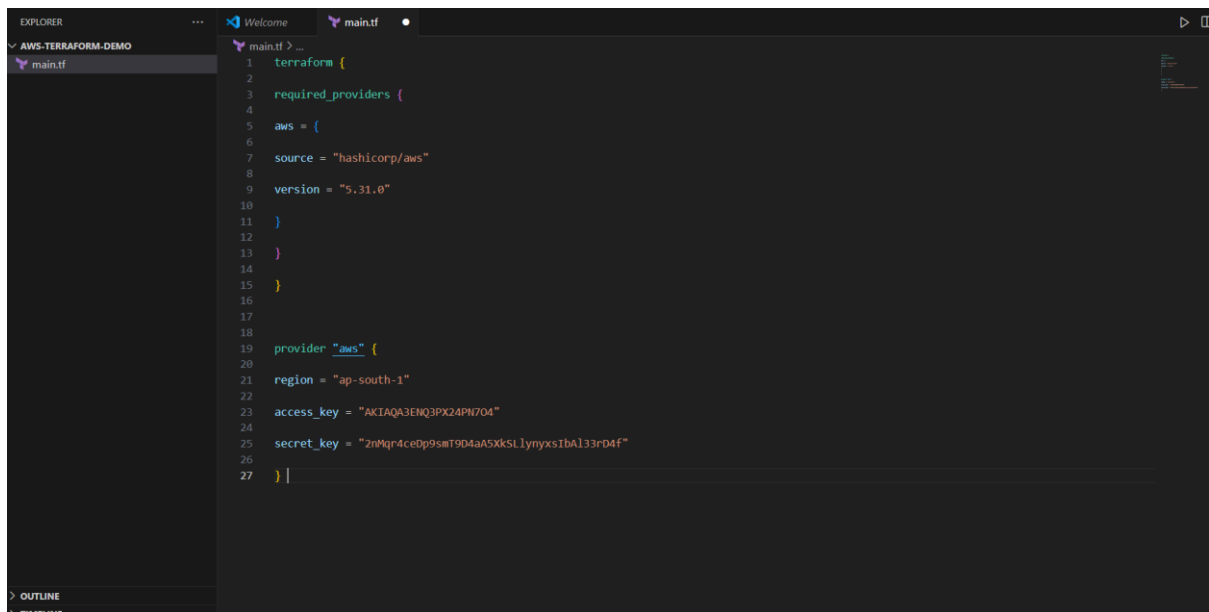
[Cancel](#) [Previous](#) [Create user](#)



```
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"  
}
```

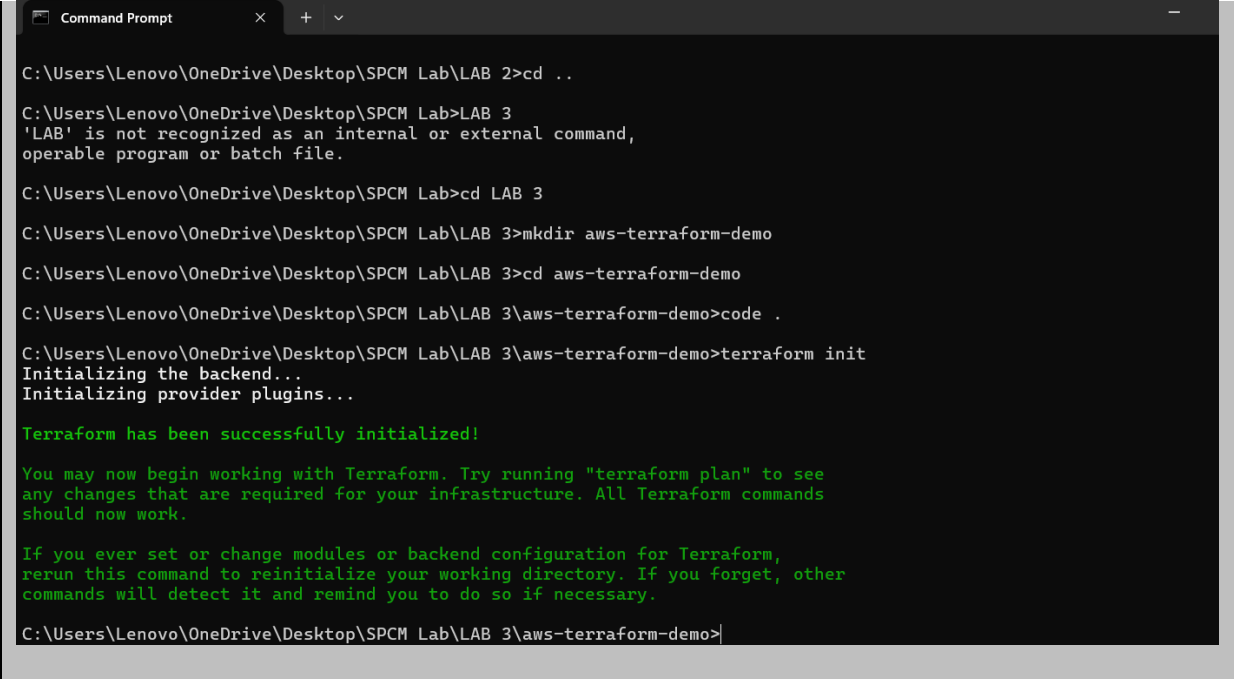


This script defines an AWS provider and provisions an EC2 instance.

Step 3: Initialize Terraform:

Run the following command to initialize your Terraform working directory:

```
terraform init
```



```
Command Prompt
C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 2>cd ..
C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 3>cd LAB 3
'LAB' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 3>mkdir aws-terraform-demo
C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 3>cd aws-terraform-demo
C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 3>code .
C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 3>aws-terraform-demo>terraform init
Initializing the backend...
Initializing provider plugins...

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.
C:\Users\Lenovo\OneDrive\Desktop\SPCM Lab\LAB 3>aws-terraform-demo>
```