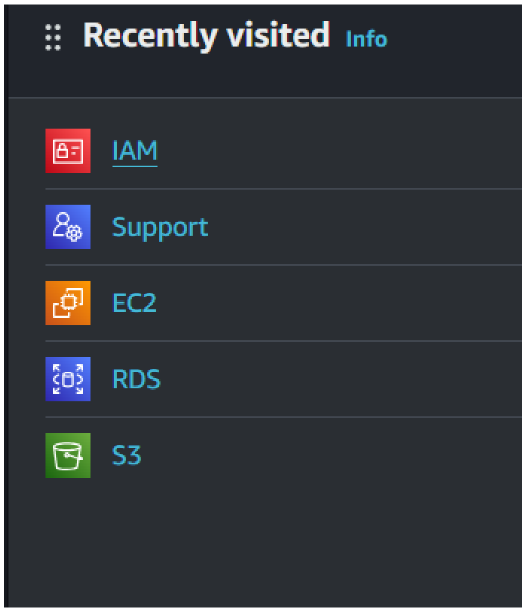
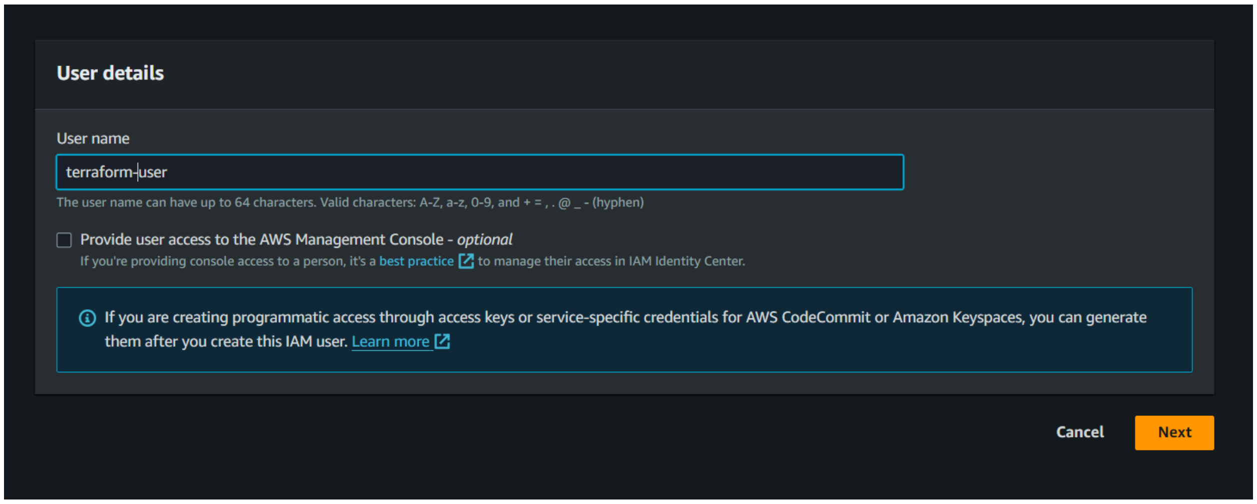
# Experiment 3

**Step 1:**

1)Create a IAM user form the AWS console



2)Write the name of the user



**User details**

User name

terraform-user

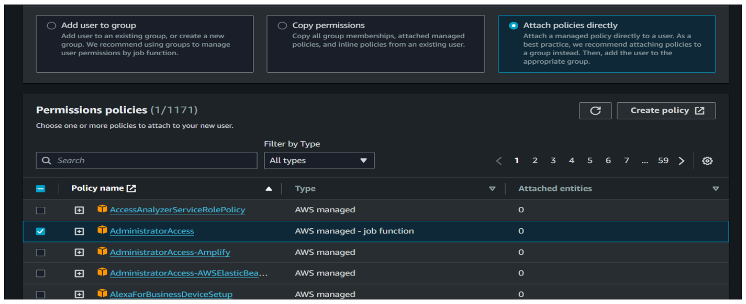
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

3)Set the permission policies



○ Add user to group — Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ Copy permissions — Copy all group memberships, attached managed policies, and inline policies from an existing user.

● Attach policies directly — Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

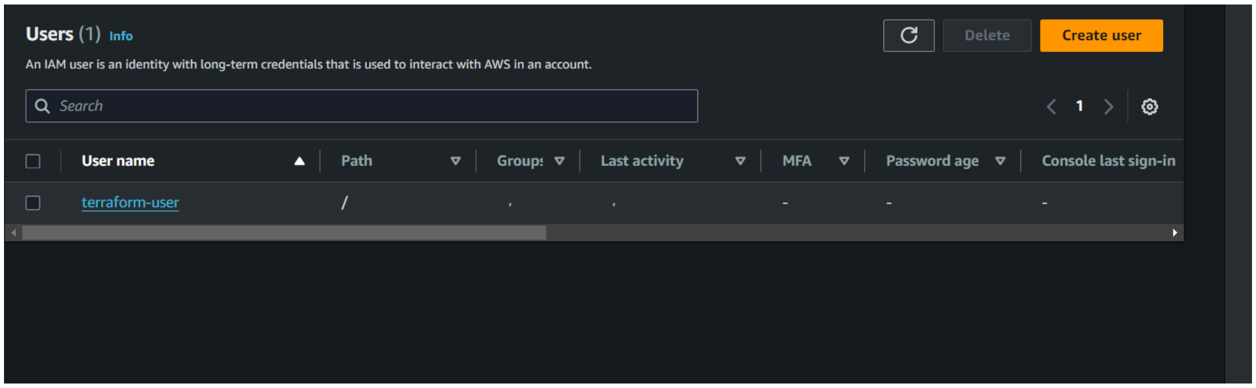
**Permissions policies** (1/1171)

Choose one or more policies to attach to your new user.

Filter by Type: All types

| | Policy name | Type | Attached entities |
|---|---|---|---|
| ☐ | AccessAnalyzerServiceRolePolicy | AWS managed | 0 |
| ✓ | AdministratorAccess | AWS managed - job function | 0 |
| ☐ | AdministratorAccess-Amplify | AWS managed | 0 |
| ☐ | AdministratorAccess-AWSElasticBea... | AWS managed | 0 |
| ☐ | AlexaForBusinessDeviceSetup | AWS managed | 0 |

4)You can now view the user

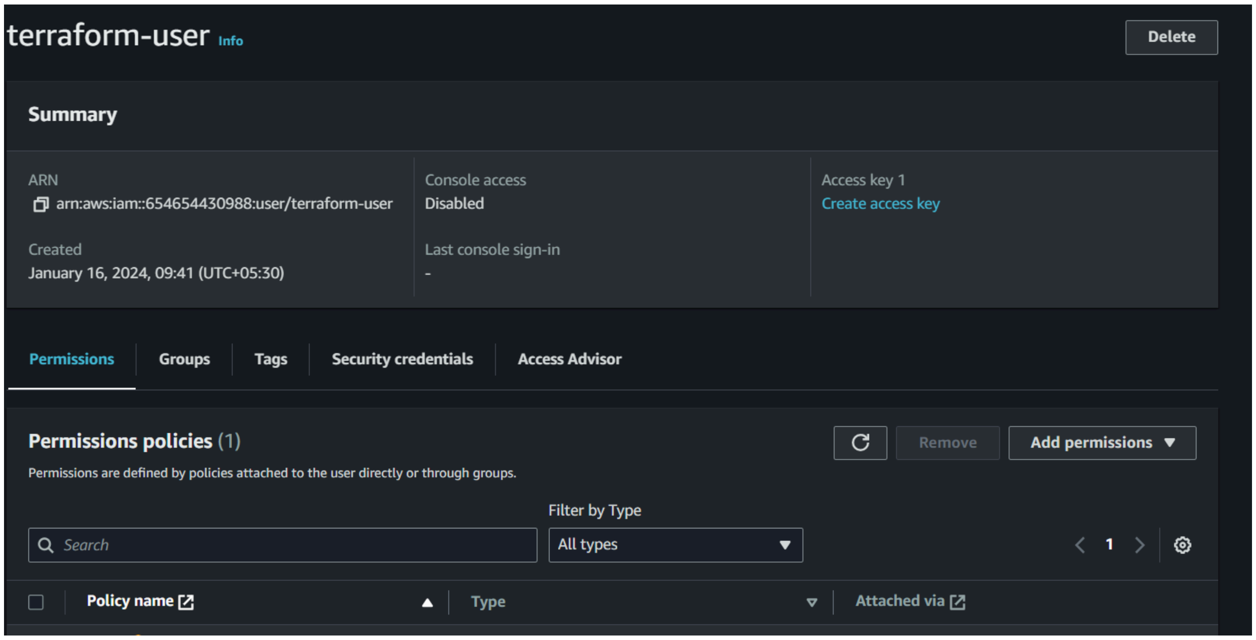


An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

| | User name | Path | Groups | Last activity | MFA | Password age | Console last sign-in |
|---|---|---|---|---|---|---|---|
| | terraform-user | / | , | , | - | - | - |

## Step 2: Create the access key

1)Click on create access key



ARN
arn:aws:iam::654654430988:user/terraform-user

Created
January 16, 2024, 09:41 (UTC+05:30)

Console access
Disabled

Last console sign-in
-

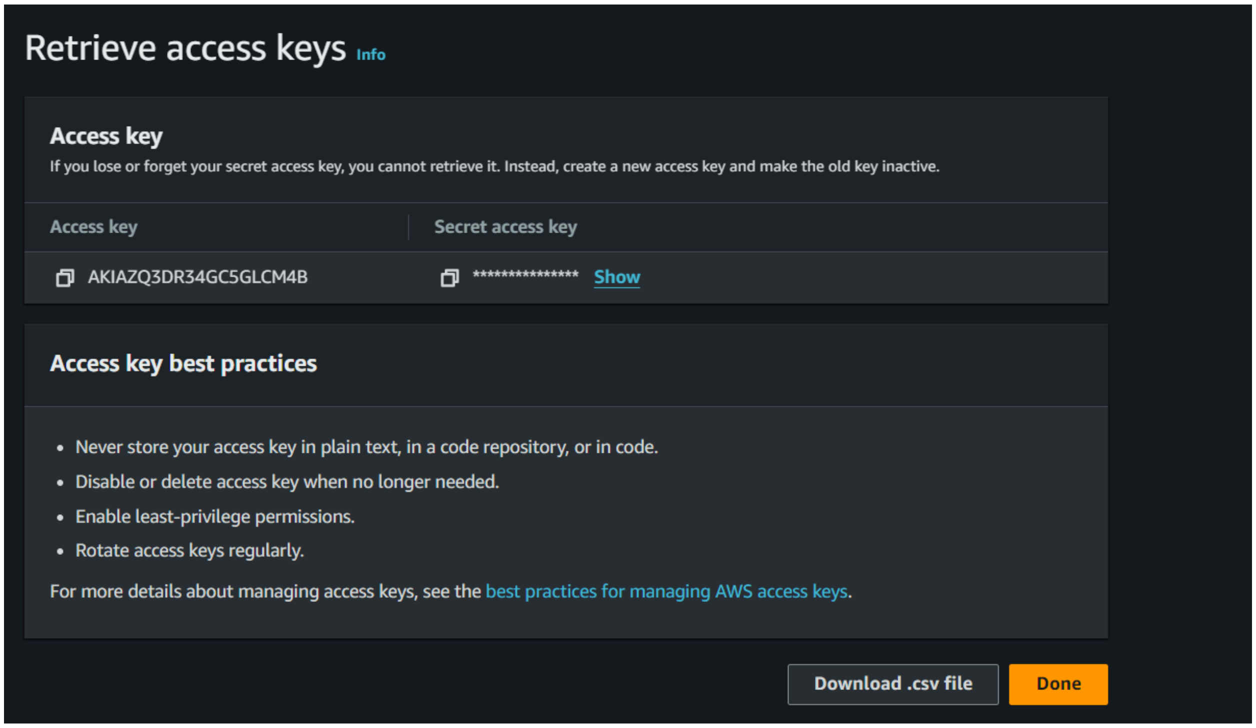
Access key 1
Create access key

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type
All types

| | Policy name | Type | Attached via |
|---|---|---|---|

2)Download the CSV file or copy paste the access key



Retrieve access keys Info

**Access key**

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

| Access key | Secret access key |
|---|---|
| AKIAZQ3DR34GC5GLCM4B | *************** Show |

**Access key best practices**

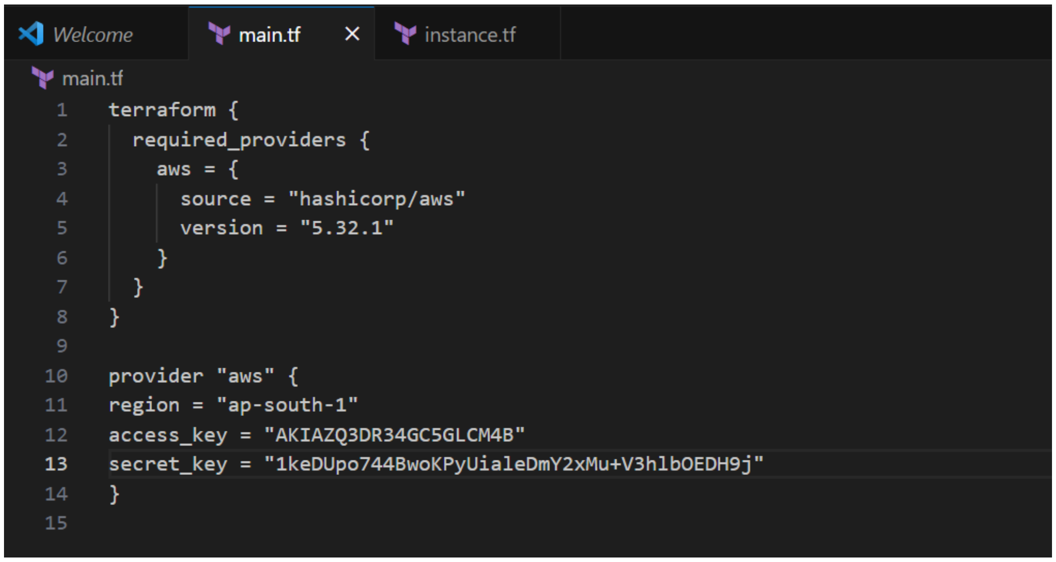
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

**Step 3: Update the terraform file**

1)Add the following contents to the main.tf file

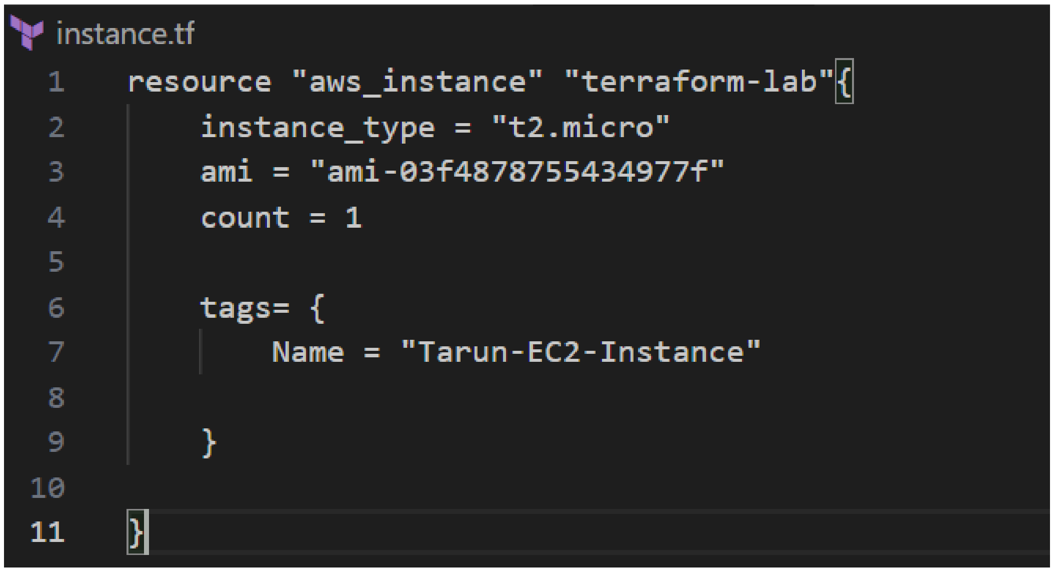
provider "aws" {
region = "ap-south-1"
access_key = "AKIAZQ3DR34GC5GLCM4B"
secret_key = "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"
}



```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.32.1"
    }
  }
}

provider "aws" {
region = "ap-south-1"
access_key = "AKIAZQ3DR34GC5GLCM4B"
secret_key = "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"
}

```

2)Create a instance.tf terraform file with the following contents

```
resource "aws_instance" "terraform-lab"{
   instance_type = "t2.micro"
   ami = "ami-03f4878755434977f"
   count = 1

   tags= {
      Name = "Tarun-EC2-Instance"

   }

}
```

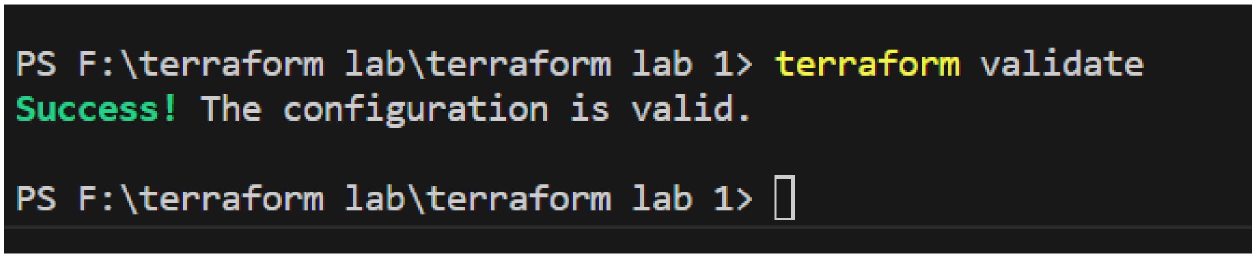


instance.tf

```
resource "aws_instance" "terraform-lab"{
    instance_type = "t2.micro"
    ami = "ami-03f4878755434977f"
    count = 1

    tags= {
        Name = "Tarun-EC2-Instance"

    }

}
```

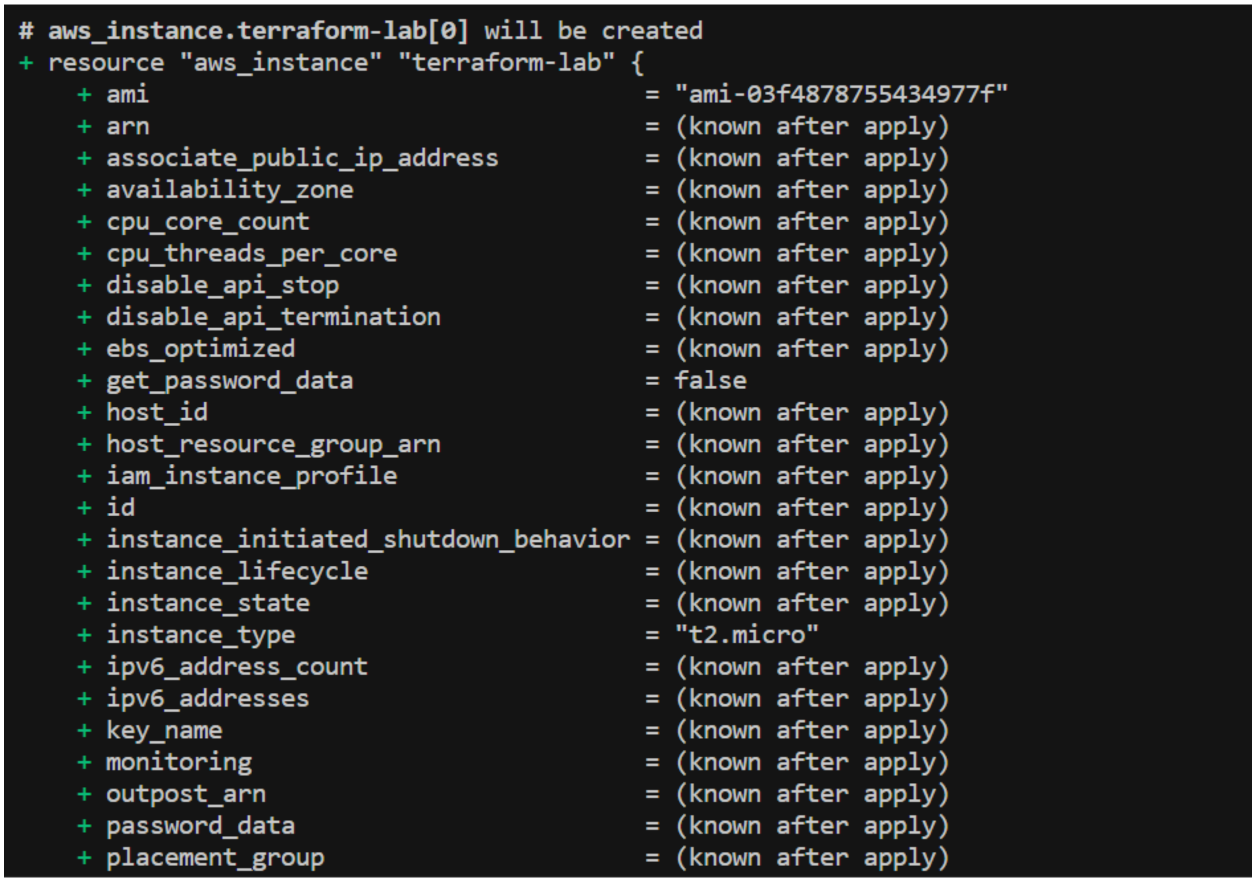
**Step 4: Run the terraform commands**

1)Run the validate command



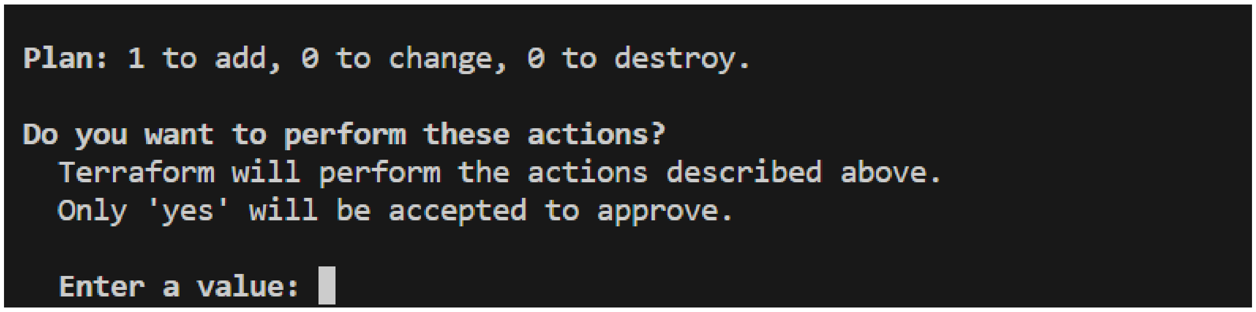
```
PS F:\terraform lab\terraform lab 1> terraform validate
Success! The configuration is valid.

PS F:\terraform lab\terraform lab 1> 
```

2)Run the plan command

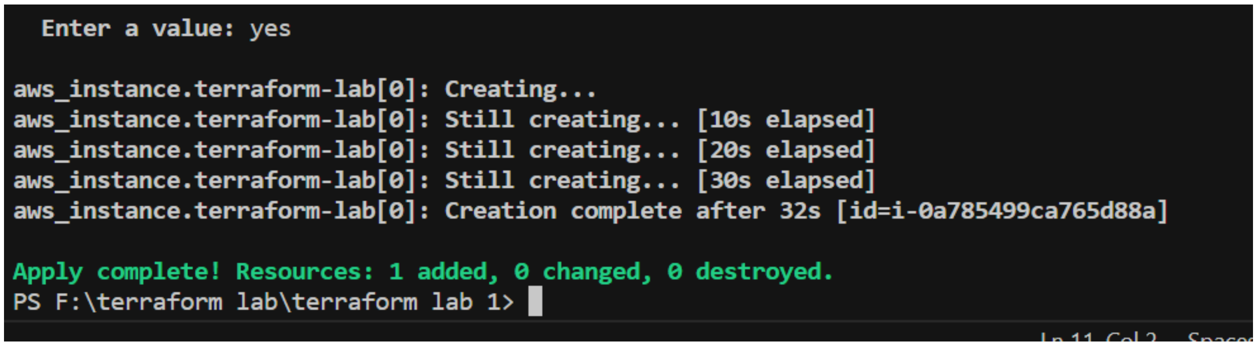


```
# aws_instance.terraform-lab[0] will be created
+ resource "aws_instance" "terraform-lab" {
    + ami                                  = "ami-03f4878755434977f"
    + arn                                  = (known after apply)
    + associate_public_ip_address          = (known after apply)
    + availability_zone                    = (known after apply)
    + cpu_core_count                       = (known after apply)
    + cpu_threads_per_core                 = (known after apply)
    + disable_api_stop                     = (known after apply)
    + disable_api_termination              = (known after apply)
    + ebs_optimized                        = (known after apply)
    + get_password_data                    = false
    + host_id                              = (known after apply)
    + host_resource_group_arn              = (known after apply)
    + iam_instance_profile                 = (known after apply)
    + id                                   = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                   = (known after apply)
    + instance_state                       = (known after apply)
    + instance_type                        = "t2.micro"
    + ipv6_address_count                   = (known after apply)
    + ipv6_addresses                       = (known after apply)
    + key_name                             = (known after apply)
    + monitoring                           = (known after apply)
    + outpost_arn                          = (known after apply)
    + password_data                        = (known after apply)
    + placement_group                      = (known after apply)
```

3)Run the apply command

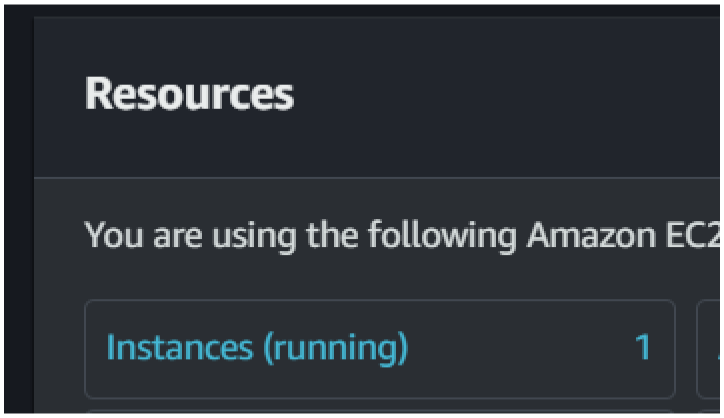


```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value:
```



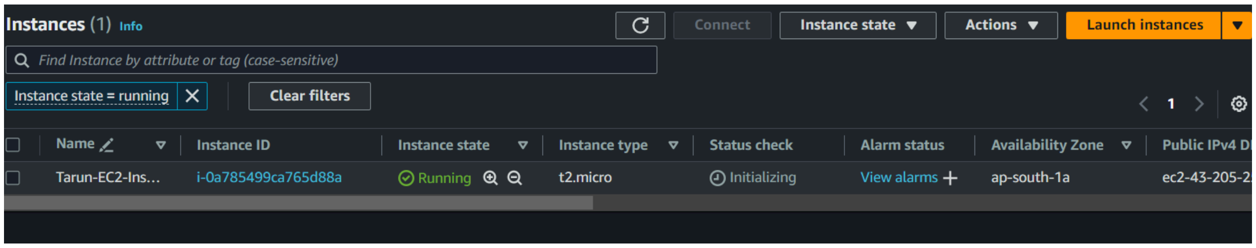
```
  Enter a value: yes

aws_instance.terraform-lab[0]: Creating...
aws_instance.terraform-lab[0]: Still creating... [10s elapsed]
aws_instance.terraform-lab[0]: Still creating... [20s elapsed]
aws_instance.terraform-lab[0]: Still creating... [30s elapsed]
aws_instance.terraform-lab[0]: Creation complete after 32s [id=i-0a785499ca765d88a]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS F:\terraform lab\terraform lab 1>
```

**Step 5: Check the aws console for the instances**


Resources

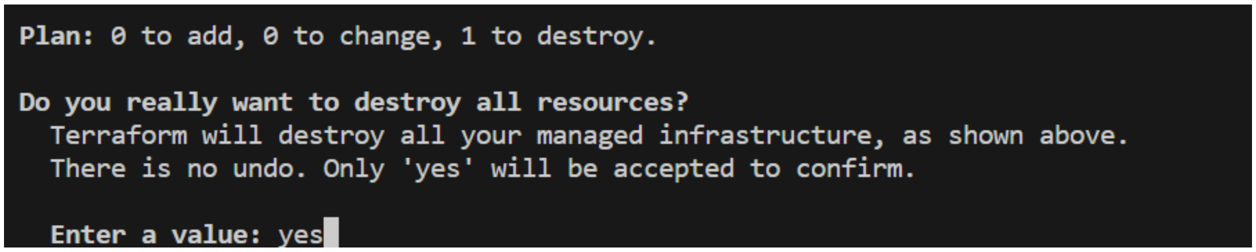
You are using the following Amazon EC2

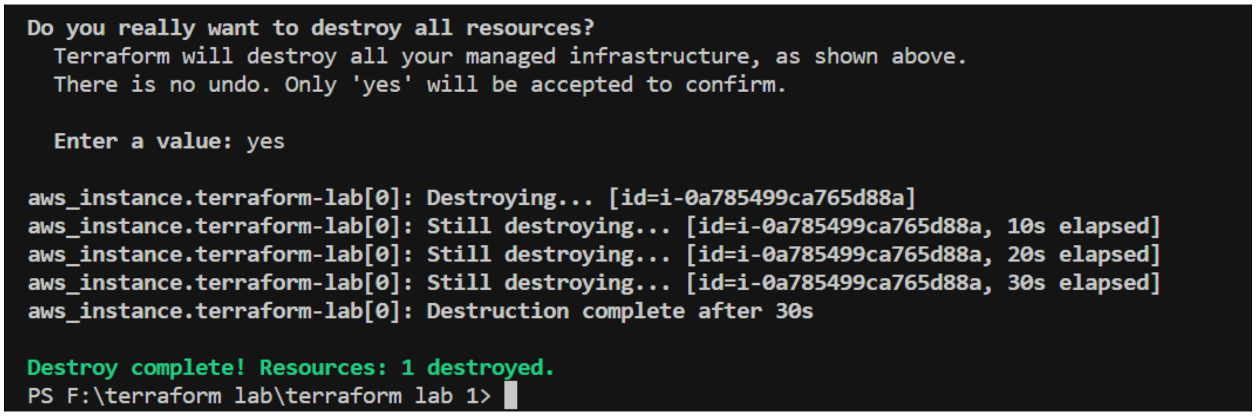
Instances (running) 1

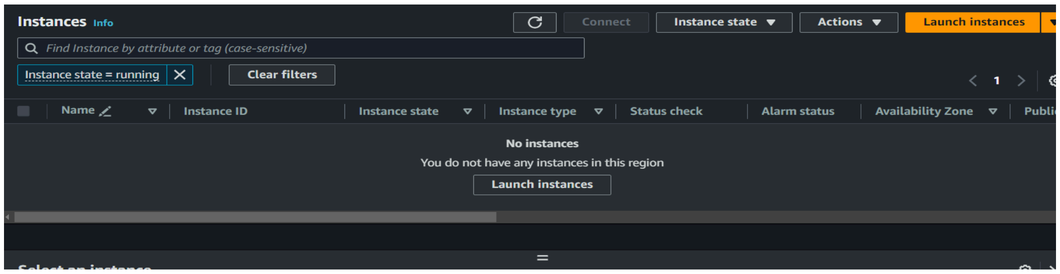

Instances (1) Info

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 D |
|---|---|---|---|---|---|---|---|
| Tarun-EC2-Ins... | i-0a785499ca765d88a | Running | t2.micro | Initializing | View alarms | ap-south-1a | ec2-43-205-2 |

**Step 6: Destroy the instance**


```
Plan: 0 to add, 0 to change, 1 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes
```


```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.terraform-lab[0]: Destroying... [id=i-0a785499ca765d88a]
aws_instance.terraform-lab[0]: Still destroying... [id=i-0a785499ca765d88a, 10s elapsed]
aws_instance.terraform-lab[0]: Still destroying... [id=i-0a785499ca765d88a, 20s elapsed]
aws_instance.terraform-lab[0]: Still destroying... [id=i-0a785499ca765d88a, 30s elapsed]
aws_instance.terraform-lab[0]: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
PS F:\terraform lab\terraform lab 1>
```


Instances Info

No instances

You do not have any instances in this region