Task 7

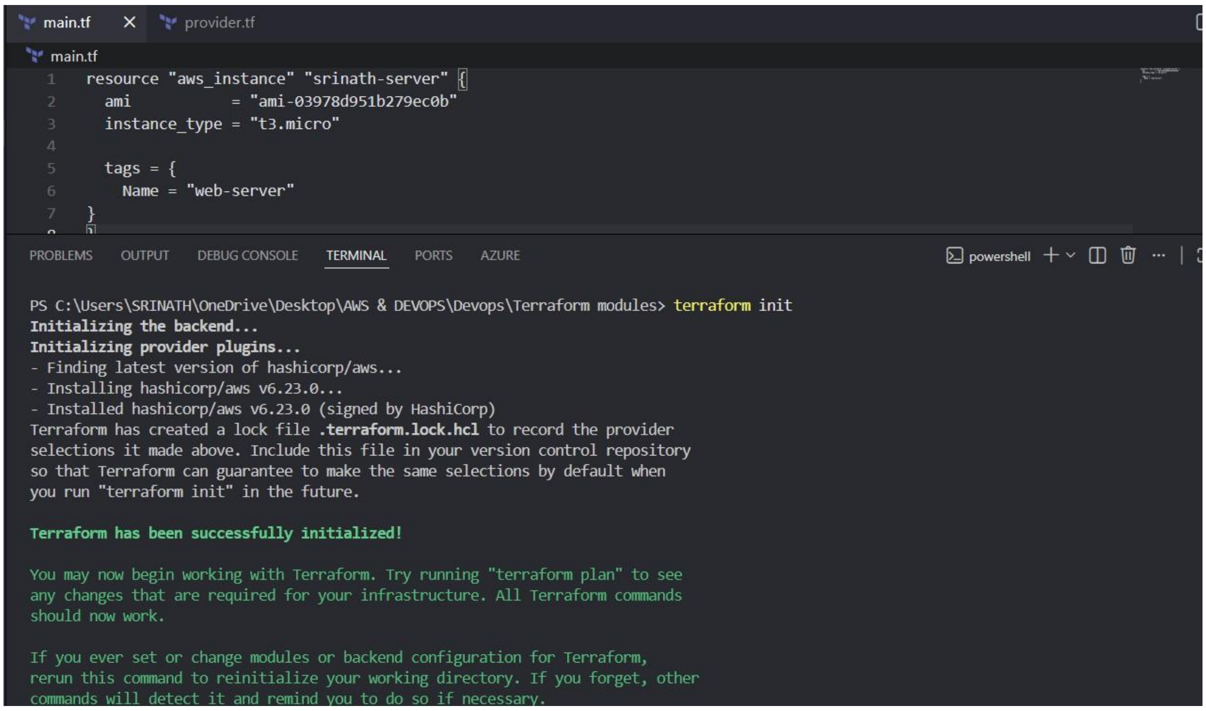
Terraform:

- Terraform is an open-source infrastructure as code (IaC) tool developed by HashiCorp that allows users to define and provision cloud and on-premises resources using a declarative configuration language.
- It enables users to manage infrastructure throughout its lifecycle by writing human□readable configuration files that can be versioned, reused, and shared.

Terraform Command:

Creation of EC2

1) terraform init:



```
resource "aws_instance" "srinath-server" {
  ami           = "ami-03978d951b279ec0b"
  instance_type = "t3.micro"

  tags = {
    Name = "web-server"
  }
}
```

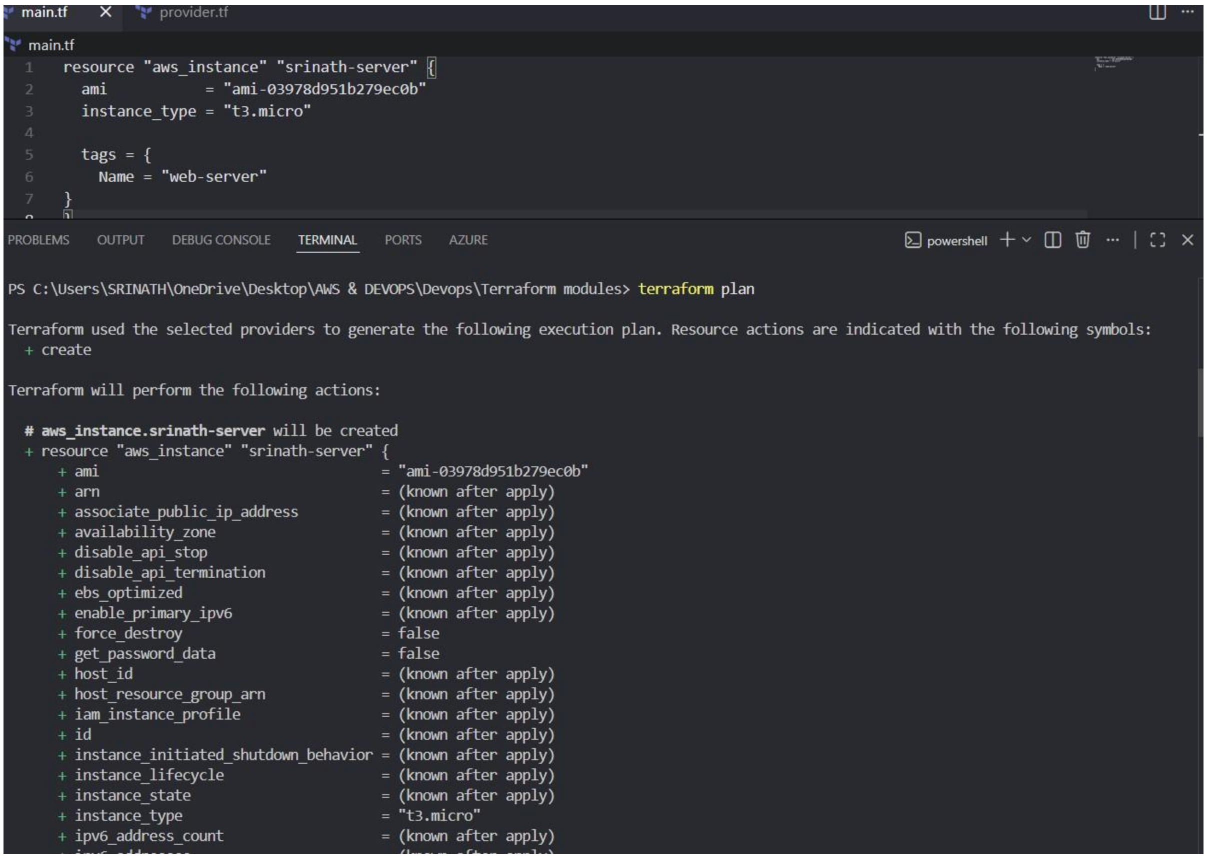
```
PS C:\Users\SRINATH\OneDrive\Desktop\AWS & DEVOPS\Devops\Terraform modules> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.23.0...
- Installed hashicorp/aws v6.23.0 (signed by HashiCorp)
Terraform 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.
```

2) terraform plan:



```
resource "aws_instance" "srinath-server" {
  ami           = "ami-03978d951b279ec0b"
  instance_type = "t3.micro"

  tags = {
    Name = "web-server"
  }
```

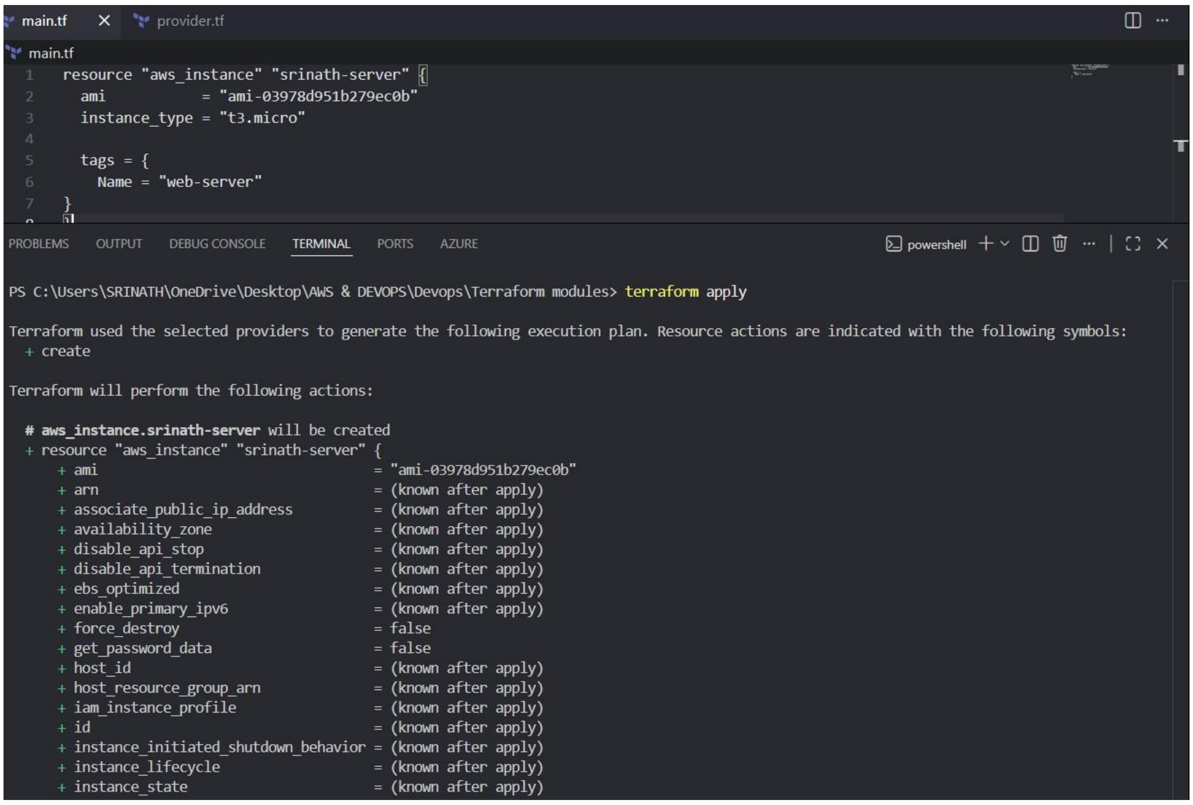
```
PS C:\Users\SRINATH\OneDrive\Desktop\AWS & DEVOPS\Devops\Terraform modules> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.srinath-server will be created
  + resource "aws_instance" "srinath-server" {
      + ami                                  = "ami-03978d951b279ec0b"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + 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                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
```

3) terraform apply



```
resource "aws_instance" "srinath-server" {
  ami           = "ami-03978d951b279ec0b"
  instance_type = "t3.micro"

  tags = {
    Name = "web-server"
  }
```

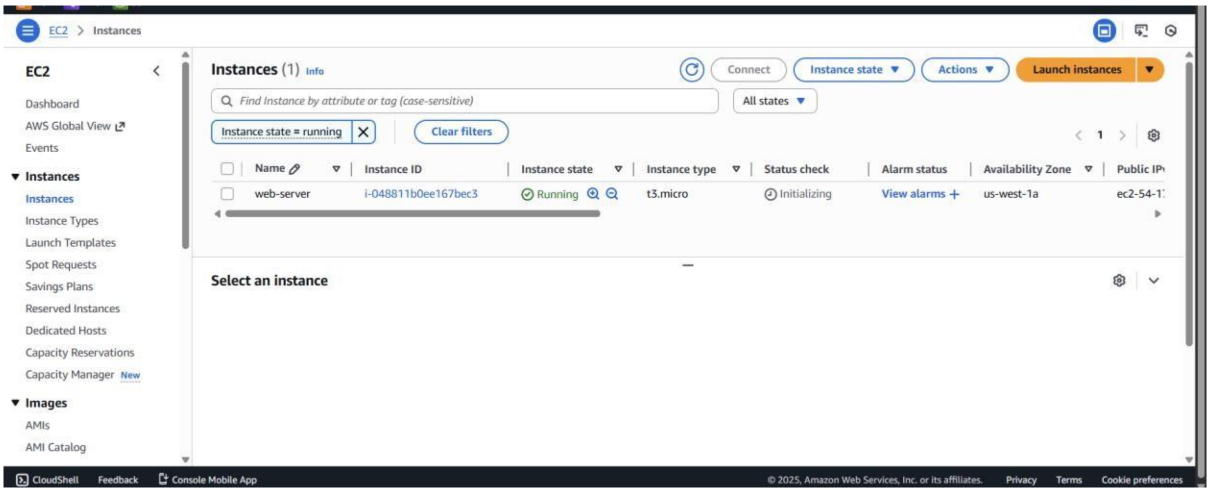
```
PS C:\Users\SRINATH\OneDrive\Desktop\AWS & DEVOPS\Devops\Terraform modules> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.srinath-server will be created
  + resource "aws_instance" "srinath-server" {
      + ami                                  = "ami-03978d951b279ec0b"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + enable_primary_ipv6                  = (known after apply)
      + force_destroy                        = false
      + 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)
```

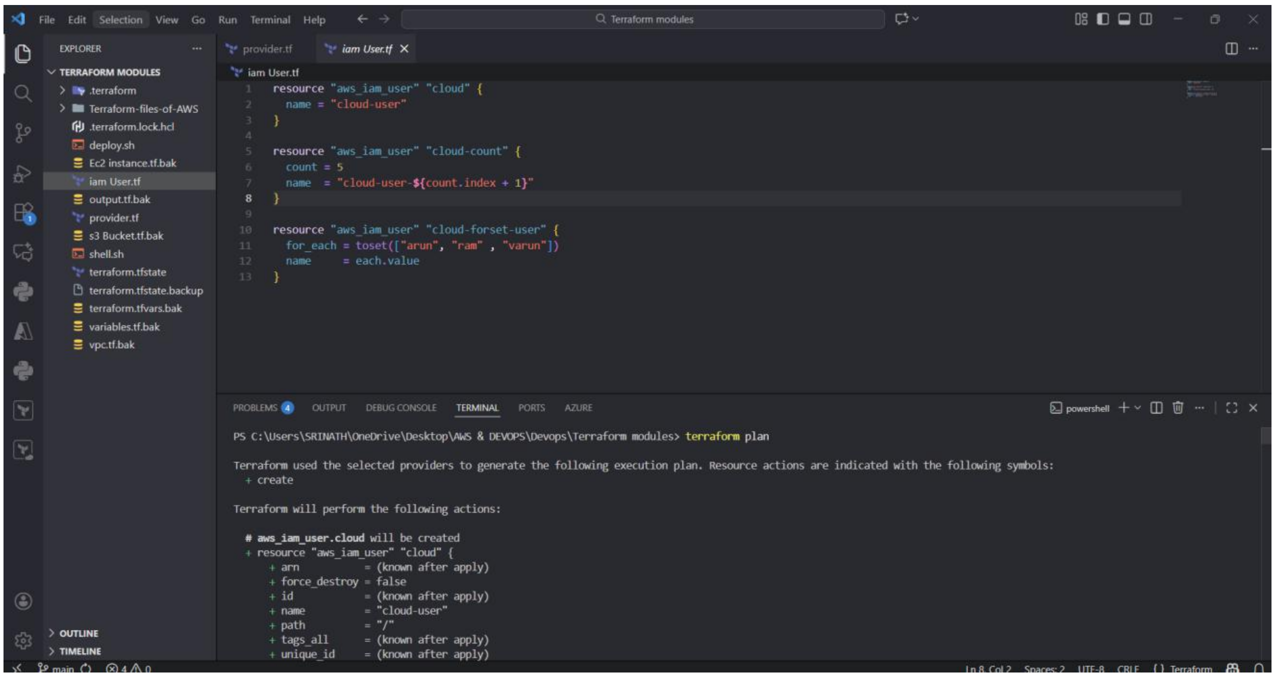
Output



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
	web-server	i-048811b0ee167bec3	Running	t3.micro	Initializing	View alarms +	us-west-1a	ec2-54-1

Creation of IAM:

1) terraform plan:



```
resource "aws_iam_user" "cloud" {
  name = "cloud-user"
}

resource "aws_iam_user" "cloud-count" {
  count = 5
  name  = "cloud-user-${count.index + 1}"
}

resource "aws_iam_user" "cloud-forset-user" {
  for_each = toset(["arun", "ram" , "varun"])
  name     = each.value
}
```

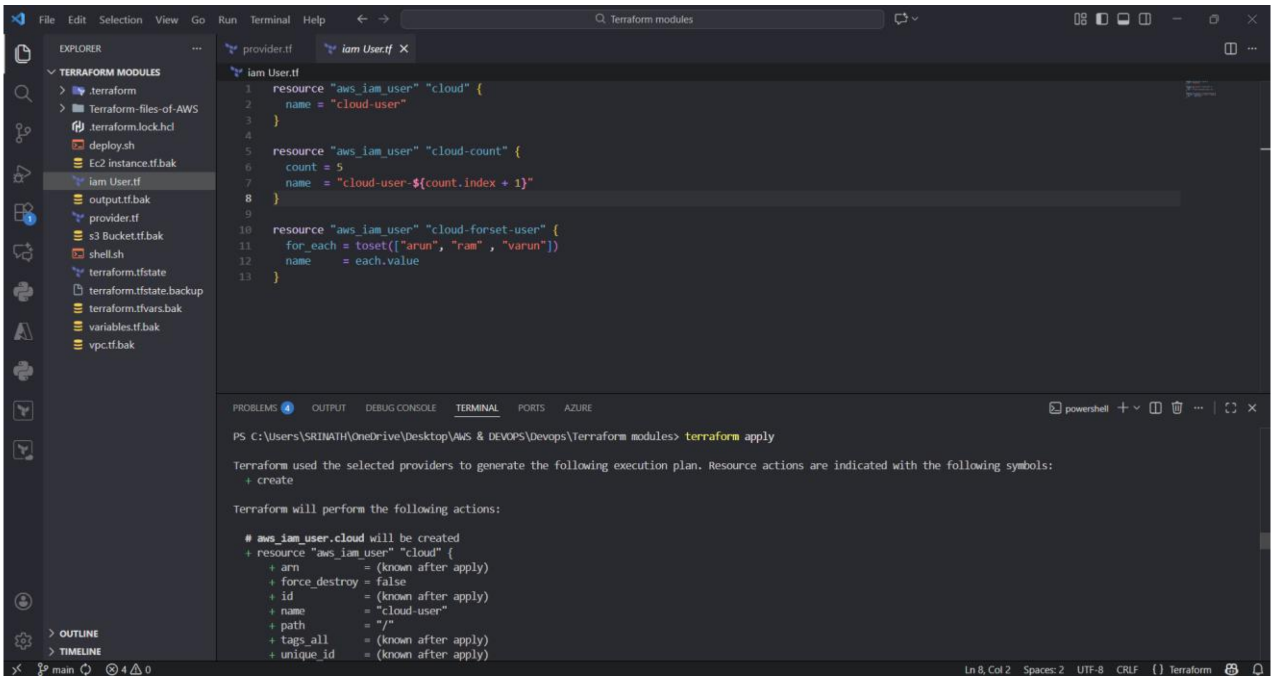
```
PS C:\Users\SRINATH\OneDrive\Desktop\AWS & DEVOPS\Devops\Terraform modules> terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_iam_user.cloud will be created
  + resource "aws_iam_user" "cloud" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "cloud-user"
      + path          = "/"
      + tags_all      = (known after apply)
      + unique_id     = (known after apply)
```

2) terraform apply:



iam User.tf

```
resource "aws_iam_user" "cloud" {
  name = "cloud-user"
}

resource "aws_iam_user" "cloud-count" {
  count = 5
  name  = "cloud-user-${count.index + 1}"
}

resource "aws_iam_user" "cloud-forset-user" {
  for_each = toset(["arun", "ram" , "varun"])
  name     = each.value
}
```

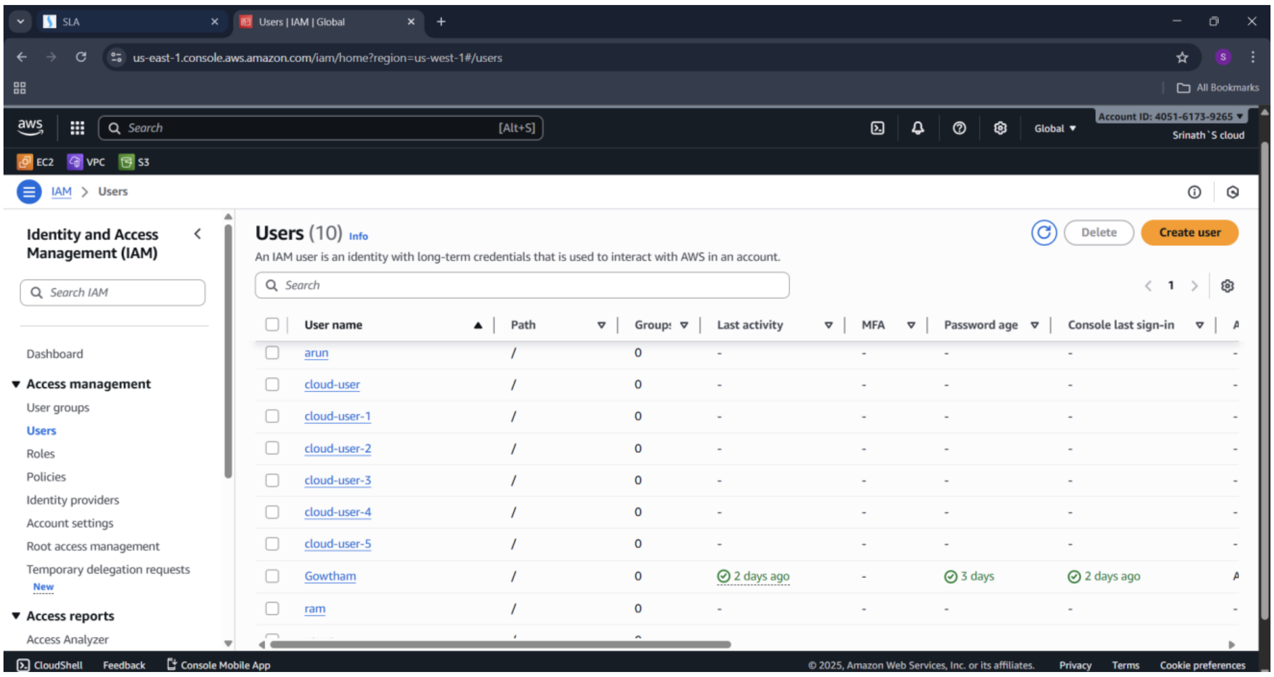
```
PS C:\Users\SRINATH\OneDrive\Desktop\AWS & DEVOPS\Devops\Terraform modules> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_iam_user.cloud will be created
  + resource "aws_iam_user" "cloud" {
      + arn           = (known after apply)
      + force_destroy = false
      + id            = (known after apply)
      + name          = "cloud-user"
      + path          = "/"
      + tags_all      = (known after apply)
      + unique_id     = (known after apply)
```

Output:



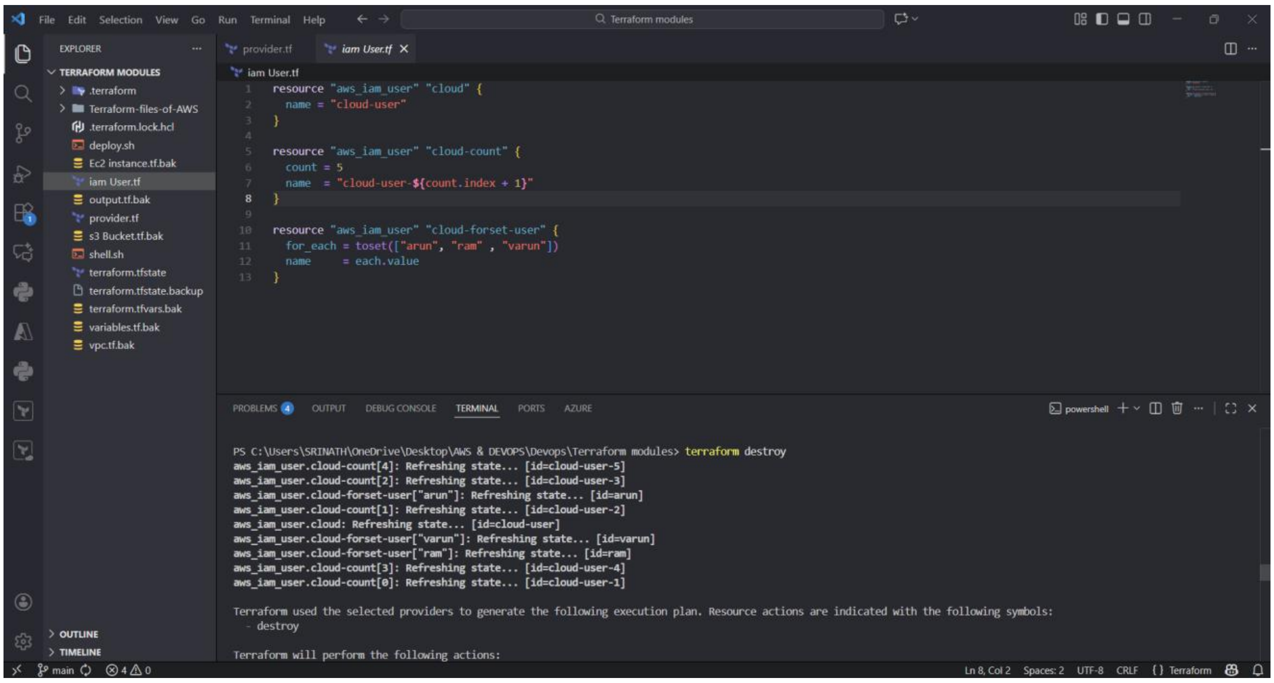
us-east-1.console.aws.amazon.com/iam/home?region=us-west-1#/users

Users (10)

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
arun	/	0	-	-	-	-
cloud-user	/	0	-	-	-	-
cloud-user-1	/	0	-	-	-	-
cloud-user-2	/	0	-	-	-	-
cloud-user-3	/	0	-	-	-	-
cloud-user-4	/	0	-	-	-	-
cloud-user-5	/	0	-	-	-	-
Gowtham	/	0	✓ 2 days ago	-	✓ 3 days	✓ 2 days ago
ram	/	0	-	-	-	-

3) terraform destroy:



```
resource "aws_iam_user" "cloud" {
  name = "cloud-user"
}

resource "aws_iam_user" "cloud-count" {
  count = 5
  name  = "cloud-user-${count.index + 1}"
}

resource "aws_iam_user" "cloud-forset-user" {
  for_each = toset(["arun", "ram" , "varun"])
  name     = each.value
}
```

```
PS C:\Users\SRINATH\OneDrive\Desktop\AWS & DEVOPS\Devops\Terraform modules> terraform destroy
aws_iam_user.cloud-count[4]: Refreshing state... [id=cloud-user-5]
aws_iam_user.cloud-count[2]: Refreshing state... [id=cloud-user-3]
aws_iam_user.cloud-forset-user["arun"]: Refreshing state... [id=arun]
aws_iam_user.cloud-count[1]: Refreshing state... [id=cloud-user-2]
aws_iam_user.cloud: Refreshing state... [id=cloud-user]
aws_iam_user.cloud-forset-user["varun"]: Refreshing state... [id=varun]
aws_iam_user.cloud-forset-user["ram"]: Refreshing state... [id=ram]
aws_iam_user.cloud-count[3]: Refreshing state... [id=cloud-user-4]
aws_iam_user.cloud-count[0]: Refreshing state... [id=cloud-user-1]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:
```