

NAME:- ROHIT SANJAY PATIL

ROLL NO:-612040

SUB:- ADVANCE DEVOPS

Experiment no:- 13

1.What is terraform?

Ans:- Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share. You can then use a consistent workflow to provision and manage all of your infrastructure throughout its lifecycle.

2.What is Infrastructure as a Code (IaC) ?

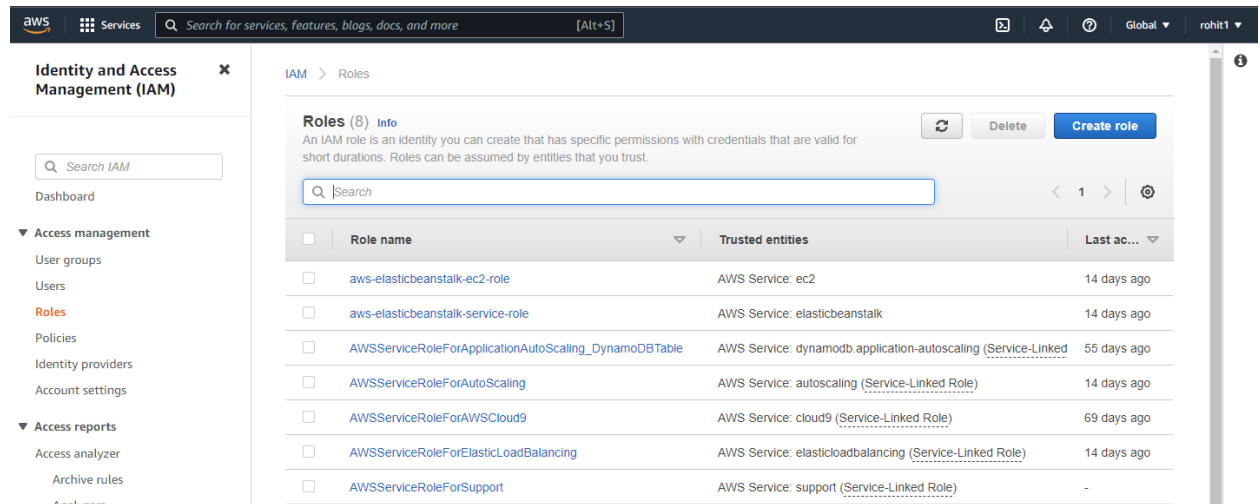
Ans:- Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.

With IaC, configuration files are created that contain your infrastructure specifications, which makes it easier to edit and distribute configurations. It also ensures that you provision the same environment every time.

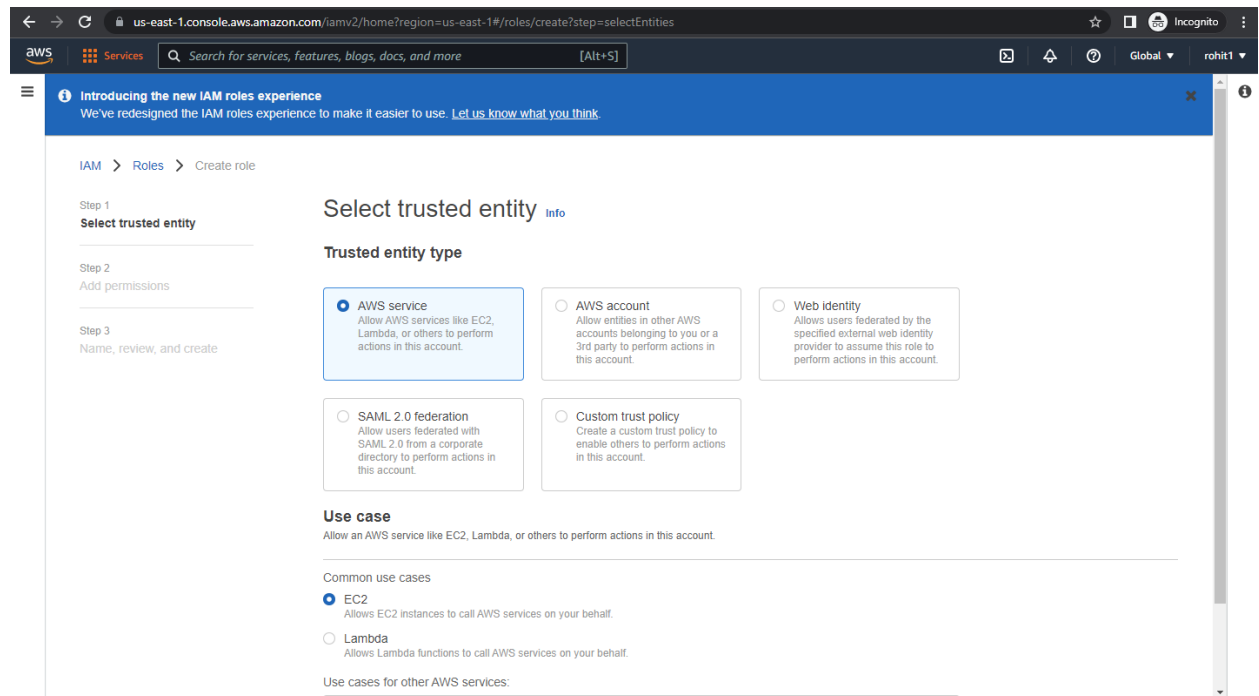
Infrastructure as code is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.

3. Perform an experiment, to understand Terraform lifecycle, core concepts/terminologies and install it on a Linux Machine.

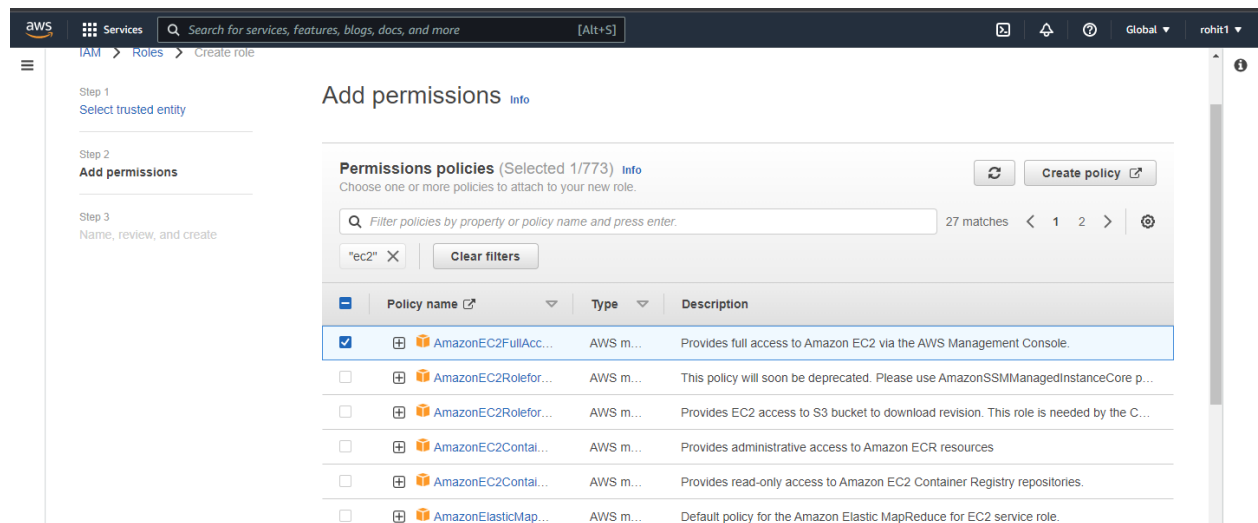
Step 1:- create new iam role



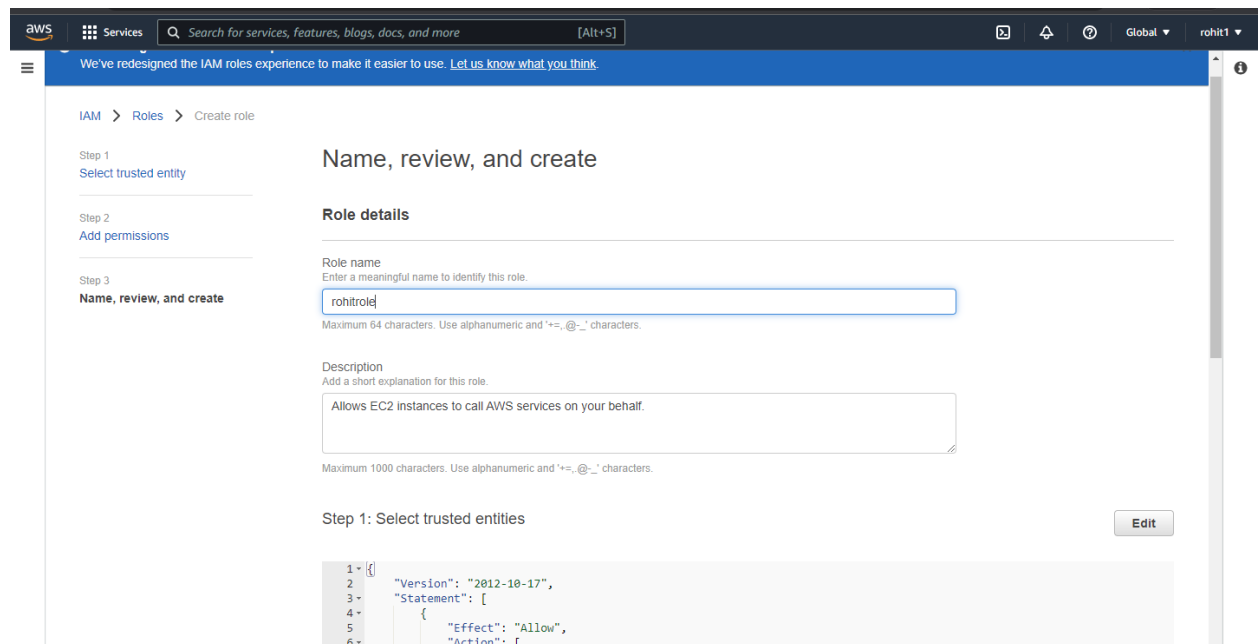
Step 2:- select aws service and Ec2



Step 3:- select amazonEC2fullaccess the click on next

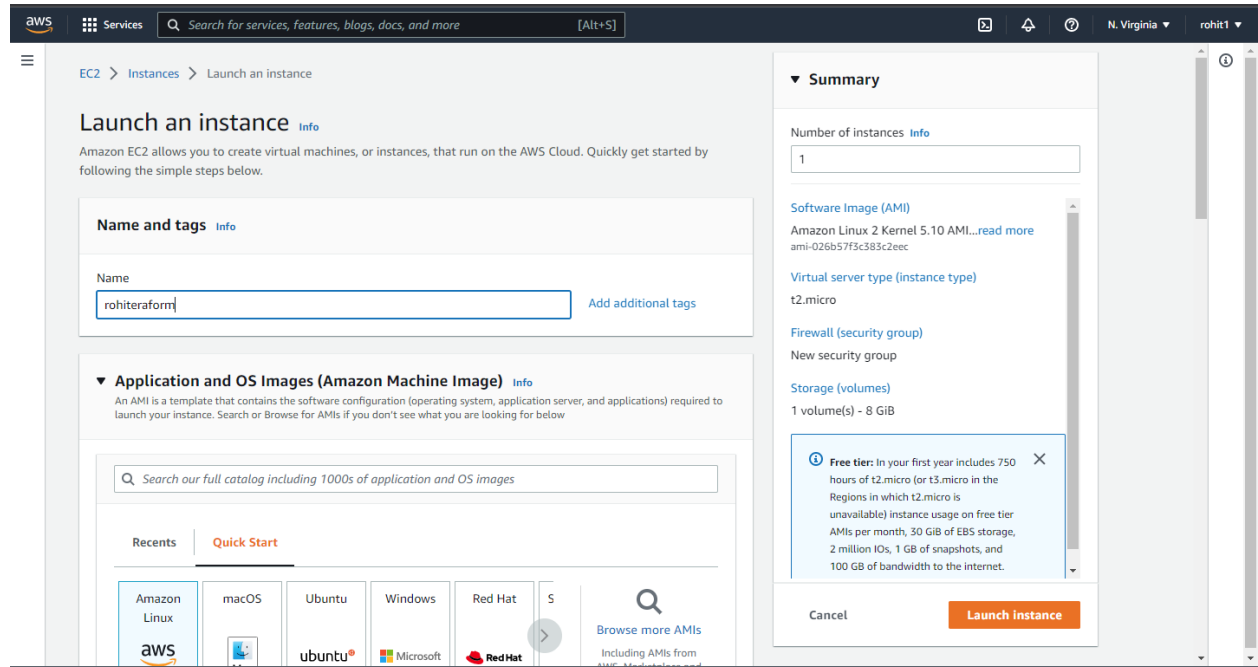


Step 4:-give any name to role and click on create role

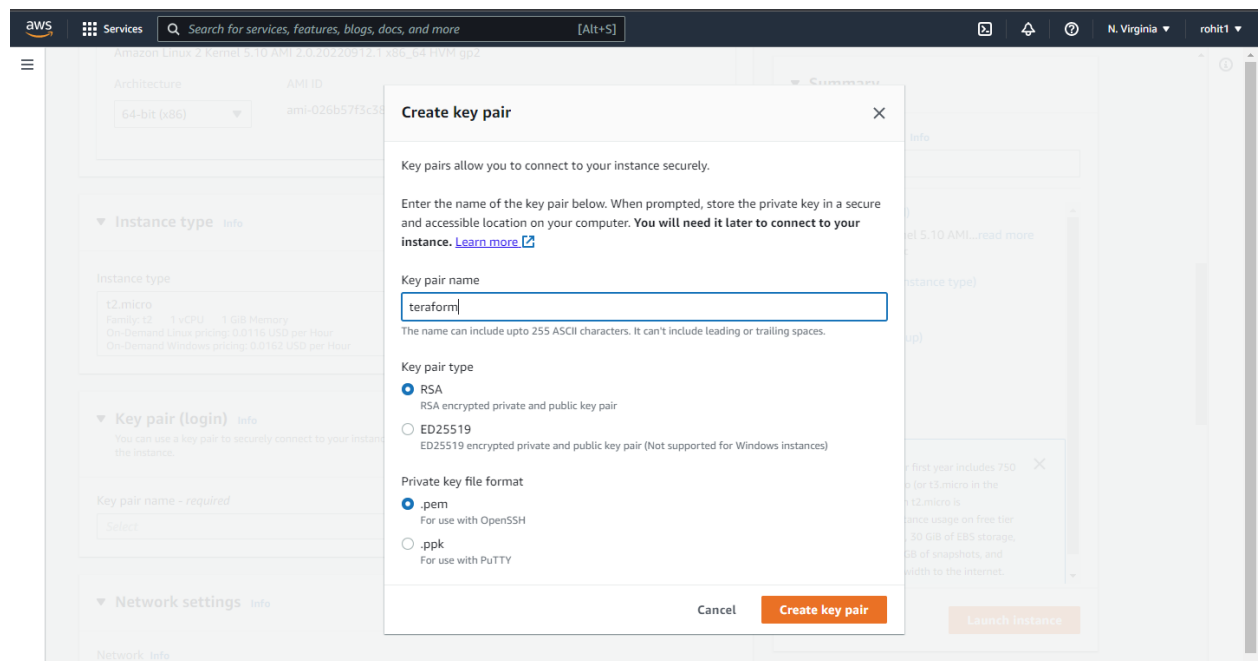


4. Using Terraform , create an EC2 instance on AWS cloud

Step 1:- now create Ec2 instance select linux machine



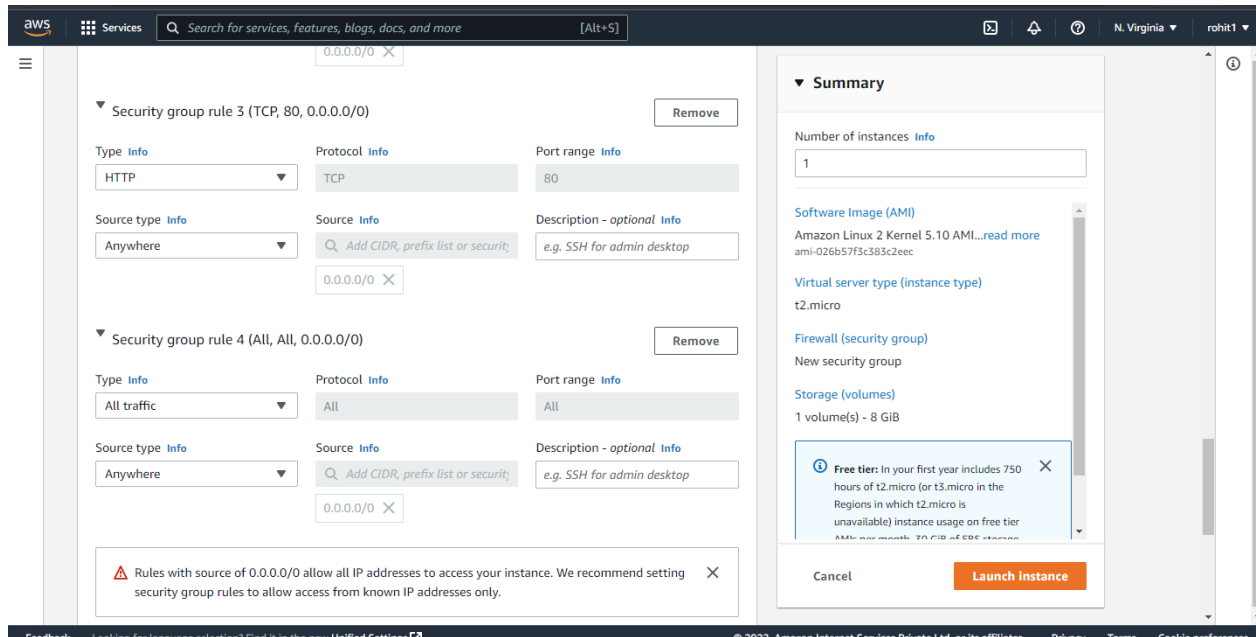
Step 2:- create key pair



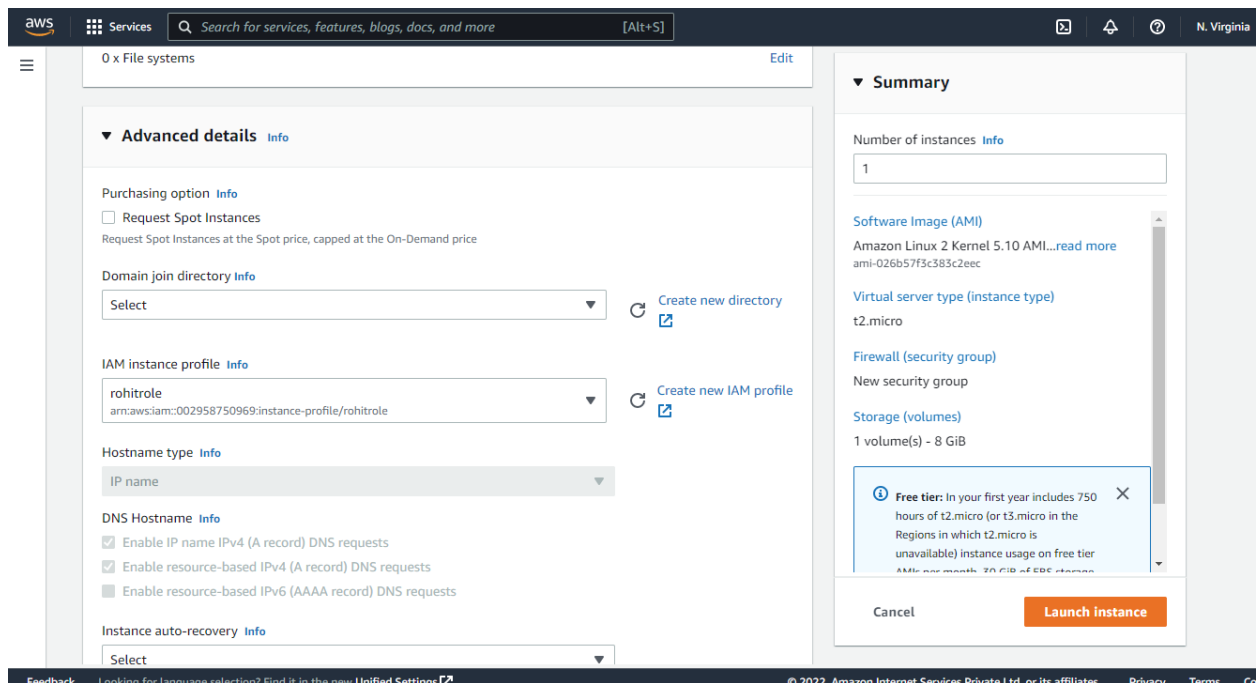
Step 3:- edit network settings add security rule

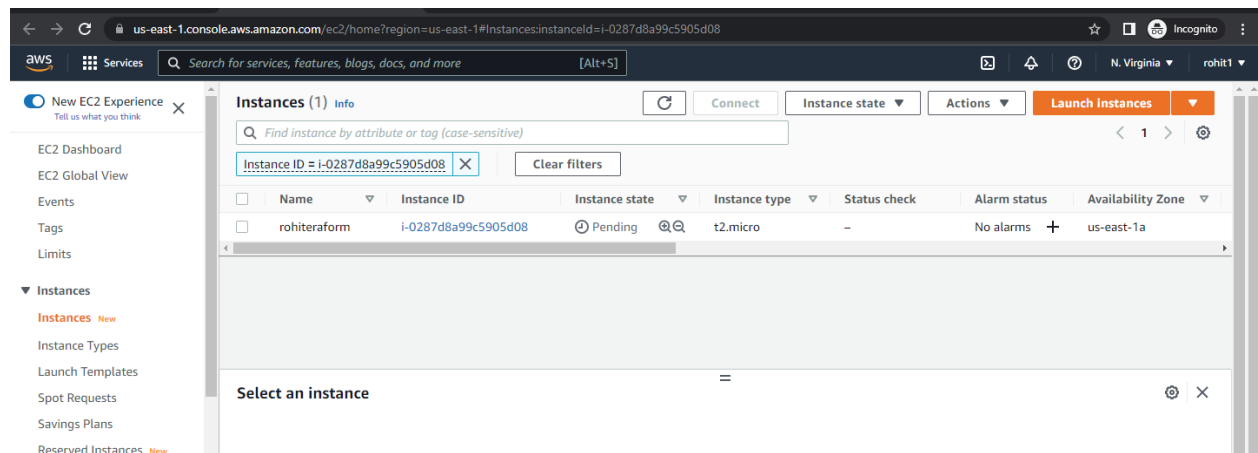
Type=all traffic

Source type =anywhere

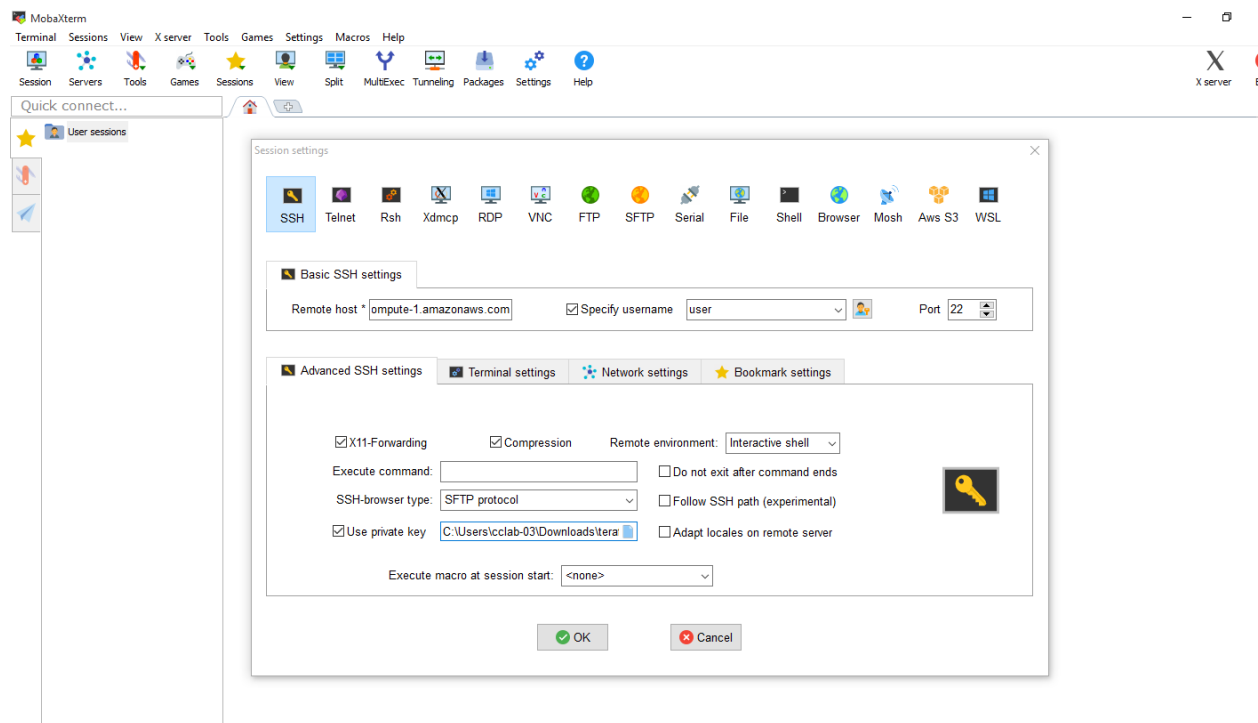


Step 4:- select the role and click on launch instance

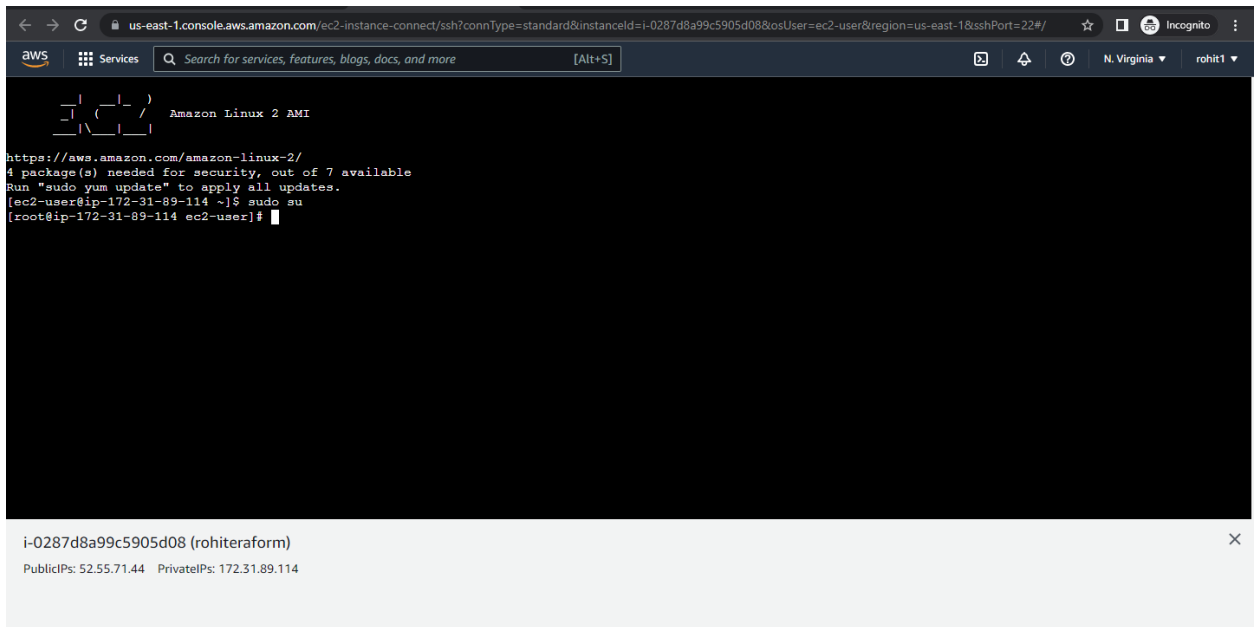




Step 5:-now go to mobaxterm and get connected to instance



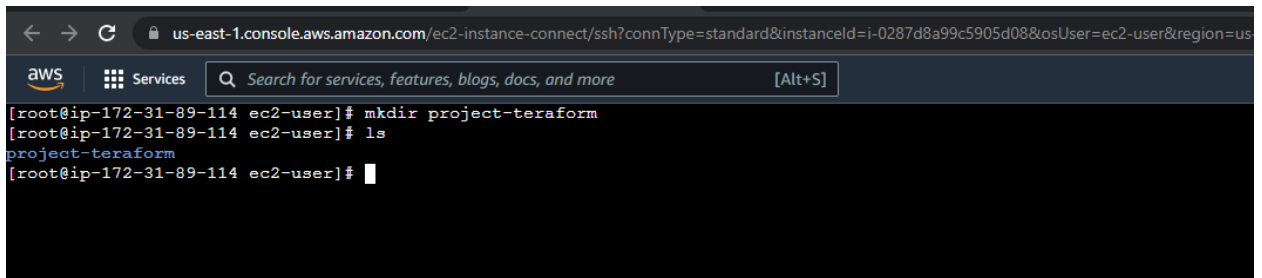
Step 6:-run sudo su to become root user



The screenshot shows a terminal window within the AWS console. The terminal displays the Amazon Linux 2 logo and a message about security updates. The user runs 'sudo su' to become the root user. The prompt changes from '[ec2-user@ip-172-31-89-114 ~]\$' to '[root@ip-172-31-89-114 ec2-user]#'. Below the terminal, the instance ID 'i-0287d8a99c5905d08 (rohiterform)' and its public/private IP addresses are listed.

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0287d8a99c5905d08&osUser=ec2-user&region=us-east-1&sshPort=22#/  
[Alt+S]  
N. Virginia rohit1  
Amazon Linux 2 AMI  
https://aws.amazon.com/amazon-linux-2/  
4 package(s) needed for security, out of 7 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-89-114 ~]$ sudo su  
[root@ip-172-31-89-114 ec2-user]#  
i-0287d8a99c5905d08 (rohiterform)  
PublicIPs: 52.55.71.44 PrivateIPs: 172.31.89.114
```

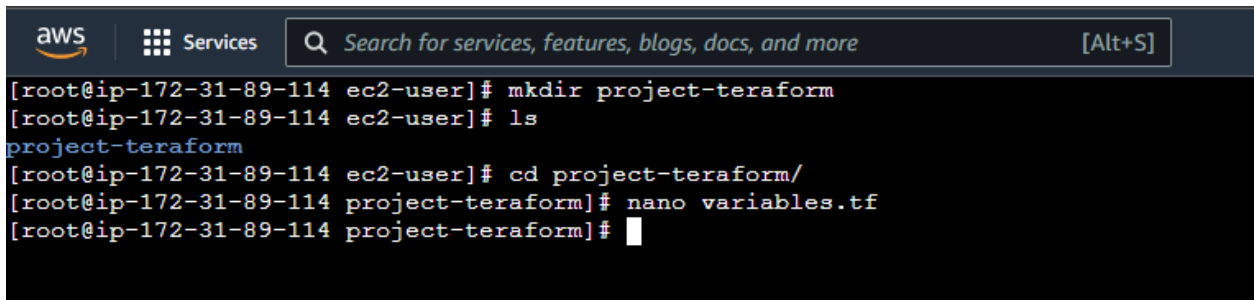
Step 7:-make directory project-teraform



The screenshot shows the terminal window where the user has become root. They run 'mkdir project-teraform' to create a new directory. Then they run 'ls' to verify its creation. The output shows 'project-teraform' as a new directory.

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0287d8a99c5905d08&osUser=ec2-user&region=us-east-1&sshPort=22#/  
[Alt+S]  
[root@ip-172-31-89-114 ec2-user]# mkdir project-teraform  
[root@ip-172-31-89-114 ec2-user]# ls  
project-teraform  
[root@ip-172-31-89-114 ec2-user]#
```

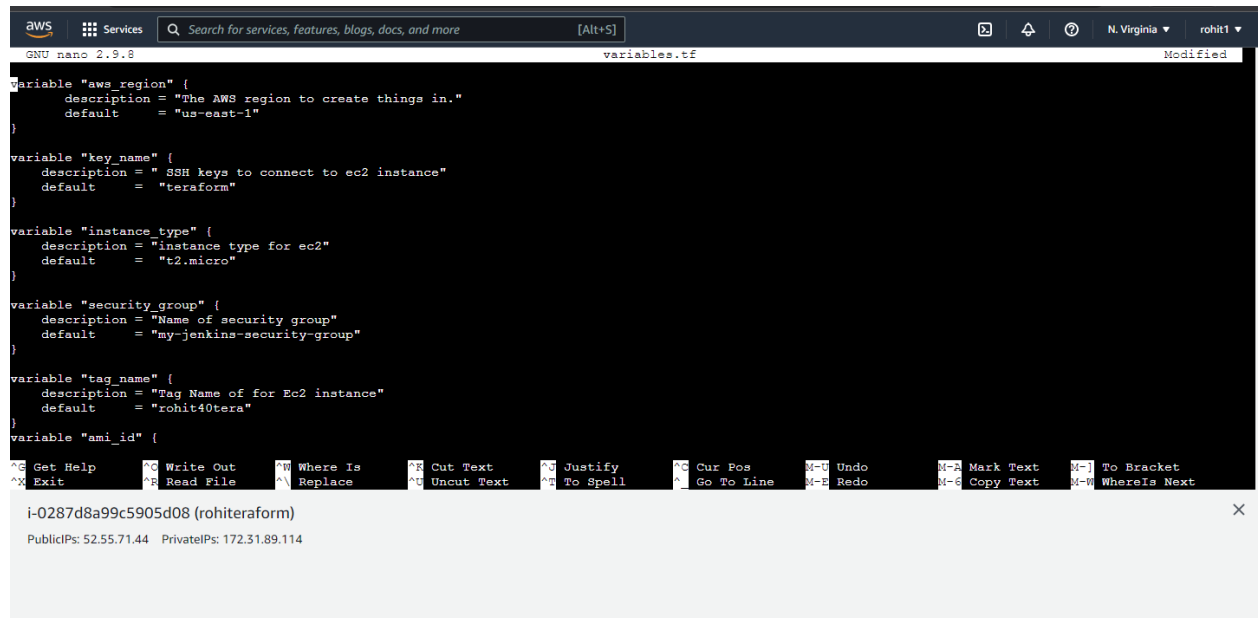
Step 8:- now create variables.tf file copy the code from classroom make changes



The screenshot shows the terminal window where the user is now in the 'project-teraform' directory. They run 'cd project-teraform/' to change the current directory. Then they run 'nano variables.tf' to create a new file named 'variables.tf'.

```
aws Services [Alt+S]  
[root@ip-172-31-89-114 ec2-user]# mkdir project-teraform  
[root@ip-172-31-89-114 ec2-user]# ls  
project-teraform  
[root@ip-172-31-89-114 ec2-user]# cd project-teraform/  
[root@ip-172-31-89-114 project-teraform]# nano variables.tf  
[root@ip-172-31-89-114 project-teraform]#
```

Edit the file



```
aws Services Search for services, features, blogs, docs, and more [Alt+S]
GNU nano 2.9.8 variables.tf Modified
variable "aws_region" {
  description = "The AWS region to create things in."
  default     = "us-east-1"
}

variable "key_name" {
  description = "SSH keys to connect to ec2 instance"
  default     = "teraform"
}

variable "instance_type" {
  description = "instance type for ec2"
  default     = "t2.micro"
}

variable "security_group" {
  description = "Name of security group"
  default     = "my-jenkins-security-group"
}

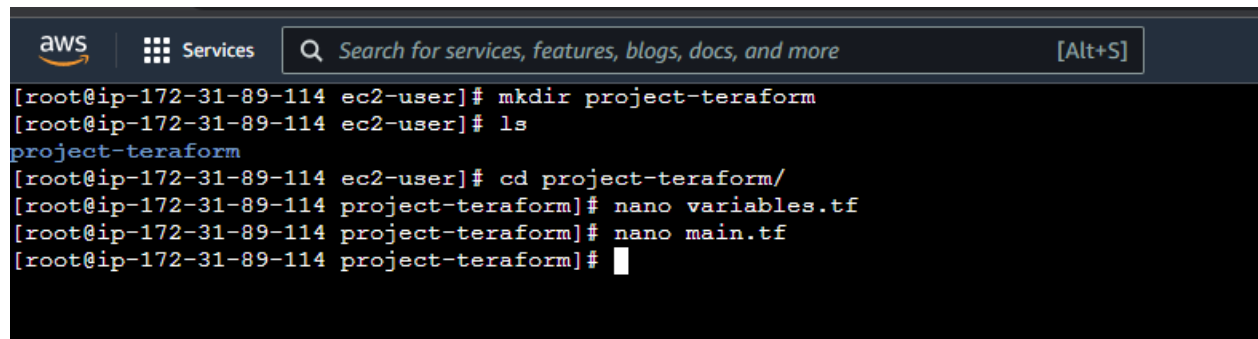
variable "tag_name" {
  description = "Tag Name of for Ec2 instance"
  default     = "rohit40tera"
}

variable "ami_id" {
  description = "AMI ID for ec2 instance"
  default     = "ami-0287d8a99c5905d08"
}

^G Get Help ^O Write Out ^W Where Is ^T Cut Text ^J Justify ^C Cur Pos ^U Undo ^M Mark Text ^I To Bracket
^X Exit ^R Read File ^\ Replace ^_ Uncut Text ^B To Spell ^_ Go To Line ^E Redo ^G Copy Text ^W WhereIs Next

i-0287d8a99c5905d08 (rohiteraform)
PublicIPs: 52.55.71.44 PrivateIPs: 172.31.89.114
```

Step 9:- create new file main.tf copy the code from classroom



```
aws Services Search for services, features, blogs, docs, and more [Alt+S]

[root@ip-172-31-89-114 ec2-user]# mkdir project-teraform
[root@ip-172-31-89-114 ec2-user]# ls
project-teraform
[root@ip-172-31-89-114 ec2-user]# cd project-teraform/
[root@ip-172-31-89-114 project-teraform]# nano variables.tf
[root@ip-172-31-89-114 project-teraform]# nano main.tf
[root@ip-172-31-89-114 project-teraform]#
```



```
aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia rohit1
GNU nano 2.9.8 main.tf Modified
provider "aws" {
  region = var.aws_region
}

#Create security group with firewall rules
resource "aws_security_group" "security_jenkins_grp" {
  name      = var.security_group
  description = "security group for jenkins"

  ingress {
    from_port = 8080
    to_port   = 8080
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port = 22
    to_port   = 22
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  # outbound from jenkins server
  egress {
    from_port = 0
    to_port   = 0
    protocol  = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

i-0287d8a99c5905d08 (rohiteraform)
PublicIPs: 52.55.71.44 PrivateIPs: 172.31.89.114

Step 10:- Install Terraform [wget]

https://releases.hashicorp.com/terraform/1.0.9/terraform_1.0.9_linux_amd64.zip

```
aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia rohit1
[root@ip-172-31-89-114 ec2-user]# mkdir project-terraform
[root@ip-172-31-89-114 ec2-user]# ls
project-terraform
[root@ip-172-31-89-114 ec2-user]# cd project-terraform/
[root@ip-172-31-89-114 project-terraform]# nano variables.tf
[root@ip-172-31-89-114 project-terraform]# nano main.tf
[root@ip-172-31-89-114 project-terraform]# wget https://releases.hashicorp.com/terraform/1.0.9/terraform_1.0.9_linux_amd64.zip
```

```
aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia rohit1
[root@ip-172-31-89-114 ec2-user]# mkdir project-terraform
[root@ip-172-31-89-114 ec2-user]# ls
project-terraform
[root@ip-172-31-89-114 ec2-user]# cd project-terraform/
[root@ip-172-31-89-114 project-terraform]# nano variables.tf
[root@ip-172-31-89-114 project-terraform]# nano main.tf
[root@ip-172-31-89-114 project-terraform]# wget https://releases.hashicorp.com/terraform/1.0.9/terraform_1.0.9_linux_amd64.zip
--2022-10-12 06:27:30-- https://releases.hashicorp.com/terraform/1.0.9/terraform_1.0.9_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 108.138.85.53, 108.138.85.65, 108.138.85.30, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|108.138.85.53|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32674820 (31M) [application/zip]
Saving to: 'terraform_1.0.9_linux_amd64.zip'

100%[=====>] 32,674,820  184MB/s  in 0.2s

2022-10-12 06:27:30 (184 MB/s) - 'terraform_1.0.9_linux_amd64.zip' saved [32674820/32674820]

[root@ip-172-31-89-114 project-terraform]#
```

```
100%[=====>] 32,674,820  184MB/s  in 0.2s
2022-10-12 06:27:30 (184 MB/s) - 'terraform_1.0.9_linux_amd64.zip' saved [32674820/32674820]
[root@ip-172-31-89-114 project-teraform]# unzip terraform_1.0.9_linux_amd64.zip
```

Step 11:-unzip that file

```
[root@ip-172-31-89-114 project-teraform]# unzip terraform_1.0.9_linux_amd64.zip
Archive:  terraform_1.0.9_linux_amd64.zip
  inflating: terraform
[root@ip-172-31-89-114 project-teraform]#
```

Step 12:- then copy the file with command cp terraform /bin/

Check version

```
[root@ip-172-31-89-114 project-teraform]# cp terraform /bin/
[root@ip-172-31-89-114 project-teraform]# terraform --version
Terraform v1.0.9
on linux_amd64

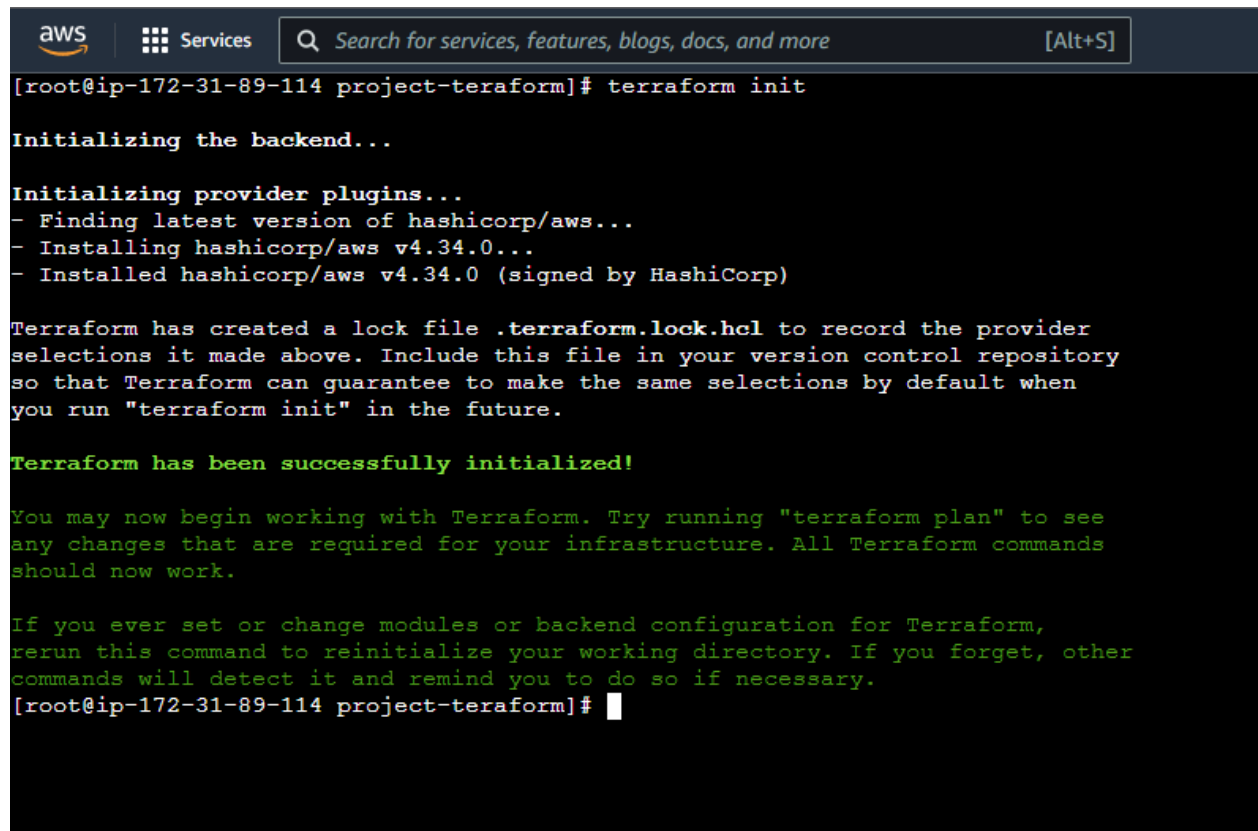
Your version of Terraform is out of date! The latest version
is 1.3.2. You can update by downloading from https://www.terraform.io/downloads.html
[root@ip-172-31-89-114 project-teraform]#
```

i-0287d8a99c5905d08 (rohiteraform)

5.Explain following Terraform commands in one line

Step 13:- then perform **terraform init** command

The terraform init command initializes a working directory containing configuration files and installs plugins for required providers.



```
[root@ip-172-31-89-114 project-teraform]# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.34.0...
- Installed hashicorp/aws v4.34.0 (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.
[root@ip-172-31-89-114 project-teraform]#
```

Step 14:- then type **terraform plan** command

The terraform plan command lets you to preview the actions Terraform would take to modify your infrastructure, or save a speculative plan which you can apply later.

```
aws Services Search for services, features, blogs, docs, and more [Alt+S]
[root@ip-172-31-89-114 project-teraform]# 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_eip.myFirstInstance will be created
+ resource "aws_eip" "myFirstInstance" {
  + allocation_id      = (known after apply)
  + association_id     = (known after apply)
  + carrier_ip         = (known after apply)
  + customer_owned_ip  = (known after apply)
  + domain             = (known after apply)
  + id                 = (known after apply)
  + instance           = (known after apply)
  + network_border_group = (known after apply)
  + network_interface  = (known after apply)
  + private_dns        = (known after apply)
  + private_ip         = (known after apply)
  + public_dns         = (known after apply)
  + public_ip          = (known after apply)
  + public_ipv4_pool    = (known after apply)
  + tags               = {
    + "Name" = "jenkins_elastic_ip"
  }
  + tags_all           = {
    + "Name" = "jenkins_elastic_ip"
  }
}
```

Step 15:- then perform **terraform apply**

The **terraform apply** command performs a plan just like **terraform plan** does, but then actually carries out the planned changes to each resource using the relevant infrastructure provider's API.

```
aws Services Search for services, features, blogs, docs, and more [Alt+S]
[root@ip-172-31-89-114 project-teraform]# 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_eip.myFirstInstance will be created
+ resource "aws_eip" "myFirstInstance" {
  + allocation_id      = (known after apply)
  + association_id     = (known after apply)
  + carrier_ip         = (known after apply)
  + customer_owned_ip  = (known after apply)
  + domain             = (known after apply)
  + id                 = (known after apply)
  + instance           = (known after apply)
  + network_border_group = (known after apply)
  + network_interface  = (known after apply)
  + private_dns        = (known after apply)
  + private_ip         = (known after apply)
  + public_dns         = (known after apply)
  + public_ip          = (known after apply)
  + public_ipv4_pool    = (known after apply)
  + tags               = {
    + "Name" = "jenkins_elastic_ip"
  }
}
```

Type “yes”

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

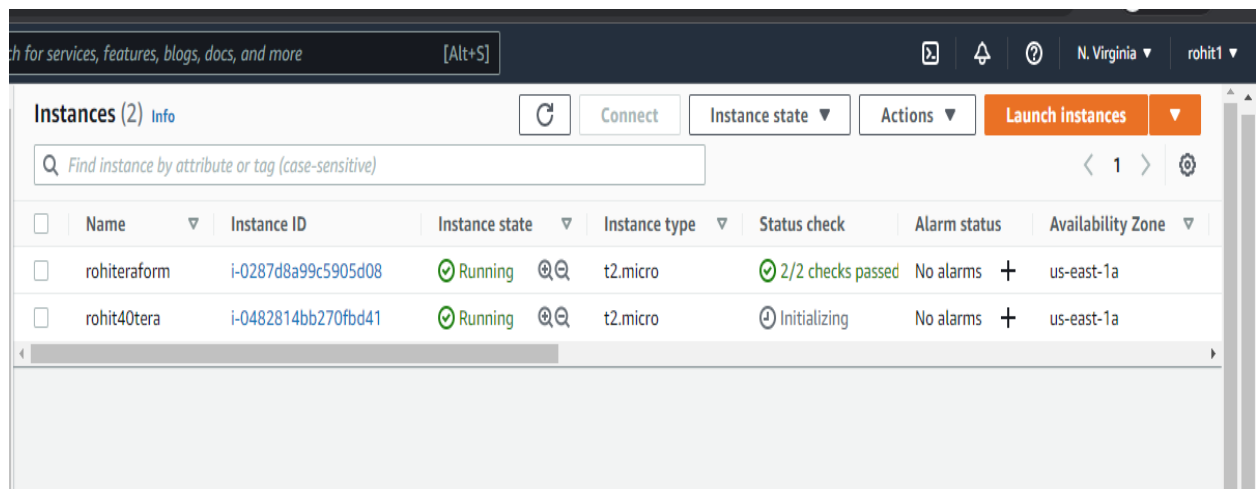
Enter a value: yes

aws_instance.myFirstInstance: Creating...
aws_security_group.security_jenkins_grp: Creating...
aws_security_group.security_jenkins_grp: Creation complete after 2s [id=sg-084385bcb8fe1f909]
aws_instance.myFirstInstance: Still creating... [10s elapsed]
aws_instance.myFirstInstance: Still creating... [20s elapsed]
aws_instance.myFirstInstance: Still creating... [30s elapsed]
aws_instance.myFirstInstance: Still creating... [40s elapsed]
aws_instance.myFirstInstance: Creation complete after 42s [id=i-0482814bb270fbd41]
aws_eip.myFirstInstance: Creating...
aws_eip.myFirstInstance: Creation complete after 2s [id=eipalloc-01155bf1a1dd68549]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
[root@ip-172-31-89-114 project-teraform]#
```

i-0287d8a99c5905d08 (rohiteraform)

New ubuntu instance is created



<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	rohiteraform	i-0287d8a99c5905d08	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a
<input type="checkbox"/>	rohit40tera	i-0482814bb270fbd41	Initializing	t2.micro	Initializing	No alarms	us-east-1a

Step 16 :- Then type terraform validate command

```
[root@ip-172-31-89-114 project-teraform]# terraform validate
Success! The configuration is valid.

[root@ip-172-31-89-114 project-teraform]#
```

i-0287d8a99c5905d08 (rohiteraform)

Step 17:- to destroy the newly created instances type **terraform destroy**

The terraform destroy command is a convenient way to destroy all remote objects managed by a particular Terraform configuration.

```
[root@ip-172-31-89-114 project-teraform]# terraform destroy
aws_instance.myFirstInstance: Refreshing state... [id=i-0482814bb270fbd41]
aws_security_group.security_jenkins_grp: Refreshing state... [id=sg-084385bc8fe1f909]
aws_eip.myFirstInstance: Refreshing state... [id=eipalloc-01155bfla1dd68549]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply":

# aws_instance.myFirstInstance has been changed
resource "aws_instance" "myFirstInstance" {
```

Type “yes”

```
Enter a value: yes

aws_security_group.security_jenkins_grp: Destroying... [id=sg-084385bc8fe1f909]
aws_eip.myFirstInstance: Destroying... [id=eipalloc-01155bfla1dd68549]
aws_eip.myFirstInstance: Destruction complete after 3s
aws_instance.myFirstInstance: Destroying... [id=i-0482814bb270fbd41]
aws_security_group.security_jenkins_grp: Still destroying... [id=sg-084385bc8fe1f909, 10s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0482814bb270fbd41, 10s elapsed]
aws_security_group.security_jenkins_grp: Still destroying... [id=sg-084385bc8fe1f909, 20s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0482814bb270fbd41, 20s elapsed]
aws_security_group.security_jenkins_grp: Still destroying... [id=sg-084385bc8fe1f909, 30s elapsed]
aws_instance.myFirstInstance: Destruction complete after 30s
aws_security_group.security_jenkins_grp: Destruction complete after 39s

Destroy complete! Resources: 3 destroyed.
[root@ip-172-31-89-114 project-teraform]#
```

The instance was deleted.

Instances (2) Info								
<input type="text" value="Find instance by attribute or tag (case-sensitive)"/>								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	
<input type="checkbox"/>	rohiteraform	i-0287d8a99c5905d08	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	
<input type="checkbox"/>	rohit40tera	i-0482814bb270fbd41	Terminated	t2.micro	-	No alarms	us-east-1a	