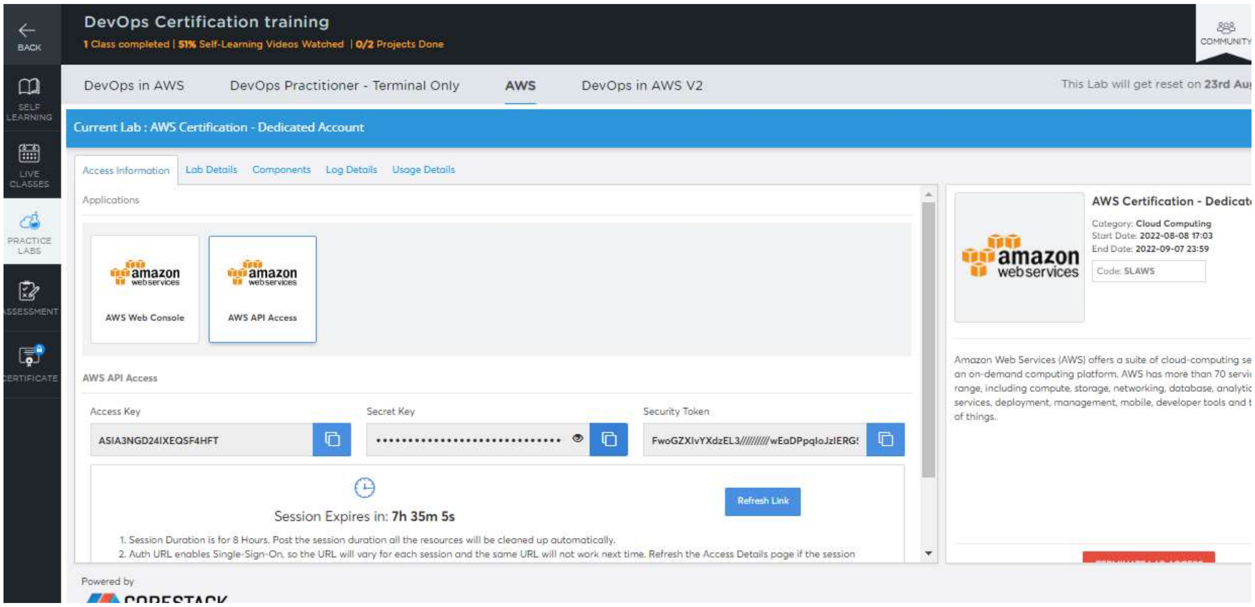
Project: Automating Infrastructure using Terraform



DevOps Certification training

1 Class completed | 51% Self-Learning Videos Watched | 0/2 Projects Done

This Lab will get reset on 23rd Au

Current Lab : AWS Certification - Dedicated Account

AWS API Access

Access Key: ASIA3NGD24IXEQSF4HFT

Secret Key: ••••••••••••••••••••••••••••••

Security Token: FwoGZXIvYXdzEL3//////////wEaDPpqloJzlERG!

Session Expires in: **7h 35m 5s**

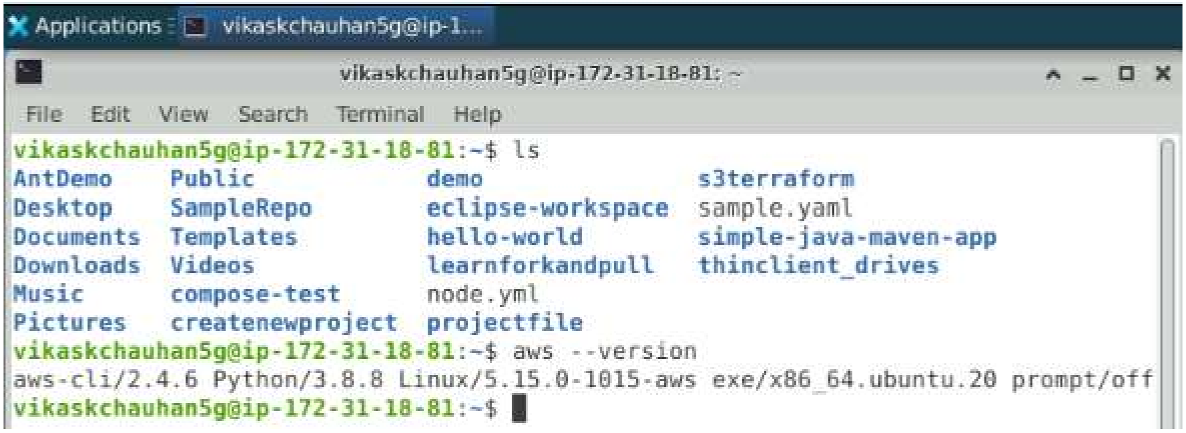
1. Session Duration is for 8 Hours. Post the session duration all the resources will be cleaned up automatically.
2. Auth URL enables Single-Sign-On, so the URL will vary for each session and the same URL will not work next time. Refresh the Access Details page if the session

AWS Certification - Dedicat

Category: **Cloud Computing**
Start Date: **2022-08-08 17:03**
End Date: **2022-09-07 23:59**
Code: **SLAWS**

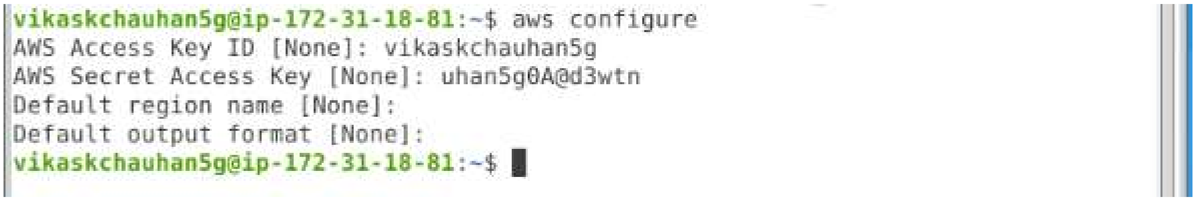
Amazon Web Services (AWS) offers a suite of cloud-computing se an on-demand computing platform. AWS has more than 70 servi range, including compute, storage, networking, database, analytic services, deployment, management, mobile, developer tools and t of things.

AWL CLI verification:



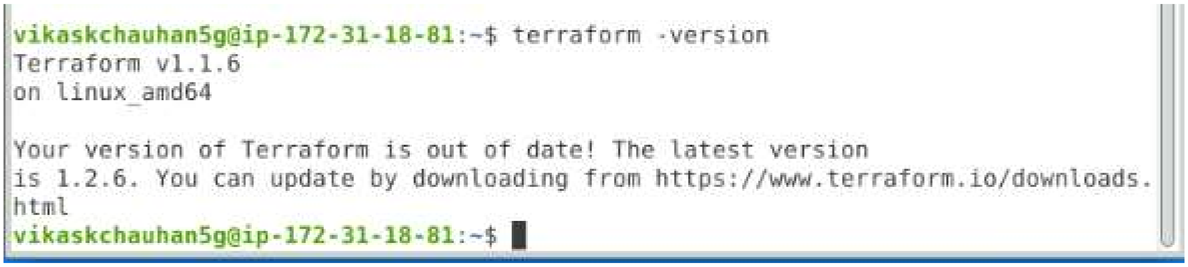
```
vikaskchauhan5g@ip-172-31-18-81:~$ ls
AntDemo     Public            demo                s3terraform
Desktop     SampleRepo        eclipse-workspace   sample.yaml
Documents   Templates         hello-world         simple-java-maven-app
Downloads   Videos            learnforkandpull    thinclient_drives
Music       compose-test      node.yml
Pictures    createnewproject  projectfile
vikaskchauhan5g@ip-172-31-18-81:~$ aws --version
aws-cli/2.4.6 Python/3.8.8 Linux/5.15.0-1015-aws exe/x86_64.ubuntu.20 prompt/off
vikaskchauhan5g@ip-172-31-18-81:~$ 
```

Configuring AWS:



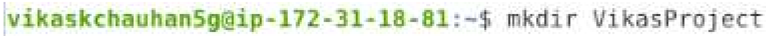
```
vikaskchauhan5g@ip-172-31-18-81:~$ aws configure
AWS Access Key ID [None]: vikaskchauhan5g
AWS Secret Access Key [None]: uhan5g0A@d3wtn
Default region name [None]:
Default output format [None]:
vikaskchauhan5g@ip-172-31-18-81:~$ 
```

Setting up Terraform:



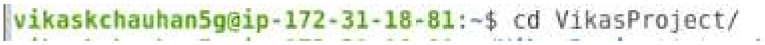
```
vikaskchauhan5g@ip-172-31-18-81:~$ terraform -version
Terraform v1.1.6
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.2.6. You can update by downloading from https://www.terraform.io/downloads.
html
vikaskchauhan5g@ip-172-31-18-81:~$ 
```

Make a directory:



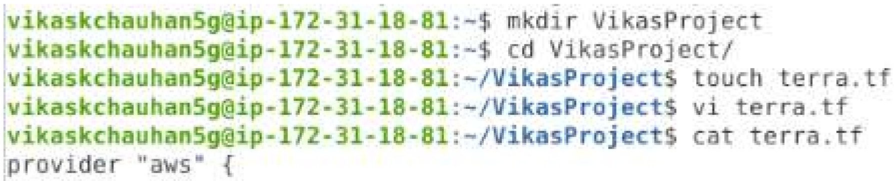
```
vikaskchauhan5g@ip-172-31-18-81:~$ mkdir VikasProject
```

Inside a directory:

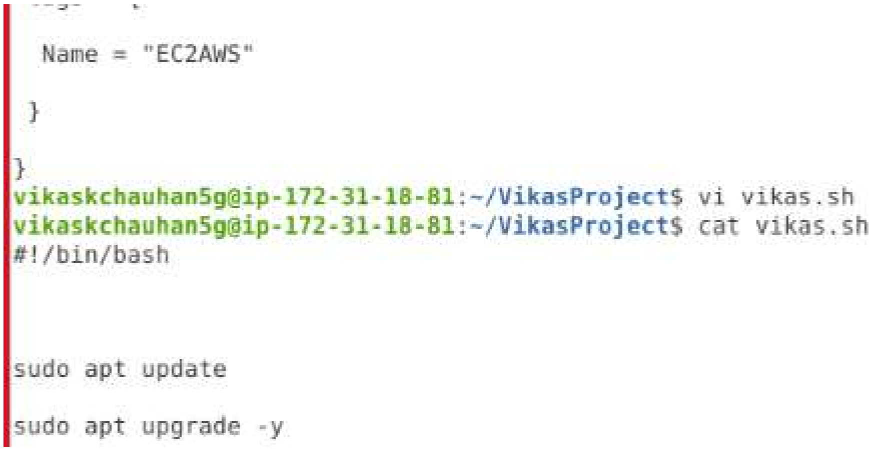


```
vikaskchauhan5g@ip-172-31-18-81:~$ cd VikasProject/
```

Make a File:



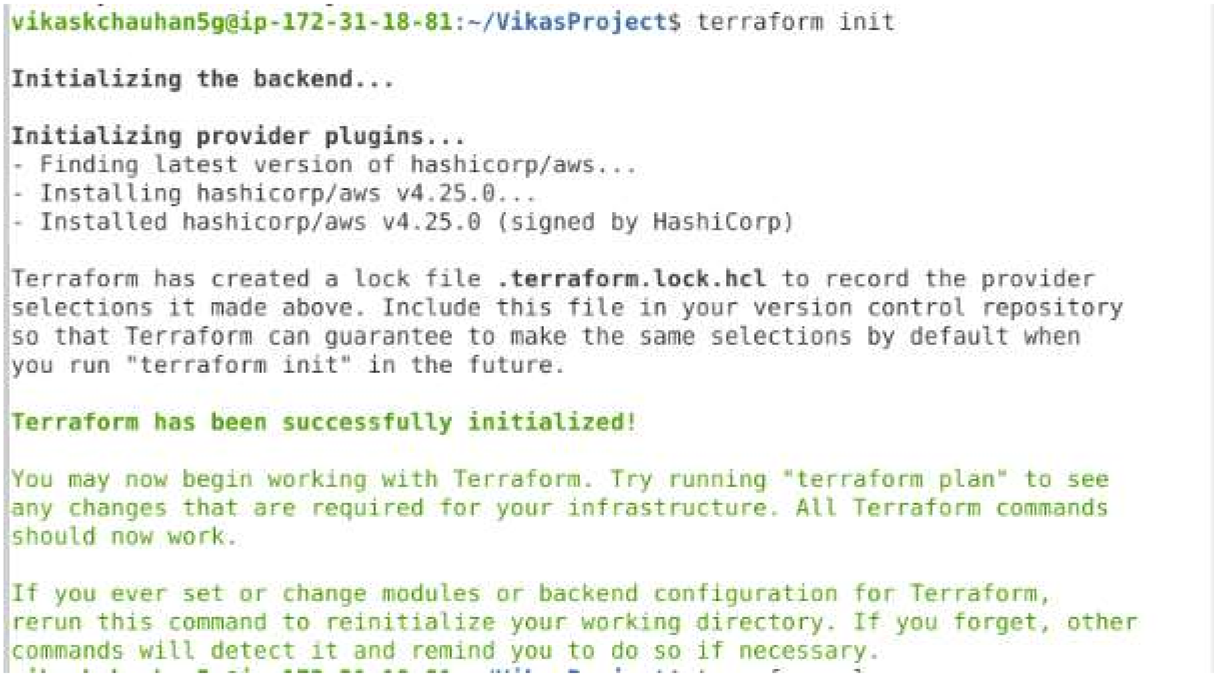
```
vikaskchauhan5g@ip-172-31-18-81:~$ mkdir VikasProject
vikaskchauhan5g@ip-172-31-18-81:~$ cd VikasProject/
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$ touch terra.tf
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$ vi terra.tf
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$ cat terra.tf
provider "aws" {
```



```
  Name = "EC2AWS"

 }

}
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$ vi vikas.sh
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$ cat vikas.sh
#!/bin/bash



sudo apt update

sudo apt upgrade -y
```

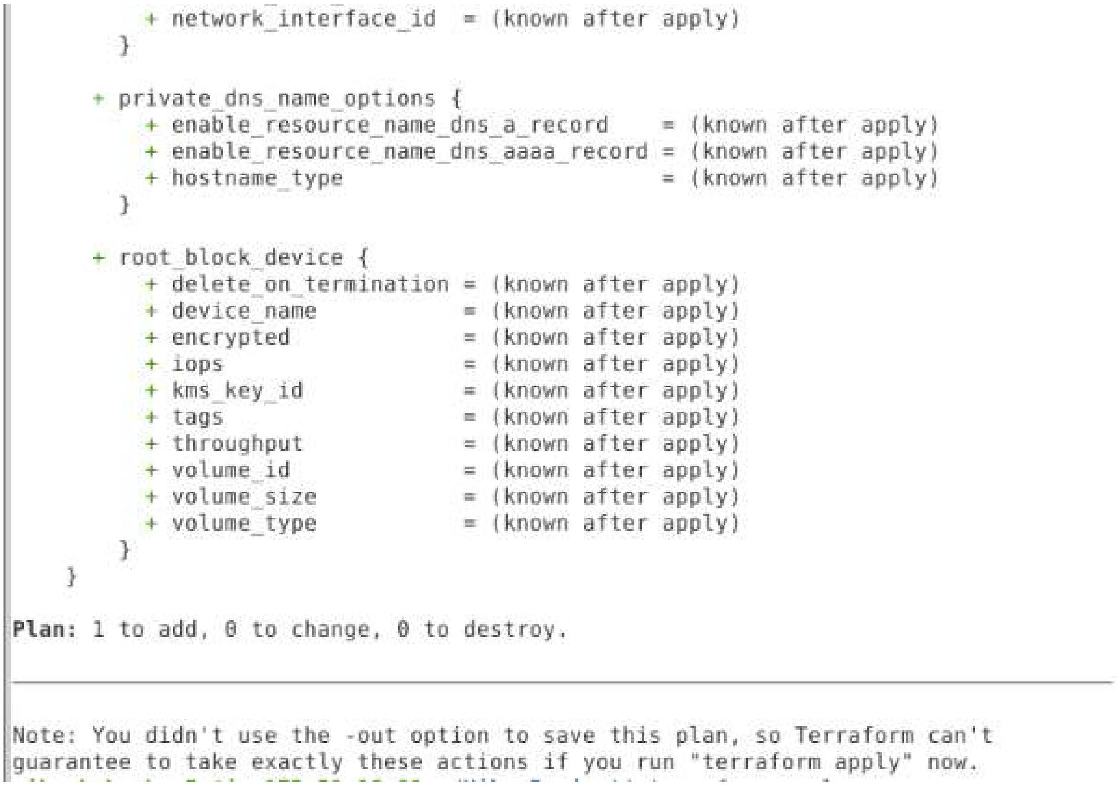
Setting up terraform infrastructure:

# terraform init



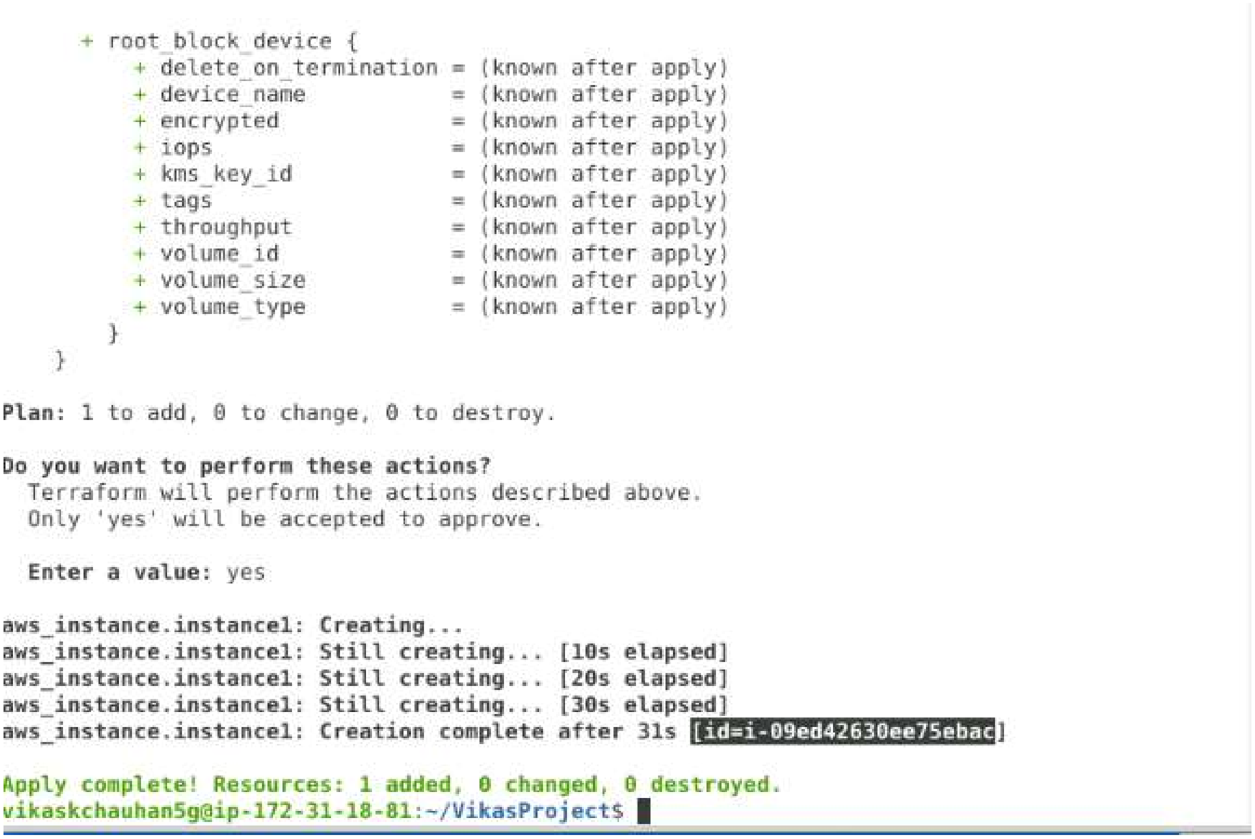
```
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.25.0...
- Installed hashicorp/aws v4.25.0 (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.
```

# terraform plan



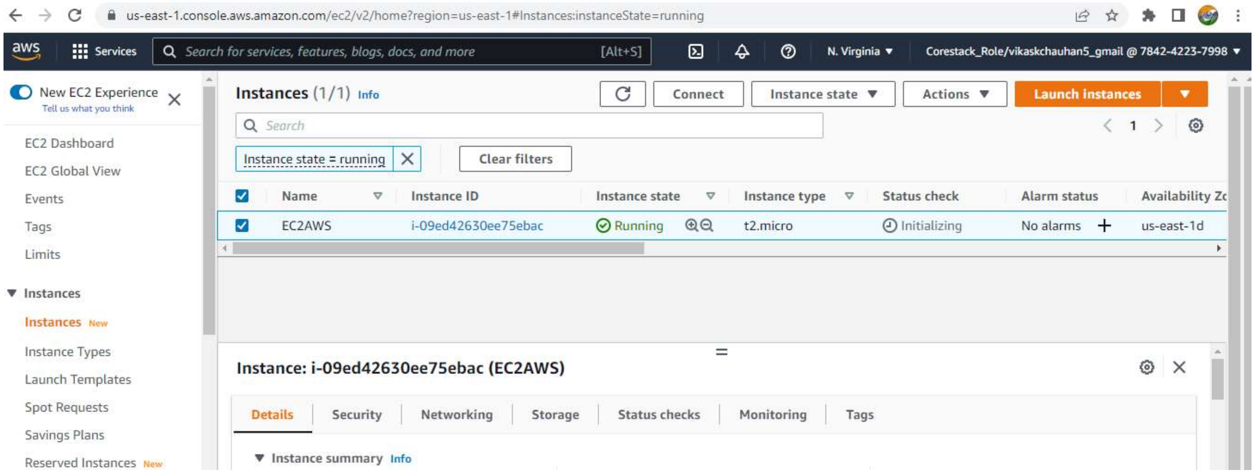
```
          + network_interface_id  = (known after apply)
        }

      + private_dns_name_options {
          + enable_resource_name_dns_a_record    = (known after apply)
          + enable_resource_name_dns_aaaa_record = (known after apply)
          + hostname_type                        = (known after apply)
        }

      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

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

───────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't
guarantee to take exactly these actions if you run "terraform apply" now.
```

# terraform apply



```
      + root_block_device {
          + delete_on_termination = (known after apply)
          + device_name           = (known after apply)
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + tags                  = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = (known after apply)
          + volume_type           = (known after apply)
        }
    }

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: yes

aws_instance.instance1: Creating...
aws_instance.instance1: Still creating... [10s elapsed]
aws_instance.instance1: Still creating... [20s elapsed]
aws_instance.instance1: Still creating... [30s elapsed]
aws_instance.instance1: Creation complete after 31s [id=i-09ed42630ee75ebac]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$ 
```

Running EC2 Instance on web console:



us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:instanceState=running

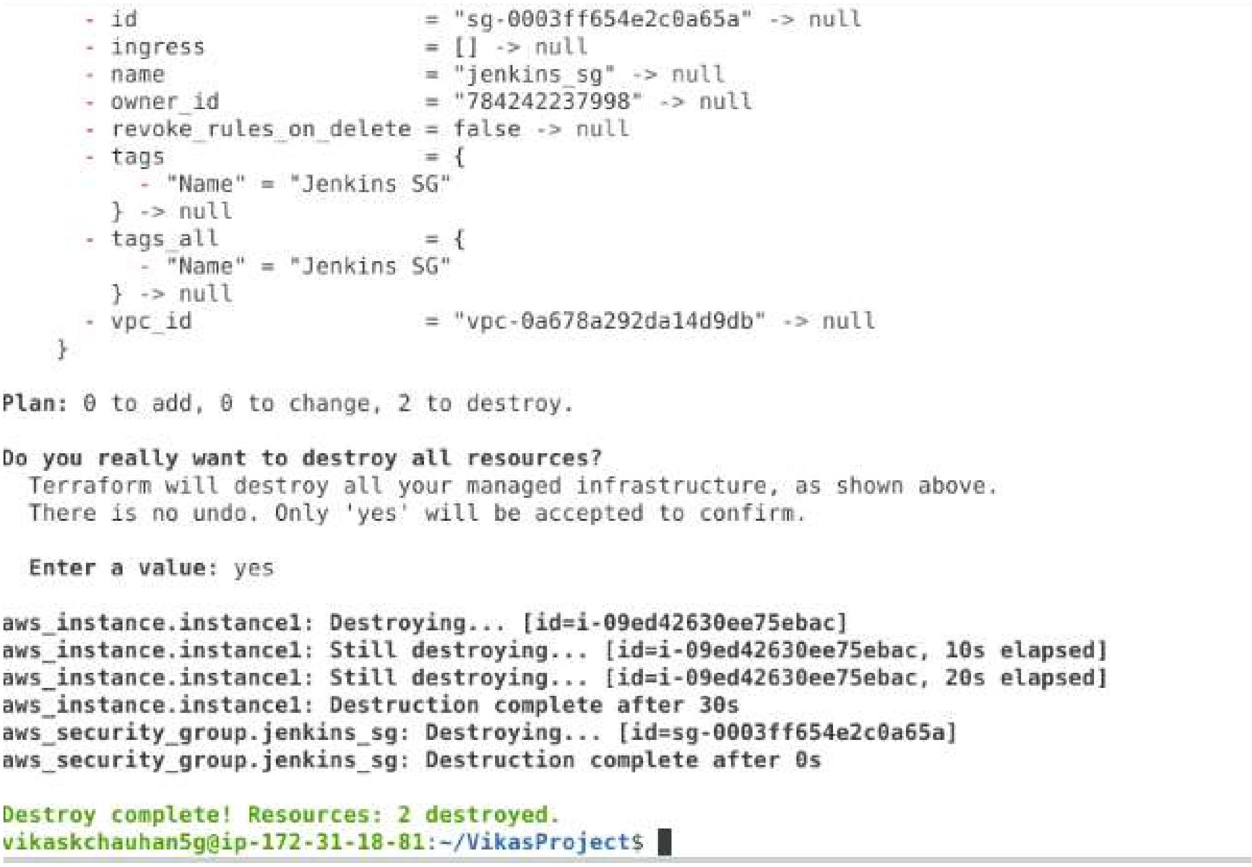
Instances (1/1)

Instance state = running

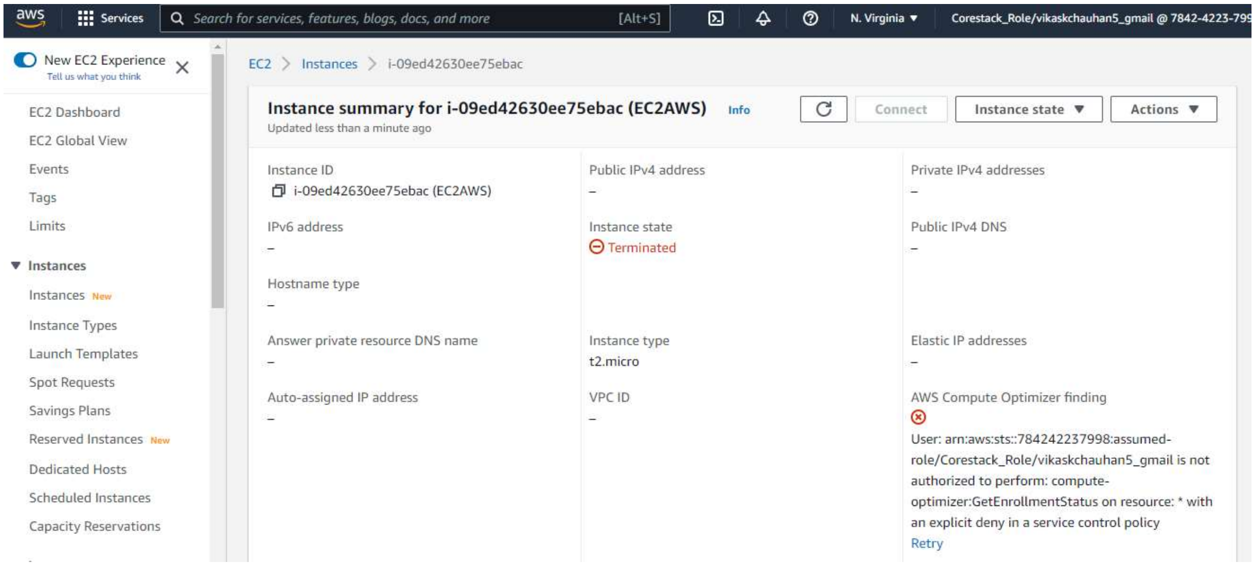
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zc
EC2AWS	i-09ed42630ee75ebac	Running	t2.micro	Initializing	No alarms	us-east-1d

Instance: i-09ed42630ee75ebac (EC2AWS)

Destroy the instance:



```
      - id                     = "sg-0003ff654e2c0a65a" -> null
      - ingress                = [] -> null
      - name                   = "jenkins_sg" -> null
      - owner_id               = "784242237998" -> null
      - revoke_rules_on_delete = false -> null
      - tags                   = {
          - "Name" = "Jenkins SG"
        } -> null
      - tags_all               = {
          - "Name" = "Jenkins SG"
        } -> null
      - vpc_id                 = "vpc-0a678a292da14d9db" -> null
    }

Plan: 0 to add, 0 to change, 2 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

aws_instance.instance1: Destroying... [id=i-09ed42630ee75ebac]
aws_instance.instance1: Still destroying... [id=i-09ed42630ee75ebac, 10s elapsed]
aws_instance.instance1: Still destroying... [id=i-09ed42630ee75ebac, 20s elapsed]
aws_instance.instance1: Destruction complete after 30s
aws_security_group.jenkins_sg: Destroying... [id=sg-0003ff654e2c0a65a]
aws_security_group.jenkins_sg: Destruction complete after 0s

Destroy complete! Resources: 2 destroyed.
vikaskchauhan5g@ip-172-31-18-81:~/VikasProject$
```



Instance summary for i-09ed42630ee75ebac (EC2AWS) Info

Updated less than a minute ago

Instance ID: i-09ed42630ee75ebac (EC2AWS)

Public IPv4 address: –

Private IPv4 addresses: –

IPv6 address: –

Instance state: ⊖ Terminated

Public IPv4 DNS: –

Hostname type: –

Answer private resource DNS name: –

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: –

VPC ID: –

AWS Compute Optimizer finding: ⊗ User: arn:aws:sts::784242237998:assumed-role/Corestack_Role/vikaskchauhan5_gmail is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * with an explicit deny in a service control policy

Retry