Plesae check the GitHub repo for better readabilty

https://github.com/ibrahimmenshawy94/PG-DevOps-Project---Hotel-Side-Hospital.git

Introduction:

Hotel-Side Hospital, a globally renowned hospital chain headquartered in Australia, is aiming to streamline its operation by setting up an infrastructure within the hotel premises. However, in order to maintain seamless functioning and scalability, they require fully managed virtual machines (VMs) on the Amazon Web Services (AWS) platform. The organization seeks an automated provisioned infrastructure solution that can enable them to effortlessly create new Amazon Elastic Kubernetes Service (EKS) clusters, whenever required, and promptly delete them when they are no longer needed. This will optimize resource allocation and enhance operational efficiency.

Prerequisites:

Skills used in the project and their usage in the industry are as below:

• Terraform: It is an infrastructure-as-code tool that allows you to define and provision resources in a cloud environment. In this project, Terraform is used to define the infrastructure components and manage their lifecycle.

• EKS: It is a managed Kubernetes service provided by AWS. In this project, an EKS cluster is created using Terraform, which provides a scalable and highly available environment for running containerized workloads.

• EC2: EC2 instances are virtual servers provided by AWS. In this project, EC2 instances are provisioned using Terraform, which allows you to specify the desired instance types, operating systems, and other configurations.

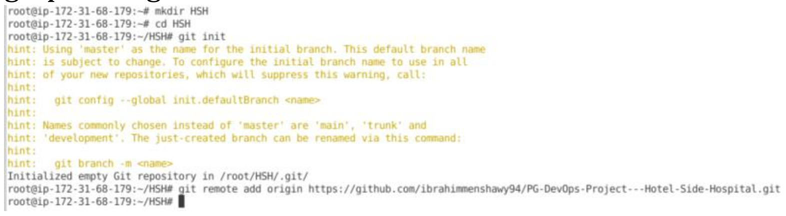
Repository Setup:

- Steps to initialize the Git repository.

mkdir HSH

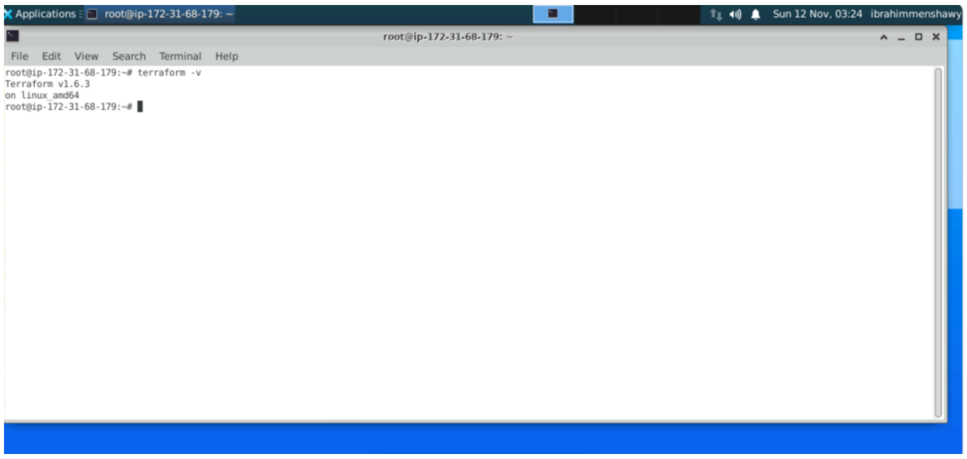
git init
git remote add origin https://github.com/ibrahimmenshawy94/PG-DevOps-Project---Hotel-Side-Hospital.git

git pull origin main



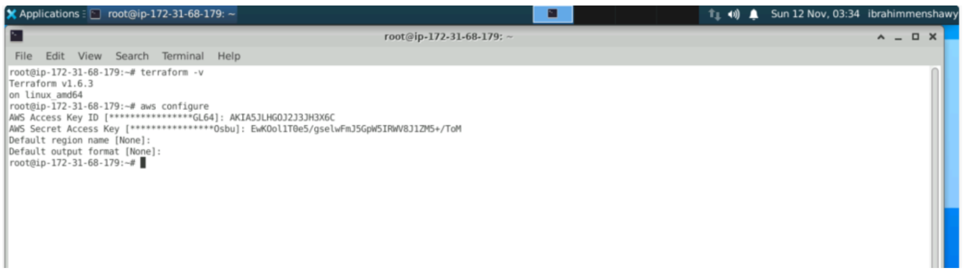
```
root@ip-172-31-68-179:~# mkdir HSH
root@ip-172-31-68-179:~# cd HSH
root@ip-172-31-68-179:~/HSH# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/HSH/.git/
root@ip-172-31-68-179:~/HSH# git remote add origin https://github.com/ibrahimmenshawy94/PG-DevOps-Project---Hotel-Side-Hospital.git
root@ip-172-31-68-179:~/HSH# 
```

Terraform Configuration:



```
root@ip-172-31-68-179:~# terraform -v
Terraform v1.6.3
on linux_amd64
root@ip-172-31-68-179:~# 
```

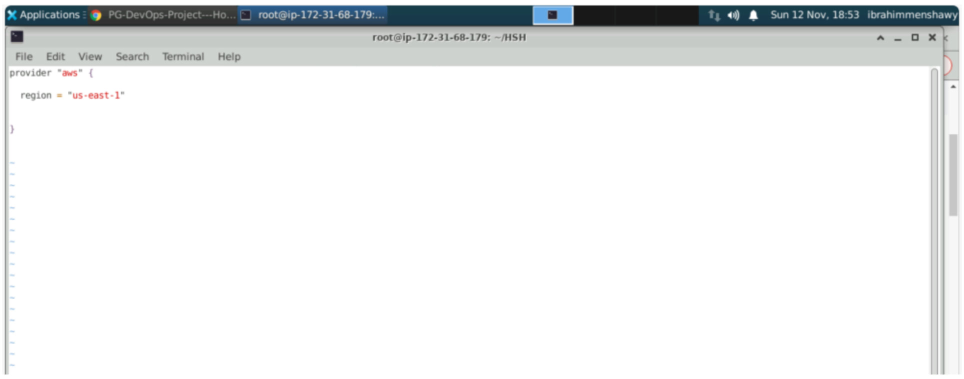
AWS CLI



```
root@ip-172-31-68-179:~# terraform -v
Terraform v1.6.3
on linux_amd64
root@ip-172-31-68-179:~# aws configure
AWS Access Key ID [****************GL64]: AKIA5JLHGOJ2J3JH3X6C
AWS Secret Access Key [****************Osbu]: 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
Default region name [None]:
Default output format [None]:
root@ip-172-31-68-179:~# 
```

Terraform files:

1- **provider.tf**



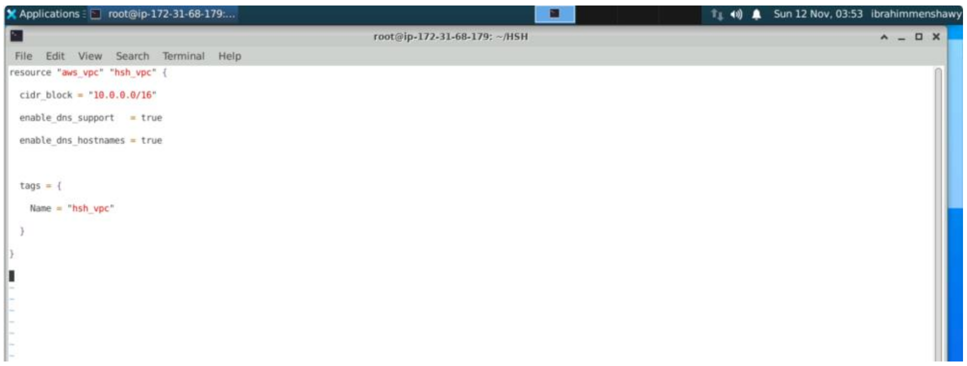
```
provider "aws" {
  region = "us-east-1"

}
```

provider "aws" {

  region = "us-east-1"

}

2 -



```
resource "aws_vpc" "hsh_vpc" {
  cidr_block = "10.0.0.0/16"
  enable_dns_support   = true
  enable_dns_hostnames = true

  tags = {
    Name = "hsh_vpc"
  }
}
```

resource "aws_vpc" "hsh_vpc" {

  cidr_block = "10.0.0.0/16"

  enable_dns_support   = true

  enable_dns_hostnames = true

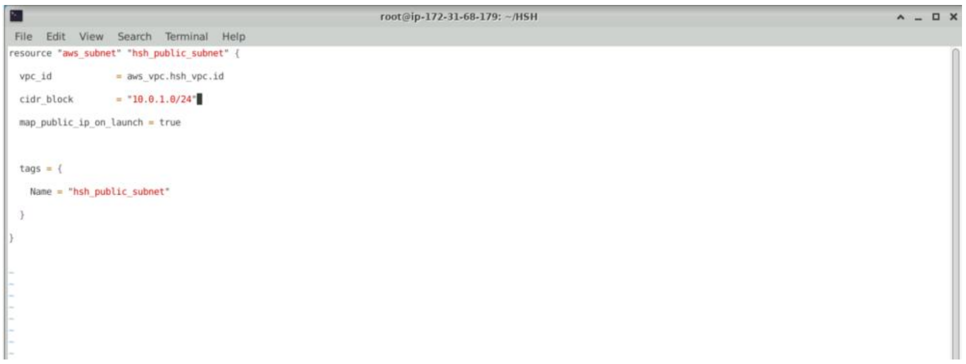
  tags = {

    Name = "hsh_vpc"

  }

}

3 - subnets.tf



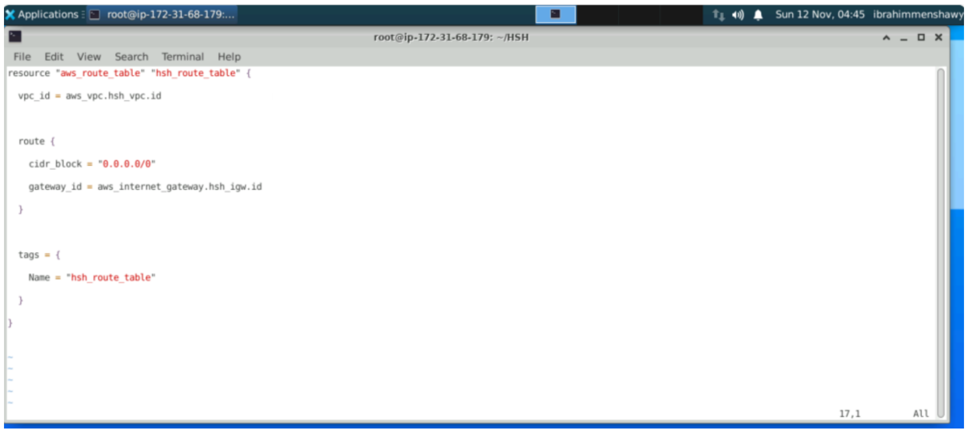
```
root@ip-172-31-68-179: ~/HSH
File  Edit  View  Search  Terminal  Help
resource "aws_subnet" "hsh_public_subnet" {
  vpc_id            = aws_vpc.hsh_vpc.id
  cidr_block        = "10.0.1.0/24"
  map_public_ip_on_launch = true

  tags = {
    Name = "hsh_public_subnet"
  }
}
```

esource "aws_subnet" "hsh_public_subnet" {

  vpc_id   = aws_vpc.hsh_vpc.id

```
  cidr_block        = "10.0.1.0/24"

  map_public_ip_on_launch = true



  tags = {

    Name = "hsh_public_subnet"

  }

}
```

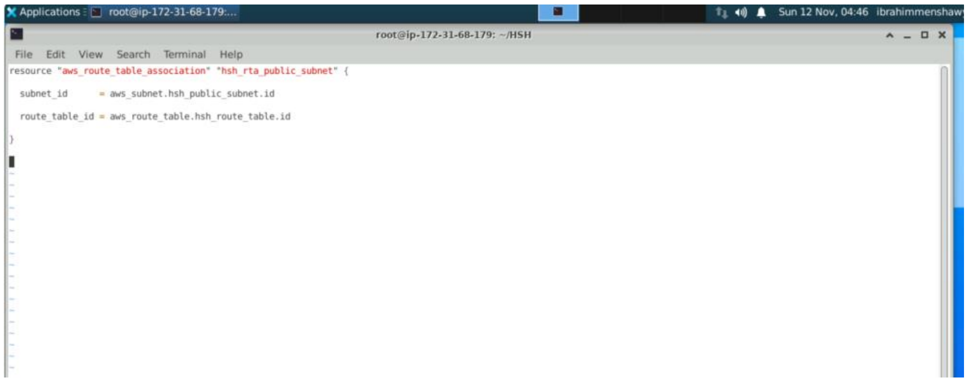
4 – routetable.tf



```
resource "aws_route_table" "hsh_route_table" {

  vpc_id = aws_vpc.hsh_vpc.id


  route {

    cidr_block = "0.0.0.0/0"

    gateway_id = aws_internet_gateway.hsh_igw.id

  }


  tags = {

    Name = "hsh_route_table"

  }

}
~
~
~
~
~
                                                   17,1          All
```

```
resource "aws_route_table" "hsh_route_table" {

  vpc_id = aws_vpc.hsh_vpc.id
  route {

    cidr_block = "0.0.0.0/0"

    gateway_id = aws_internet_gateway.hsh_igw.id

  }



  tags = {

    Name = "hsh_route_table"

  }

}
```

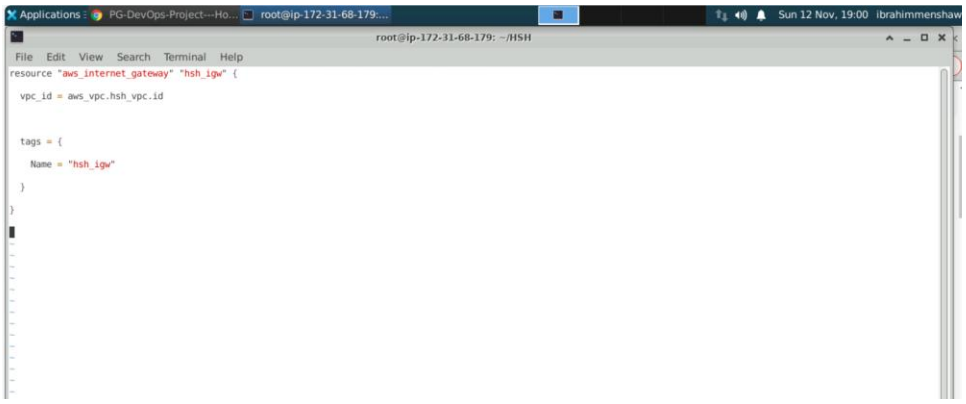
5 - Route_table_association_with_public_subnets.tf


Applications
root@ip-172-31-68-179:...
Sun 12 Nov, 04:46 ibrahimmenshaw
root@ip-172-31-68-179: ~/HSH
File Edit View Search Terminal Help

```
resource "aws_route_table_association" "hsh_rta_public_subnet" {

  subnet_id      = aws_subnet.hsh_public_subnet.id

  route_table_id = aws_route_table.hsh_route_table.id

}
```

```
resource "aws_route_table_association" "hsh_rta_public_subnet" {

  subnet_id      = aws_subnet.hsh_public_subnet.id

  route_table_id = aws_route_table.hsh_route_table.id

}
```

6 - internetgateway.tf

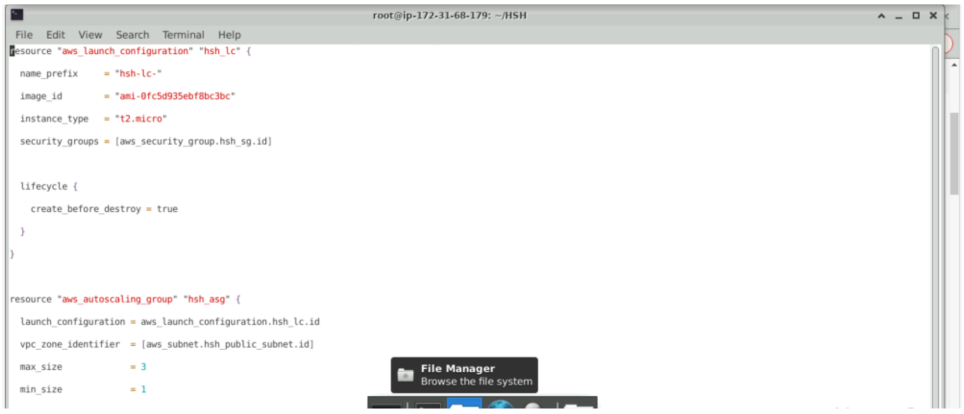

Applications
PG-DevOps-Project---Ho...
root@ip-172-31-68-179:...
Sun 12 Nov, 19:00 ibrahimmenshaw
root@ip-172-31-68-179: ~/HSH
File Edit View Search Terminal Help

```
resource "aws_internet_gateway" "hsh_igw" {
  vpc_id = aws_vpc.hsh_vpc.id


  tags = {
    Name = "hsh_igw"
  }
}
~
~
~
```

```
resource "aws_internet_gateway" "hsh_igw" {

  vpc_id = aws_vpc.hsh_vpc.id



  tags = {

    Name = "hsh_igw"

  }
```

}

7 - autoscaling.tf


root@ip-172-31-68-179: ~/HSH
File Edit View Search Terminal Help
File Manager
Browse the file system

```
resource "aws_launch_configuration" "hsh_lc" {
  name_prefix     = "hsh-lc-"
  image_id        = "ami-0fc5d935ebf8bc3bc"
  instance_type   = "t2.micro"
  security_groups = [aws_security_group.hsh_sg.id]

  lifecycle {
    create_before_destroy = true
  }
}

resource "aws_autoscaling_group" "hsh_asg" {
  launch_configuration = aws_launch_configuration.hsh_lc.id
  vpc_zone_identifier  = [aws_subnet.hsh_public_subnet.id]
  max_size             = 3
  min_size             = 1
```

```
resource "aws_launch_configuration" "hsh_lc" {

  name_prefix     = "hsh-lc-"

  image_id        = "ami-0fc5d935ebf8bc3bc"

  instance_type   = "t2.micro"

  security_groups = [aws_security_group.hsh_sg.id]



  lifecycle {

    create_before_destroy = true
```

```
 }

}



resource "aws_autoscaling_group" "hsh_asg" {

  launch_configuration = aws_launch_configuration.hsh_lc.id

  vpc_zone_identifier  = [aws_subnet.hsh_public_subnet.id]

  max_size             = 3

  min_size             = 1

  desired_capacity     = 1



  tag {

    key                 = "Name"

    value               = "hsh-instance"

    propagate_at_launch = true
  }
}


resource "aws_autoscaling_policy" "hsh_target_tracking" {
  name                   = "hsh-cpu-target-tracking"
  autoscaling_group_name = aws_autoscaling_group.hsh_asg.name
  policy_type            = "TargetTrackingScaling"
  estimated_instance_warmup = 200


  target_tracking_configuration {
    predefined_metric_specification {
      predefined_metric_type = "ASGAverageCPUUtilization"
    }
    target_value = 30.0
  }
```

}

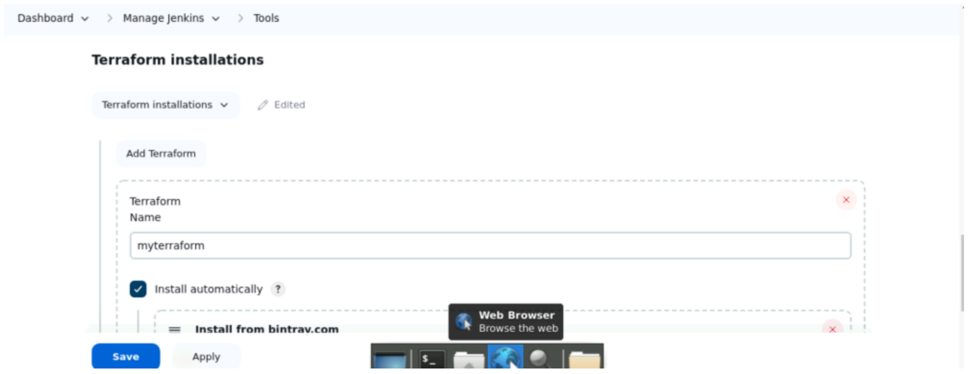
git add .

git commit -m "Initial commit or a description of changes"

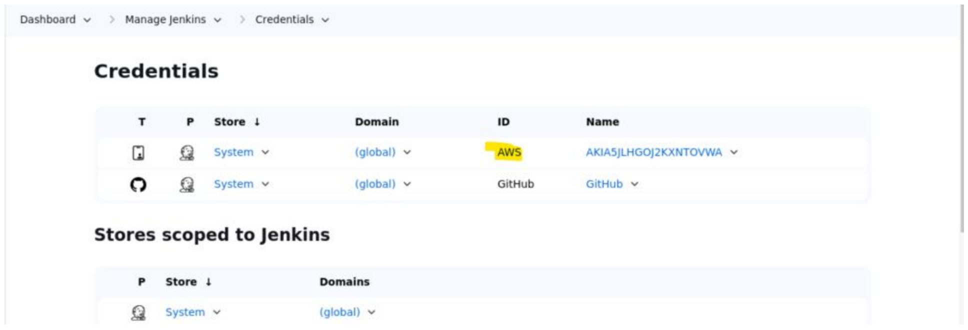
git push -u origin main

Jenkins congiuration:

1- Add terraform plugin: myterraform



1- Create a credentials for AWS IAM



Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	AWS	AKIA5JLHGOJ2KXNTOVWA
		System	(global)	GitHub	GitHub

Stores scoped to Jenkins

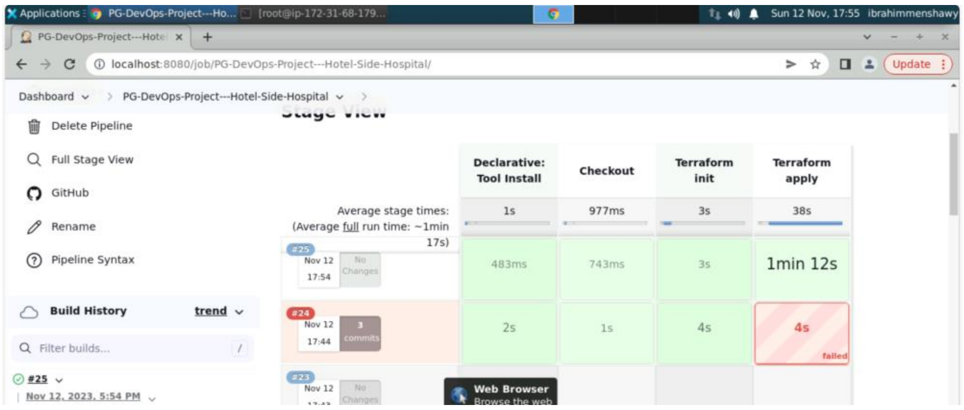
P	Store ↓	Domains
	System	(global)

- Create a new pipeline that checkout the github repo and apply the terraform scripts:

```
pipeline {
  agent any
  tools {
    terraform 'myterraform'
  }

  stages {
    stage('Checkout') {
      steps {
        checkout([$class: 'GitSCM', branches: [[name: '*/main']], extensions: [], userRemoteConfigs:
[[url: 'https://github.com/ibrahimmenshawy94/PG-DevOps-Project---Hotel-Side-Hospital.git']]])
      }
    }
    stage('Terraform init') {
      steps {
        sh 'terraform init'
      }
    }
    stage('Terraform apply') {
      steps {
        withCredentials([[
          $class: 'AmazonWebServicesCredentialsBinding',
          credentialsId: 'AWS'
        ]]) {
          sh '''
            export AWS_ACCESS_KEY_ID=$AWS_ACCESS_KEY_ID
            export AWS_SECRET_ACCESS_KEY=$AWS_SECRET_ACCESS_KEY
```

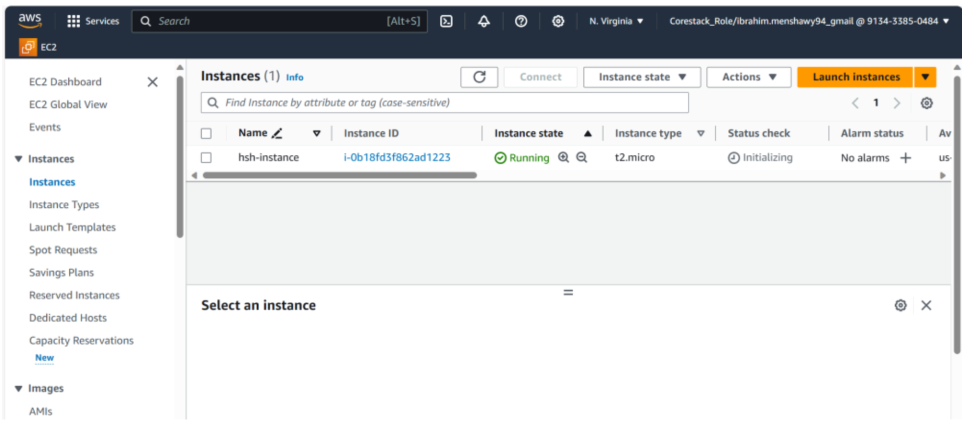
```
                    export AWS_DEFAULT_REGION='us-east-1'

                    terraform apply -auto-approve

                  '''

                }

              }

            }

          }

        }
```

- Build the terraform pipeline:


Applications
PG-DevOps-Project---Ho...
[root@ip-172-31-68-179...
Sun 12 Nov, 17:55 ibrahimmenshawy
localhost:8080/job/PG-DevOps-Project---Hotel-Side-Hospital/
Update
Dashboard
PG-DevOps-Project---Hotel-Side-Hospital
Stage View
Delete Pipeline
Full Stage View
GitHub
Rename
Pipeline Syntax
Build History
trend
Filter builds...
#25
Nov 12, 2023, 5:54 PM
Web Browser
Browse the web

	Declarative: Tool Install	Checkout	Terraform init	Terraform apply
Average stage times: (Average full run time: ~1min 17s)	1s	977ms	3s	38s
#25 Nov 12 17:54 No Changes	483ms	743ms	3s	1min 12s
#24 Nov 12 17:44 3 commits	2s	1s	4s	4s failed
#23 Nov 12 No Changes				

Check the instance on AWS

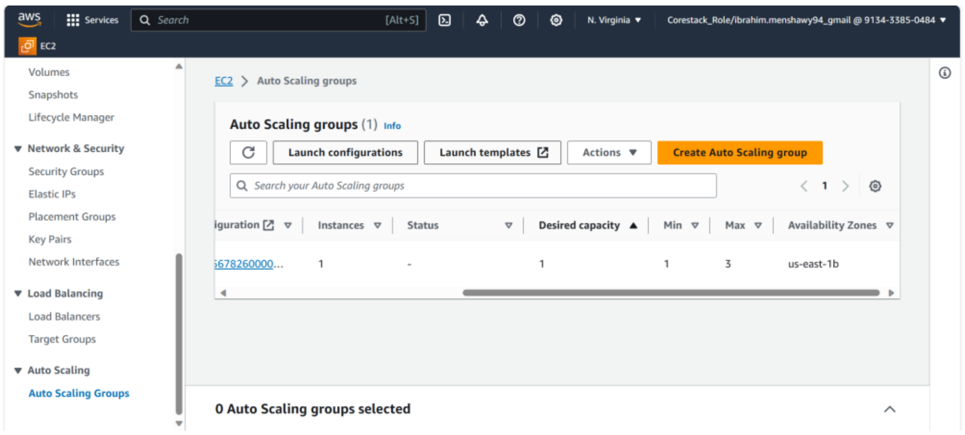


Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
hsh-instance	i-0b18fd3f862ad1223	Running	t2.micro	Initializing	No alarms	us-

Select an instance

- Check the Autoscaling group:



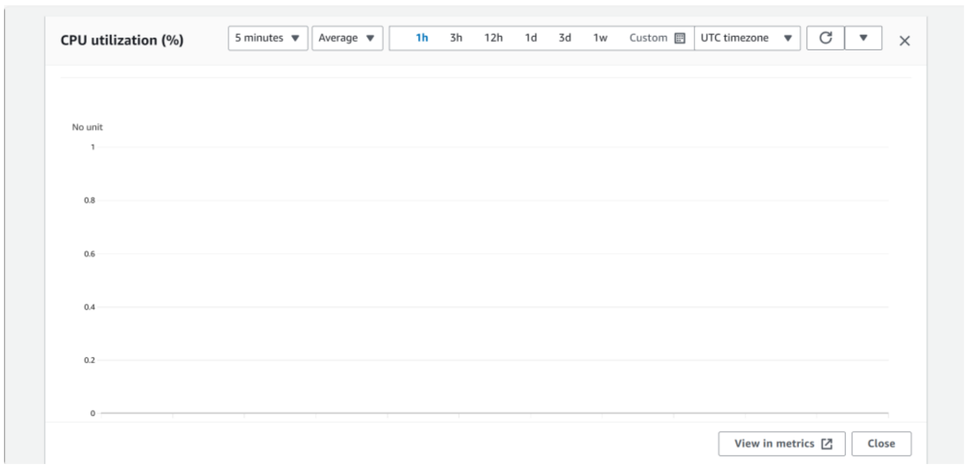
EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

iguration	Instances	Status	Desired capacity	Min	Max	Availability Zones
5678260000...	1	-	1	1	3	us-east-1b

0 Auto Scaling groups selected

CPU usage before installing stress utility:

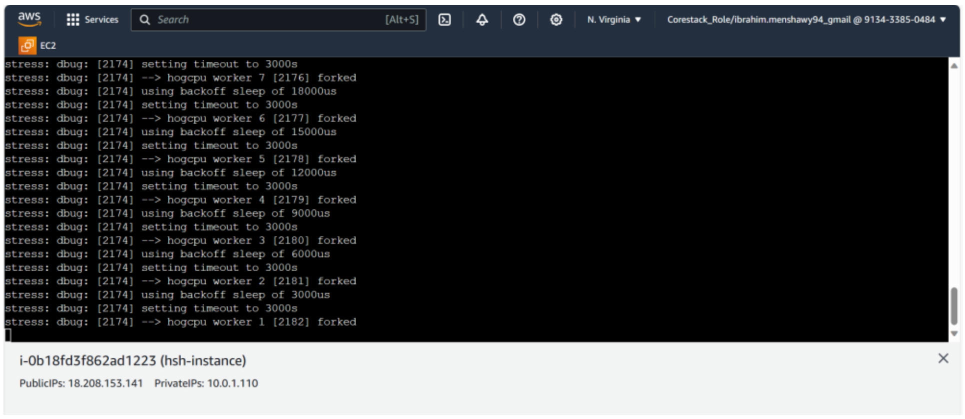


Installing stress utility on the instance (I used Ubuntu instance hence the difference in command)

sudo apt update

sudo apt install stress -y

sudo stress --cpu 8 --verbose --timeout 3000s

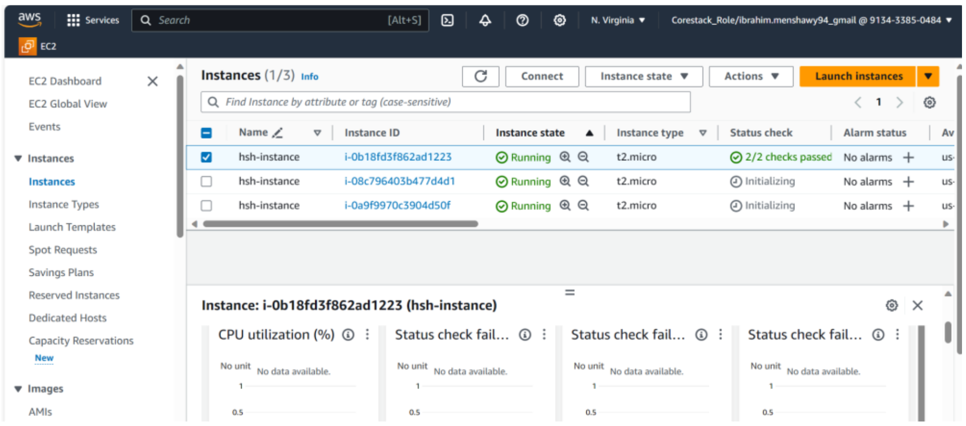


```
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 7 [2176] forked
stress: dbug: [2174] using backoff sleep of 18000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 6 [2177] forked
stress: dbug: [2174] using backoff sleep of 15000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 5 [2178] forked
stress: dbug: [2174] using backoff sleep of 12000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 4 [2179] forked
stress: dbug: [2174] using backoff sleep of 9000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 3 [2180] forked
stress: dbug: [2174] using backoff sleep of 6000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 2 [2181] forked
stress: dbug: [2174] using backoff sleep of 3000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 1 [2182] forked
```

i-0b18fd3f862ad1223 (hsh-instance)

PublicIPs: 18.208.153.141 PrivateIPs: 10.0.1.110

- Check the status of the instances afterr the triger is applied (cpu > 20%)



Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
hsh-instance	i-0b18fd3f862ad1223	Running	t2.micro	2/2 checks passed	No alarms
hsh-instance	i-08c796403b477d4d1	Running	t2.micro	Initializing	No alarms
hsh-instance	i-0a9f9970c3904d50f	Running	t2.micro	Initializing	No alarms

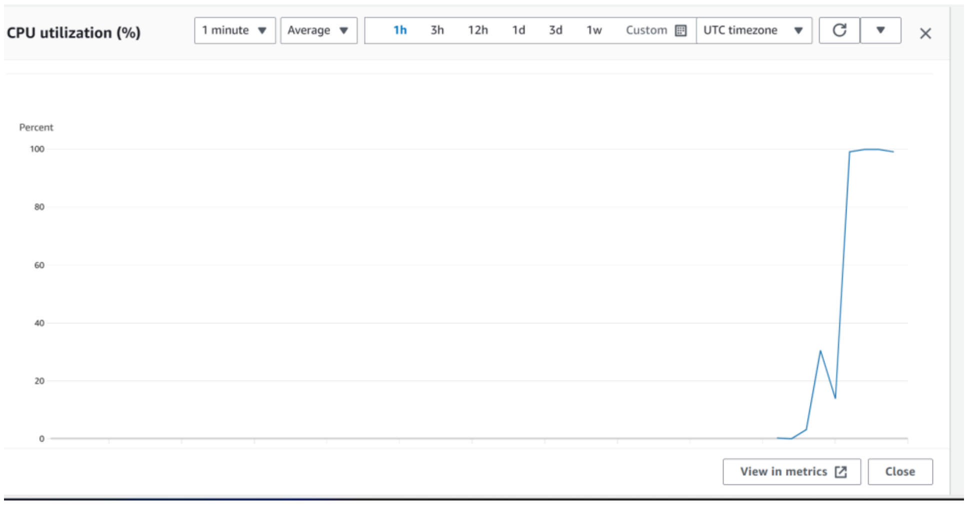
Instance: i-0b18fd3f862ad1223 (hsh-instance)

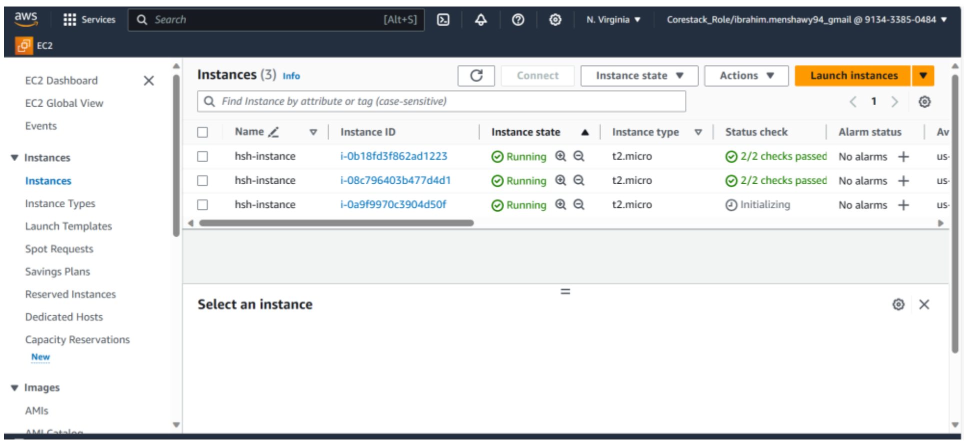
CPU utilization (%) — No unit — No data available.

Status check fail... — No unit — No data available.

Status check fail... — No unit — No data available.

Status check fail... — No unit — No data available.

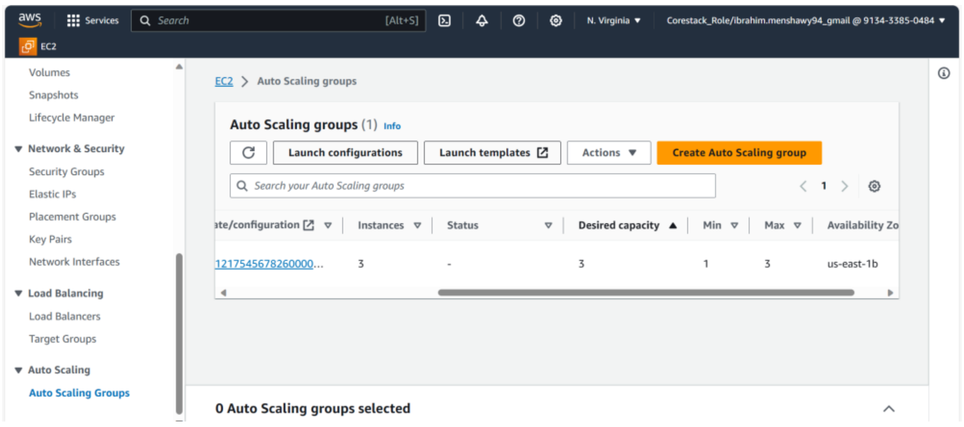




Instances (3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
hsh-instance	i-0b18fd3f862ad1223	Running	t2.micro	2/2 checks passed	No alarms	us-
hsh-instance	i-08c796403b477d4d1	Running	t2.micro	2/2 checks passed	No alarms	us-
hsh-instance	i-0a9f9970c3904d50f	Running	t2.micro	Initializing	No alarms	us-

Select an instance

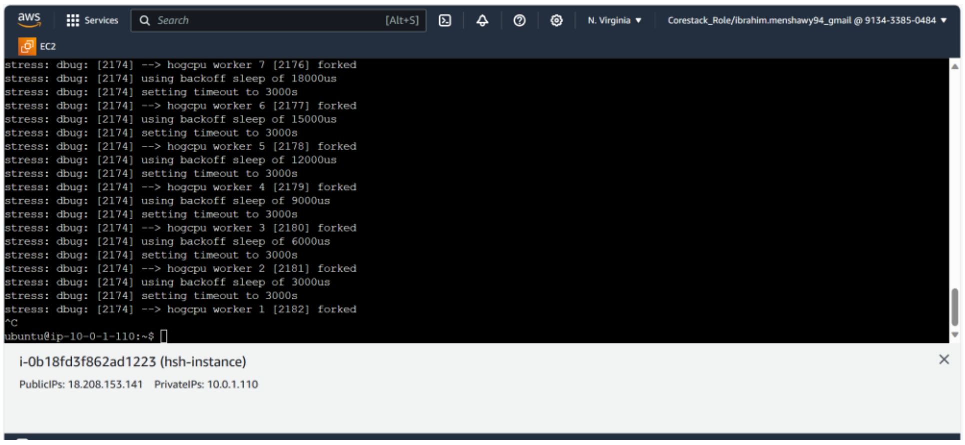


Auto Scaling groups (1) Info

ate/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zo
1217545678260000...	3	-	3	1	3	us-east-1b

0 Auto Scaling groups selected

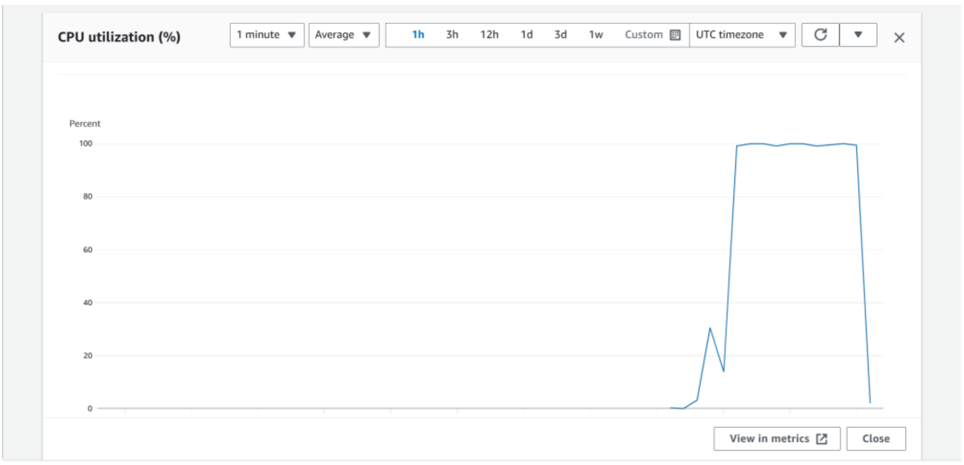
- Now stop the stress utility



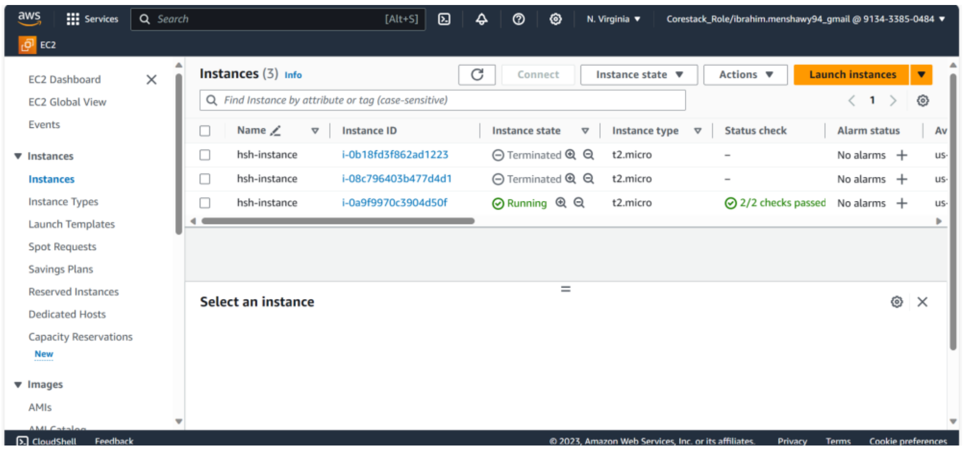
```
stress: dbug: [2174] --> hogcpu worker 7 [2176] forked
stress: dbug: [2174] using backoff sleep of 18000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 6 [2177] forked
stress: dbug: [2174] using backoff sleep of 15000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 5 [2178] forked
stress: dbug: [2174] using backoff sleep of 12000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 4 [2179] forked
stress: dbug: [2174] using backoff sleep of 9000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 3 [2180] forked
stress: dbug: [2174] using backoff sleep of 6000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 2 [2181] forked
stress: dbug: [2174] using backoff sleep of 3000us
stress: dbug: [2174] setting timeout to 3000s
stress: dbug: [2174] --> hogcpu worker 1 [2182] forked
^C
ubuntu@ip-10-0-1-110:~$
```

i-0b18fd3f862ad1223 (hsh-instance)

PublicIPs: 18.208.153.141 PrivateIPs: 10.0.1.110



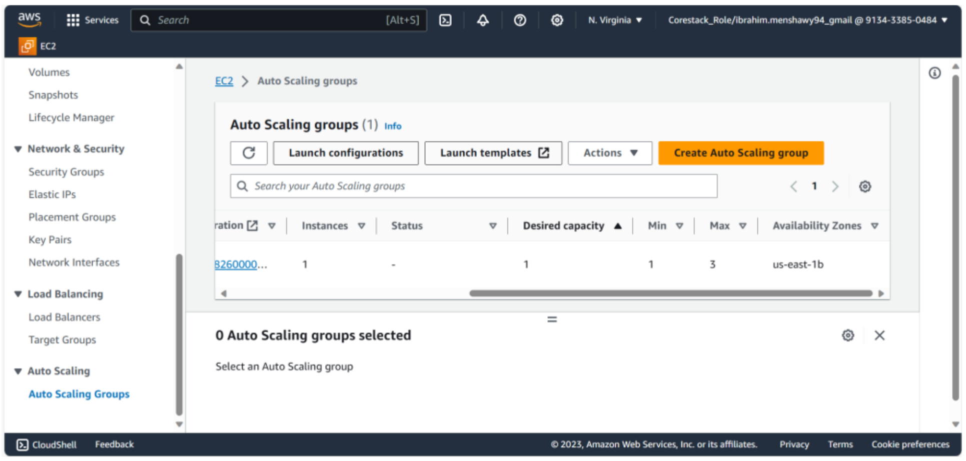
After some time, the group returnes to the desired state of 1 instance



Instances (3) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Av
☐	hsh-instance	i-0b18fd3f862ad1223	Terminated	t2.micro	–	No alarms	us-
☐	hsh-instance	i-08c796403b477d4d1	Terminated	t2.micro	–	No alarms	us-
☐	hsh-instance	i-0a9f9970c3904d50f	Running	t2.micro	2/2 checks passed	No alarms	us-

Select an instance



Auto Scaling groups (1)

ration	Instances	Status	Desired capacity	Min	Max	Availability Zones
8260000…	1	-	1	1	3	us-east-1b

0 Auto Scaling groups selected

Select an Auto Scaling group

The END

Thank YOU