CASE STUDY -CREATING AN ARCHITECTURE USING TERRAFORM ON AWS

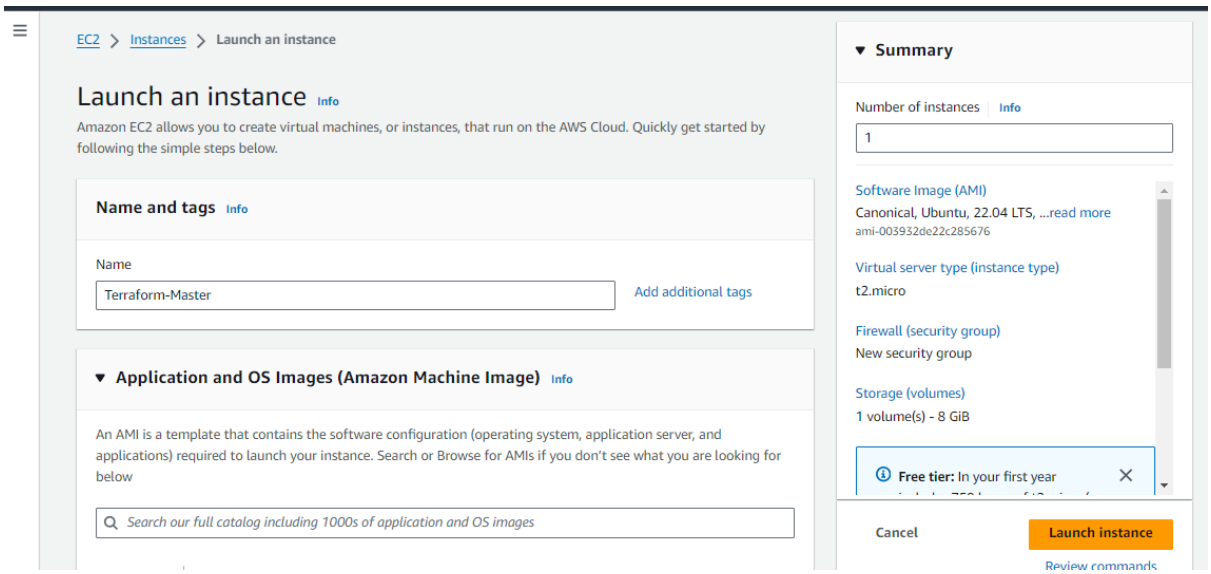
You work as a DevOps Engineer in leading Software Company. You have been asked to build an infrastructure safely and efficiently.

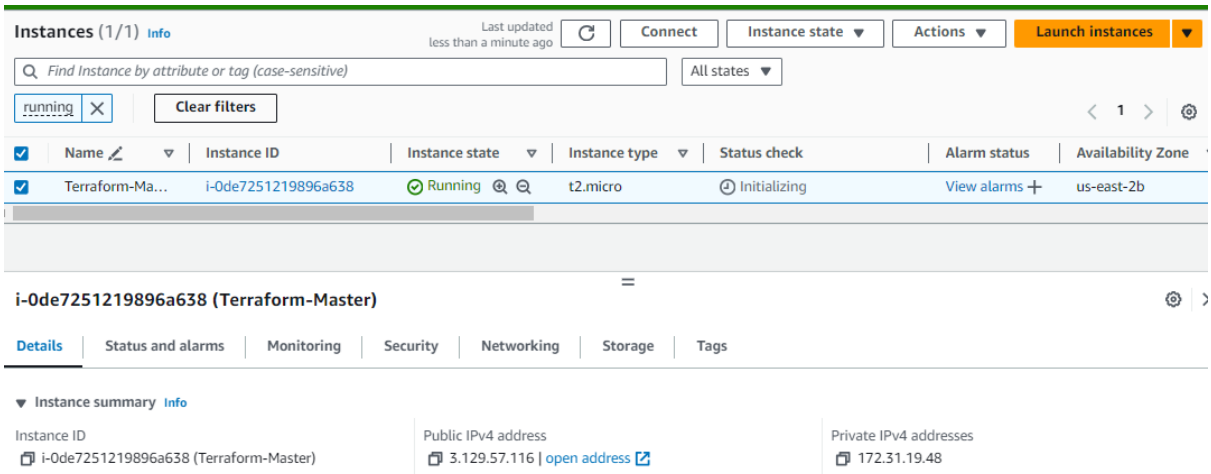
The company Requirements:

1. Use AWS cloud Provider and the software to be installed is Apache2
2. Use Ubuntu AMI

The company wants the Architecture to have the following services:

1. Create a template with a VPC, 2 subnets and 1 instance in each subnet
2. Attach Security groups, internet gateway and network interface to the instance





Example:

ssh -i "salman-Ohio.pem" ubuntu@ec2-3-129-57-116.us-east-2.compute.amazonaws.com

> **Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

```
C:\Users\shaik\Downloads>ssh -i "salman-Ohio.pem" ubuntu@ec2-3-129-57-116.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-129-57-116.us-east-2.compute.amazonaws.com (3.129.57.116)' can't be established.
ED25519 key fingerprint is SHA256:QdjEekwBhd20MbxJCdWOfxfk6fnL05j0Kd+pGdu5TMU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-129-57-116.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)
```

```
C:\WINDOWS\system32\cmd.

ubuntu@ip-172-31-19-48:~$ sudo hostnamectl set-hostname Terraform-Master
ubuntu@ip-172-31-19-48:~$ exit
logout
Connection to ec2-3-129-57-116.us-east-2.compute.amazonaws.com closed.

C:\Users\shaik\Downloads>ssh -i "salman-Ohio.pem" ubuntu@ec2-3-129-57-116.us-east-2.compute.amazonaws.com
```

```
ubuntu@Terraform-Master: ~

ubuntu@Terraform-Master:~$ sudo apt update
```

```
ubuntu@Terraform-Master: ~

ubuntu@Terraform-Master:~$ sudo nano terraform-install.sh
```

Download and add the GPG key for the HashiCorp repository

wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

Add the HashiCorp repository to your sources list

echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list

Update the package list and install Terraform

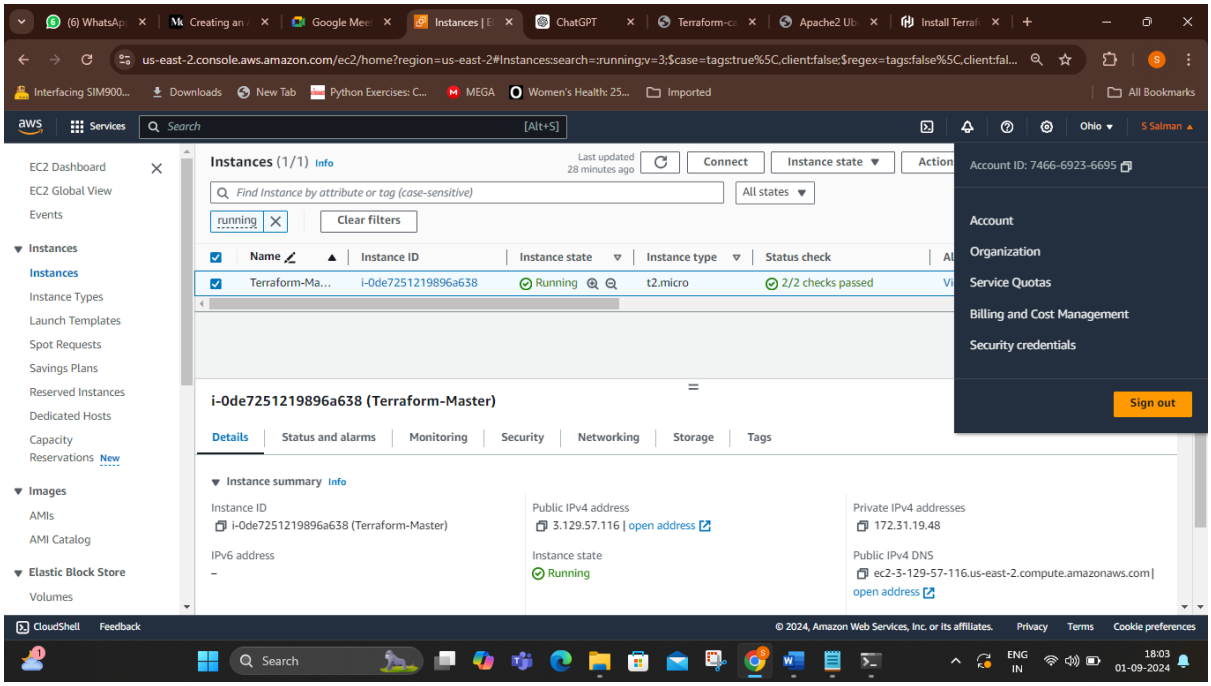
sudo apt update && sudo apt install terraform -y

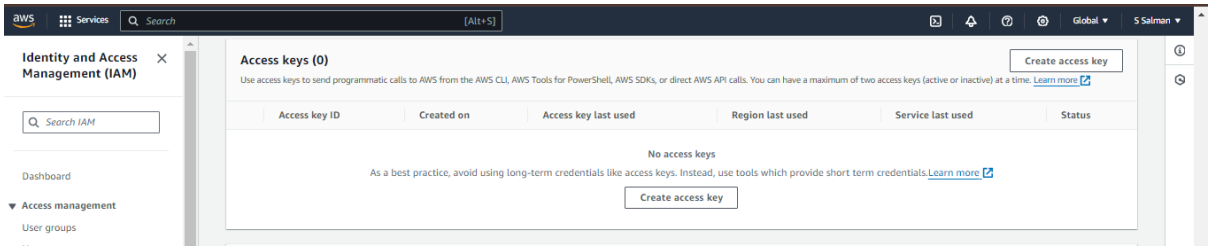
```
ubuntu@Terraform-Master: ~

  GNU nano 6.2                                    terraform-install.sh *
# Download and add the GPG key for the HashiCorp repository
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

# Add the HashiCorp repository to your sources list
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc

# Update the package list and install Terraform
sudo apt update && sudo apt install terraform -y
```

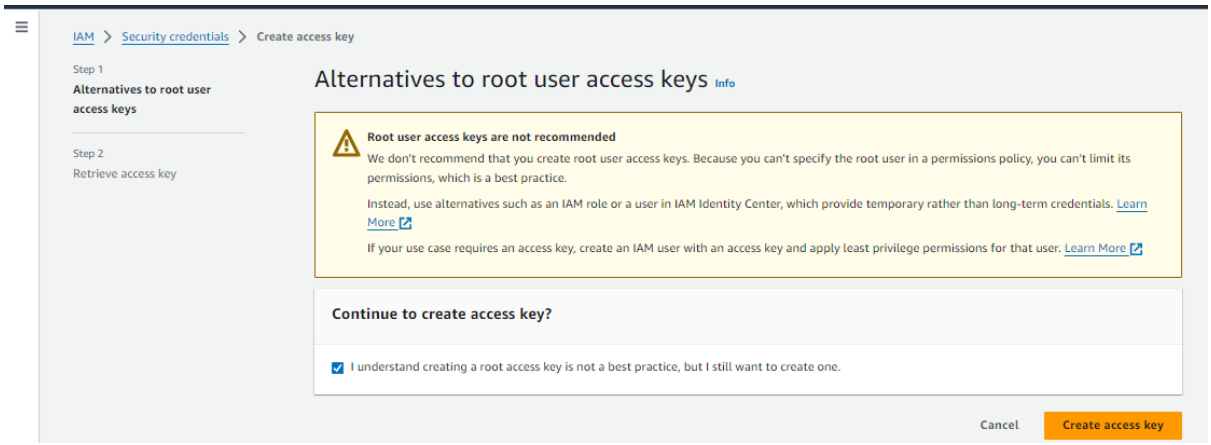
```
ubuntu@Terraform-Master: ~
ubuntu@Terraform-Master:~$ sudo nano terraform-install.sh
ubuntu@Terraform-Master:~$ sudo bash terraform-install.sh
--2024-09-01 12:27:06--  https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 108.156.184.46, 108.156.184.65, 108.156.184.5, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|108.156.184.46|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

-                                   100%[===================================================================>]   3.89K  --.-KB/s    in 0s

2024-09-01 12:27:07 (1.26 GB/s) - written to stdout [3980/3980]

deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://apt.releases.hashicorp.com jammy/main amd64 Packages [149 kB]
Fetched 161 kB in 1s (182 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
48 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 48 not upgraded.
Need to get 28.0 MB of archives.
After this operation, 89.1 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.9.5-1 [28.0 MB]
Fetched 28.0 MB in 0s (79.8 MB/s)
```

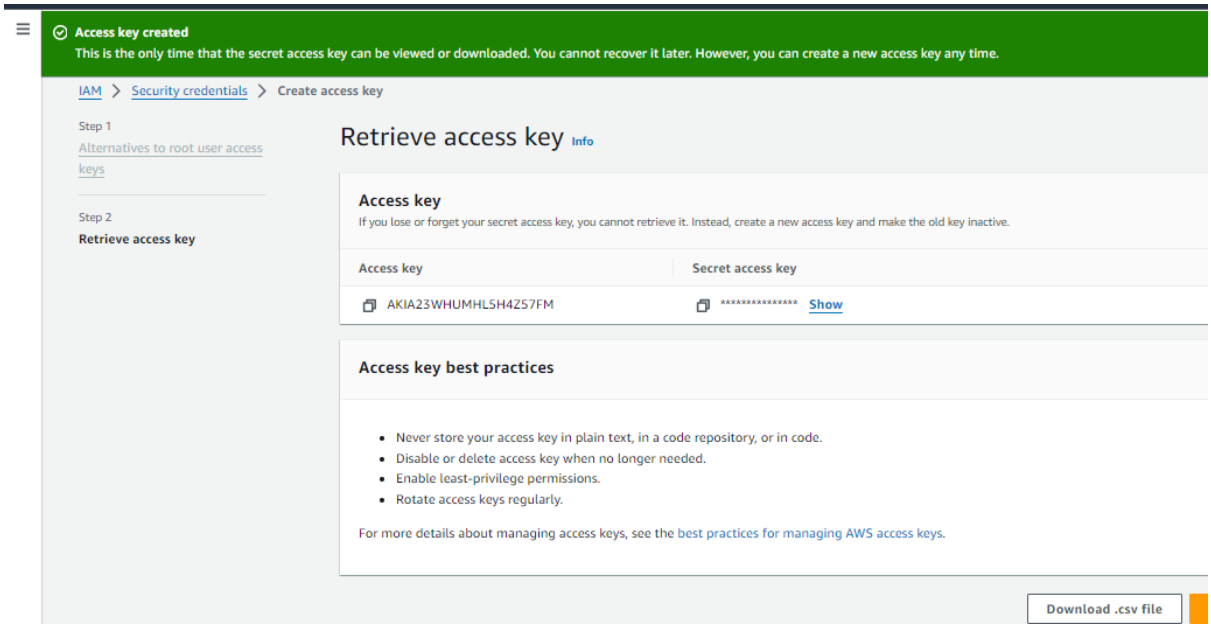
```
ubuntu@Terraform-Master: ~
ubuntu@Terraform-Master:~$ terraform --version
Terraform v1.9.5
on linux_amd64
ubuntu@Terraform-Master:~$
```

```
ubuntu@Terraform-Master: ~
ubuntu@Terraform-Master:~$ ls
terraform-install.sh
ubuntu@Terraform-Master:~$ sudo nano main.tf
```


us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#Instances:search=:running;v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:fal...
Services
Search
[Alt+S]
Ohio
S Salman
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Instances (1/1) Info
Last updated 28 minutes ago
Connect
Instance state
Action
Find Instance by attribute or tag (case-sensitive)
All states
running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Terraform-Ma...
i-0de7251219896a638
Running
t2.micro
2/2 checks passed
Account ID: 7466-6923-6695
Account
Organization
Service Quotas
Billing and Cost Management
Security credentials
Sign out
i-0de7251219896a638 (Terraform-Master)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0de7251219896a638 (Terraform-Master)
IPv6 address
–
Public IPv4 address
3.129.57.116 | open address
Instance state
Running
Private IPv4 addresses
172.31.19.48
Public IPv4 DNS
ec2-3-129-57-116.us-east-2.compute.amazonaws.com | open address
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Global
S Salman
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Access keys (0)
Create access key
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more
Access key ID
Created on
Access key last used
Region last used
Service last used
Status
No access keys
As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more
Create access key


IAM > Security credentials > Create access key
Step 1
Alternatives to root user access keys
Step 2
Retrieve access key
Alternatives to root user access keys Info
Root user access keys are not recommended
We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.
Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. Learn More
If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. Learn More
Continue to create access key?
I understand creating a root access key is not a best practice, but I still want to create one.
Cancel
Create access key




```
ubuntu@Terraform-Master:~$ sudo cat main.tf
provider "aws" {
        region = "us-east-2"
        access_key = "AKIA23WHUMHL5H4Z57FM"
        secret_key = "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"
}

# Create a VPC

resource "aws_vpc" "testvpc" {
  cidr_block = "10.0.0.0/16"

  tags = {
     Name = "testvpc"
  }
}
# Create a Public Subnet

resource "aws_subnet" "testsbnt1" {
   vpc_id = aws_vpc.testvpc.id
   cidr_block = "10.0.1.0/24"
     map_public_ip_on_launch = "true"
     availability_zone = "us-east-2a"

   tags = {
       Name = "testsbnt1"
    }
}
# Create a Private Subnet

resource "aws_subnet" "testsbnt2" {
    vpc_id = aws_vpc.testvpc.id
    cidr_block = "10.0.2.0/24"
      map_public_ip_on_launch = "false"
      availability_zone = "us-east-2b"

     tags = {
       Name = "testsbnt2"
     }
```

```
ubuntu@Terraform-Master: ~

      availability_zone = "us-east-2b"

     tags = {
       Name = "testsbnt2"
     }
}
# Create an Internet Gateway

resource "aws_internet_gateway" "testigw" {
    vpc_id = aws_vpc.testvpc.id
    tags = {
      Name = "testigw"
    }
}
# Create a Route Table for Public Subnet

resource "aws_route_table" "testrtb1" {
      vpc_id = aws_vpc.testvpc.id

      route {
        cidr_block = "0.0.0.0/0"
        gateway_id = aws_internet_gateway.testigw.id
      }

      route {
        ipv6_cidr_block = "::/0"
        gateway_id = aws_internet_gateway.testigw.id
      }
}
# Associate Public Route Table with Public Subnet

resource "aws_route_table_association" "testassoc1" {
        subnet_id = aws_subnet.testsbnt1.id
        route_table_id = aws_route_table.testrtb1.id
}
```

```
ubuntu@Terraform-Master: ~

# Create a Private Route Table for Subnet 2

resource "aws_route_table" "testrtb2" {
     vpc_id = aws_vpc.testvpc.id

     route {
        cidr_block = "0.0.0.0/0"
        gateway_id = aws_nat_gateway.nat.id
     }

     tags = {
       Name = "testrtb2"
     }
}
# Associate Route Table with Private Subnet

resource "aws_route_table_association" "testassoc2" {
      subnet_id = aws_subnet.testsbnt2.id
      route_table_id = aws_route_table.testrtb2.id
}
# Assign ENI with IP

resource "aws_network_interface" "testeni1" {
           subnet_id = aws_subnet.testsbnt1.id
           private_ips = ["10.0.1.10"]
           security_groups = [aws_security_group.testsg.id]
}

resource "aws_network_interface" "testeni2" {
           subnet_id = aws_subnet.testsbnt2.id
           private_ips = ["10.0.2.10"]
           security_groups = [aws_security_group.testsg.id]
}
# Assign Elastic IP to ENI
```

ubuntu@Terraform-Master: ~

```
# Assign Elastic IP to ENI

resource "aws_eip" "testeip1" {
        domain = "vpc"
        network_interface = aws_network_interface.testeni1.id
        associate_with_private_ip = "10.0.1.10"
        depends_on= [aws_internet_gateway.testigw, aws_instance.Instance1]
        tags = {
           Name = "testeip1"
        }
}
# Create an Elastic IP Address for NAT Gateway

resource "aws_eip" "testeip2" {
        domain = "vpc"
        associate_with_private_ip = "10.0.2.10"
        depends_on= [aws_internet_gateway.testigw]
        tags = {
           Name = "testeip2"
        }
}
# Create a NAT Gateway for VPC

resource "aws_nat_gateway" "nat" {
     allocation_id = aws_eip.testeip2.id
     subnet_id = aws_subnet.testsbnt2.id

     tags = {
      Name = "nat"
     }
}
# Create a Security Group

resource "aws_security_group" "testsg" {
   description = "Allow limited inbound external traffic"
```

ubuntu@Terraform-Master: ~

```
# Create a Security Group

resource "aws_security_group" "testsg" {
   description = "Allow limited inbound external traffic"
   vpc_id = aws_vpc.testvpc.id
   name = "testsg"

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

   ingress {
     protocol = "tcp"
     cidr_blocks = ["0.0.0.0/0"]
     from_port = 80
     to_port = 80
   }

    ingress {
      protocol = "tcp"
      cidr_blocks = ["0.0.0.0/0"]
      from_port = 443
      to_port = 443
     }
    egress {
      from_port = 0
      to_port = 0
      protocol = "-1"
      cidr_blocks = ["0.0.0.0/0"]

    }
    tags = {
```

```
      to_port = 0
      protocol = "-1"
      cidr_blocks = ["0.0.0.0/0"]

    }
    tags = {
      Name = "testsg"
    }
}
# Create Linux Server & Install/Enable Apache2 (Instance 1)

resource "aws_instance" "Instance1" {
    ami = "ami-0b8b44ec9a8f90422"
    instance_type = "t2.micro"
    availability_zone = "us-east-2a"
    key_name = "salman-Ohio"
    network_interface {
      device_index = 0
      network_interface_id = aws_network_interface.testeni1.id
   }

 user_data = <<-EOF
   #!/bin/bash
   sudo apt update -y
   sudo apt install apache2 -y
   sudo systemctl start apache2
   sudo systemctl enable apache2
   EOF

   tags = {
     Name = "Instance1"
    }
}
# Create Linux Server & Install/Enable Apache2 Here (Instance 2)

resource "aws_instance" "Instance2" {
    ami = "ami-0b8b44ec9a8f90422"
    instance_type = "t2.micro"
    availability_zone = "us-east-2b"
```

```
 user_data = <<-EOF
   #!/bin/bash
   sudo apt update -y
   sudo apt install apache2 -y
   sudo systemctl start apache2
   sudo systemctl enable apache2
   EOF

   tags = {
     Name = "Instance1"
    }
}
# Create Linux Server & Install/Enable Apache2 Here (Instance 2)

resource "aws_instance" "Instance2" {
    ami = "ami-0b8b44ec9a8f90422"
    instance_type = "t2.micro"
    availability_zone = "us-east-2b"
    key_name = "salman-Ohio"
    network_interface {
      device_index = 0
      network_interface_id = aws_network_interface.testeni2.id
   }

 user_data = <<-EOF
   #!/bin/bash
   sudo apt update -y
   sudo apt install apache2 -y
   sudo systemctl start apache2
   sudo systemctl enable apache2
   EOF

   tags = {
     Name = "Instance2"
    }
}
ubuntu@Terraform-Master:~$
```

```
ubuntu@Terraform-Master: ~

ubuntu@Terraform-Master:~$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.65.0...
- Installed hashicorp/aws v5.65.0 (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.
ubuntu@Terraform-Master:~$
```

```
ubuntu@Terraform-Master: ~

ubuntu@Terraform-Master:~$ 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.testeip1 will be created
  + resource "aws_eip" "testeip1" {
      + allocation_id             = (known after apply)
      + arn                       = (known after apply)
      + associate_with_private_ip = "10.0.1.10"
      + association_id            = (known after apply)
      + carrier_ip                = (known after apply)
      + customer_owned_ip         = (known after apply)
      + domain                    = "vpc"
      + 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)
      + ptr_record                = (known after apply)
      + public_dns                = (known after apply)
      + public_ip                 = (known after apply)
      + public_ipv4_pool          = (known after apply)
      + tags                      = {
          + "Name" = "testeip1"
        }
      + tags_all                  = {
          + "Name" = "testeip1"
        }
      + vpc                       = (known after apply)
    }

  # aws_eip.testeip2 will be created
  + resource "aws_eip" "testeip2" {
      + allocation_id             = (known after apply)
      + arn                       = (known after apply)
```

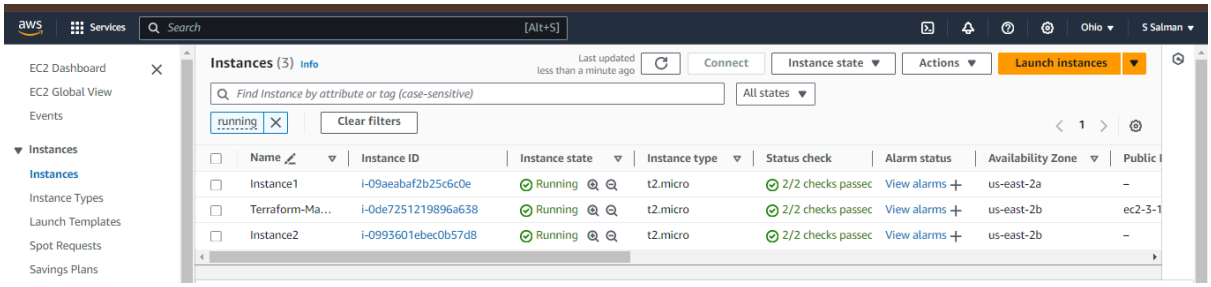
```
Plan: 16 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------

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.
ubuntu@Terraform-Master:~$
```

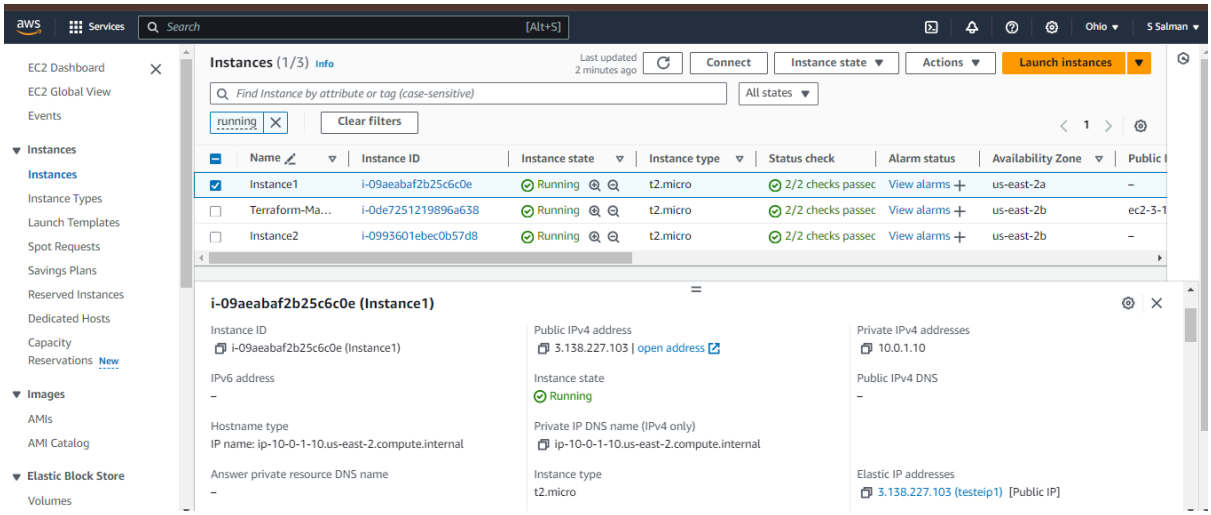
```
ubuntu@Terraform-Master: ~
ubuntu@Terraform-Master:~$ terraform apply
aws_vpc.testvpc: Refreshing state... [id=vpc-08227f094b1c8868e]
aws_subnet.testsbnt2: Refreshing state... [id=subnet-06951980ac40ff6a6]
aws_internet_gateway.testigw: Refreshing state... [id=igw-0bb49190d67b7e892]
aws_subnet.testsbnt1: Refreshing state... [id=subnet-0502225d02da11056]
aws_security_group.testsg: Refreshing state... [id=sg-084f9233a37f3aa0e]
aws_eip.testeip2: Refreshing state... [id=eipalloc-019e4d58a933535aa]
aws_route_table.testrtb1: Refreshing state... [id=rtb-0143c8e976879fbce]
aws_network_interface.testeni2: Refreshing state... [id=eni-0e9a472194ced5062]
aws_network_interface.testeni1: Refreshing state... [id=eni-05cdaa9ae8dbcfa15]
aws_route_table_association.testassoc1: Refreshing state... [id=rtbassoc-028fe31840a85d18d]
aws_nat_gateway.nat: Refreshing state... [id=nat-042538c565eaafe34]
aws_route_table.testrtb2: Refreshing state... [id=rtb-038180af973b29bbc]
aws_route_table_association.testassoc2: Refreshing state... [id=rtbassoc-00dfdd654b2b4a177]

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

Terraform will perform the following actions:

  # aws_eip.testeip1 will be created
  + resource "aws_eip" "testeip1" {
      + allocation_id             = (known after apply)
      + arn                       = (known after apply)
      + associate_with_private_ip = "10.0.1.10"
      + association_id            = (known after apply)
      + carrier_ip                = (known after apply)
      + customer_owned_ip         = (known after apply)
      + domain                    = "vpc"
      + id                        = (known after apply)
      + instance                  = (known after apply)
      + network_border_group      = (known after apply)
      + network_interface         = "eni-05cdaa9ae8dbcfa15"
      + private_dns               = (known after apply)
      + private_ip                = (known after apply)
      + ptr_record                = (known after apply)
      + public_dns                = (known after apply)
      + public_ip                 = (known after apply)
```

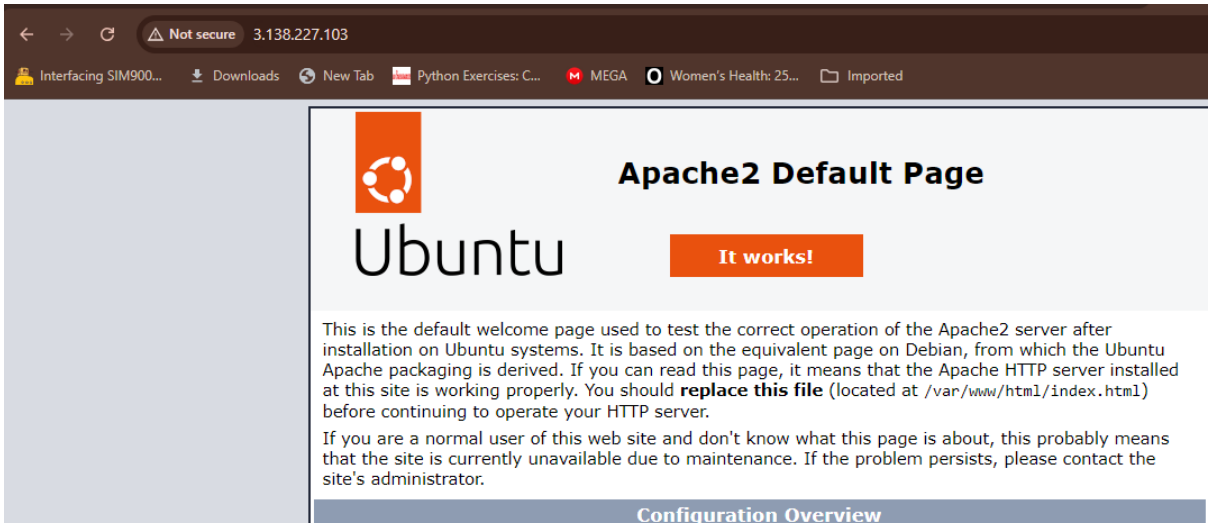
```
Plan: 16 to add, 0 to change, 0 to destroy.

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_vpc.testvpc: Creating...
aws_vpc.testvpc: Creation complete after 1s [id=vpc-08227f094b1c8868e]
aws_internet_gateway.testigw: Creating...
aws_security_group.testsg: Creating...
aws_subnet.testsbnt1: Creating...
aws_subnet.testsbnt2: Creating...
aws_internet_gateway.testigw: Creation complete after 0s [id=igw-0bb49190d67b7e892]
aws_route_table.testrtb1: Creating...
aws_eip.testeip2: Creating...
aws_subnet.testsbnt2: Creation complete after 1s [id=subnet-06951980ac40ff6a6]
aws_eip.testeip2: Creation complete after 1s [id=eipalloc-019e4d58a933535aa]
aws_nat_gateway.nat: Creating...
aws_route_table.testrtb1: Creation complete after 1s [id=rtb-0143c8e976879fbce]
aws_security_group.testsg: Creation complete after 2s [id=sg-084f9233a37f3aa0e]
aws_network_interface.testeni2: Creating...
aws_network_interface.testeni2: Creation complete after 1s [id=eni-0e9a472194ced5062]
aws_instance.Instance2: Creating...
aws_subnet.testsbnt1: Still creating... [10s elapsed]
aws_nat_gateway.nat: Still creating... [10s elapsed]
aws_subnet.testsbnt1: Creation complete after 11s [id=subnet-0502225d02da11056]
aws_network_interface.testeni1: Creating...
aws_route_table_association.testassoc1: Creating...
aws_route_table_association.testassoc1: Creation complete after 0s [id=rtbassoc-028fe31840a85d18d]
aws_network_interface.testeni1: Creation complete after 0s [id=eni-05cdaa9ae8dbcfa15]
aws_instance.Instance1: Creating...
aws_nat_gateway.nat: Still creating... [20s elapsed]
aws_nat_gateway.nat: Still creating... [30s elapsed]
aws_nat_gateway.nat: Still creating... [40s elapsed]
aws_nat_gateway.nat: Still creating... [50s elapsed]
aws_nat_gateway.nat: Still creating... [1m0s elapsed]
```

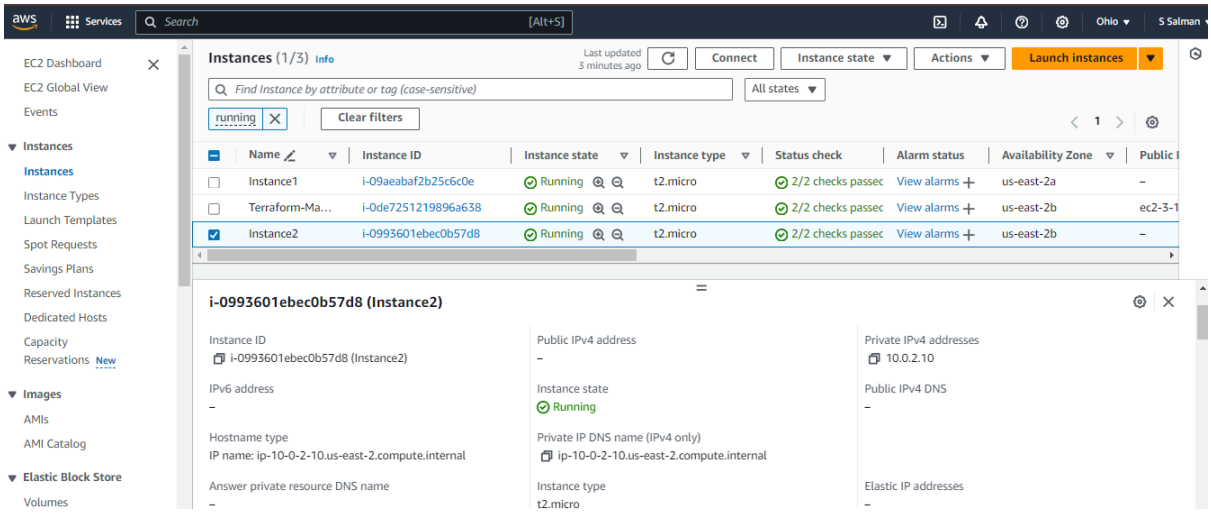
```
aws_route_table_association.testassoc2: Creation complete after 0s [id=rtbassoc-00dfdd654b2b4a177]

Error: creating EC2 Instance: operation error EC2: RunInstances, https response error StatusCode: 400, RequestID: 2319da4f-e6be-445c-9a80-ee03461a
2328, api error InvalidKeyPair.NotFound: The key pair 'Terraform' does not exist

  with aws_instance.Instance1,
  on main.tf line 173, in resource "aws_instance" "Instance1":
 173: resource "aws_instance" "Instance1" {


Error: creating EC2 Instance: operation error EC2: RunInstances, https response error StatusCode: 400, RequestID: e6f72b15-17f6-401d-92ab-449df77f
9a59, api error InvalidKeyPair.NotFound: The key pair 'Terraform' does not exist

  with aws_instance.Instance2,
  on main.tf line 197, in resource "aws_instance" "Instance2":
 197: resource "aws_instance" "Instance2" {
```

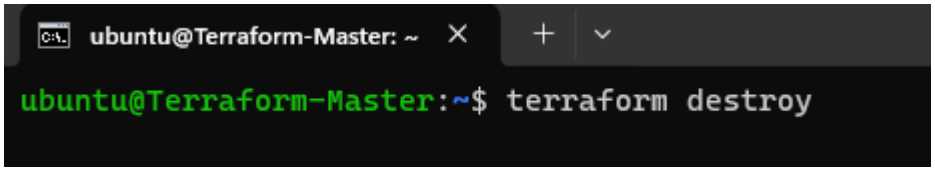
```
Plan: 3 to add, 1 to change, 0 to destroy.

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.Instance2: Creating...
aws_instance.Instance1: Creating...
aws_route_table.testrtb2: Modifying... [id=rtb-038180af973b29bbc]
aws_route_table.testrtb2: Modifications complete after 0s [id=rtb-038180af973b29bbc]
aws_instance.Instance2: Still creating... [10s elapsed]
aws_instance.Instance1: Still creating... [10s elapsed]
aws_instance.Instance2: Still creating... [20s elapsed]
aws_instance.Instance1: Still creating... [20s elapsed]
aws_instance.Instance2: Still creating... [30s elapsed]
aws_instance.Instance1: Still creating... [30s elapsed]
aws_instance.Instance1: Creation complete after 32s [id=i-09aeabaf2b25c6c0e]
aws_eip.testeip1: Creating...
aws_instance.Instance2: Creation complete after 32s [id=i-0993601ebec0b57d8]
aws_eip.testeip1: Creation complete after 1s [id=eipalloc-0175632987fbdc116]

Apply complete! Resources: 3 added, 1 changed, 0 destroyed.
ubuntu@Terraform-Master:~$
```












```
Plan: 0 to add, 0 to change, 16 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_eip.testeip1: Destroying... [id=eipalloc-0175632987fbdc116]
aws_route_table_association.testassoc2: Destroying... [id=rtbassoc-00dfdd654b2b4a177]
aws_instance.Instance2: Destroying... [id=i-0993601ebec0b57d8]
aws_route_table_association.testassoc1: Destroying... [id=rtbassoc-028fe31840a85d18d]
aws_route_table_association.testassoc1: Destruction complete after 0s
aws_route_table_association.testassoc2: Destruction complete after 0s
aws_route_table.testrtb2: Destroying... [id=rtb-038180af973b29bbc]
aws_route_table.testrtb1: Destroying... [id=rtb-0143c8e976879fbce]
aws_route_table.testrtb1: Destruction complete after 0s
aws_route_table.testrtb2: Destruction complete after 0s
aws_nat_gateway.nat: Destroying... [id=nat-042538c565eaafe34]
aws_eip.testeip1: Destruction complete after 1s
aws_instance.Instance1: Destroying... [id=i-09aeabaf2b25c6c0e]
```

```
  Enter a value: yes

aws_eip.testeip1: Destroying... [id=eipalloc-0175632987fbdc116]
aws_route_table_association.testassoc2: Destroying... [id=rtbassoc-00dfdd654b2b4a177]
aws_instance.Instance2: Destroying... [id=i-0993601ebec0b57d8]
aws_route_table_association.testassoc1: Destroying... [id=rtbassoc-028fe31840a85d18d]
aws_route_table_association.testassoc1: Destruction complete after 0s
aws_route_table_association.testassoc2: Destruction complete after 0s
aws_route_table.testrtb2: Destroying... [id=rtb-038180af973b29bbc]
aws_route_table.testrtb1: Destroying... [id=rtb-0143c8e976879fbce]
aws_route_table.testrtb1: Destruction complete after 0s
aws_route_table.testrtb2: Destruction complete after 0s
aws_nat_gateway.nat: Destroying... [id=nat-042538c565eaafe34]
aws_eip.testeip1: Destruction complete after 1s
aws_instance.Instance1: Destroying... [id=i-09aeabaf2b25c6c0e]
aws_instance.Instance2: Still destroying... [id=i-0993601ebec0b57d8, 10s elapsed]
aws_nat_gateway.nat: Still destroying... [id=nat-042538c565eaafe34, 10s elapsed]
aws_instance.Instance1: Still destroying... [id=i-09aeabaf2b25c6c0e, 10s elapsed]
aws_instance.Instance2: Still destroying... [id=i-0993601ebec0b57d8, 20s elapsed]
aws_nat_gateway.nat: Still destroying... [id=nat-042538c565eaafe34, 20s elapsed]
aws_instance.Instance1: Still destroying... [id=i-09aeabaf2b25c6c0e, 20s elapsed]
aws_instance.Instance2: Destruction complete after 30s
aws_network_interface.testeni2: Destroying... [id=eni-0e9a472194ced5062]
aws_network_interface.testeni2: Destruction complete after 0s
aws_nat_gateway.nat: Still destroying... [id=nat-042538c565eaafe34, 30s elapsed]
aws_instance.Instance1: Destruction complete after 29s
aws_network_interface.testeni1: Destroying... [id=eni-05cdaa9ae8dbcfa15]
aws_network_interface.testeni1: Destruction complete after 1s
aws_subnet.testsbnt1: Destroying... [id=subnet-0502225d02da11056]
aws_security_group.testsg: Destroying... [id=sg-084f9233a37f3aa0e]
aws_subnet.testsbnt1: Destruction complete after 0s
aws_security_group.testsg: Destruction complete after 0s
aws_nat_gateway.nat: Still destroying... [id=nat-042538c565eaafe34, 40s elapsed]
aws_nat_gateway.nat: Still destroying... [id=nat-042538c565eaafe34, 50s elapsed]
aws_nat_gateway.nat: Destruction complete after 51s
aws_eip.testeip2: Destroying... [id=eipalloc-019e4d58a933535aa]
aws_subnet.testsbnt2: Destroying... [id=subnet-06951980ac40ff6a6]
aws_subnet.testsbnt2: Destruction complete after 0s
aws_eip.testeip2: Destruction complete after 1s
aws_internet_gateway.testigw: Destroying... [id=igw-0bb49190d67b7e892]
aws_internet_gateway.testigw: Destruction complete after 0s
aws_vpc.testvpc: Destroying... [id=vpc-08227f094b1c8868e]
aws_vpc.testvpc: Destruction complete after 1s

Destroy complete! Resources: 16 destroyed.
ubuntu@Terraform-Master:~$
```