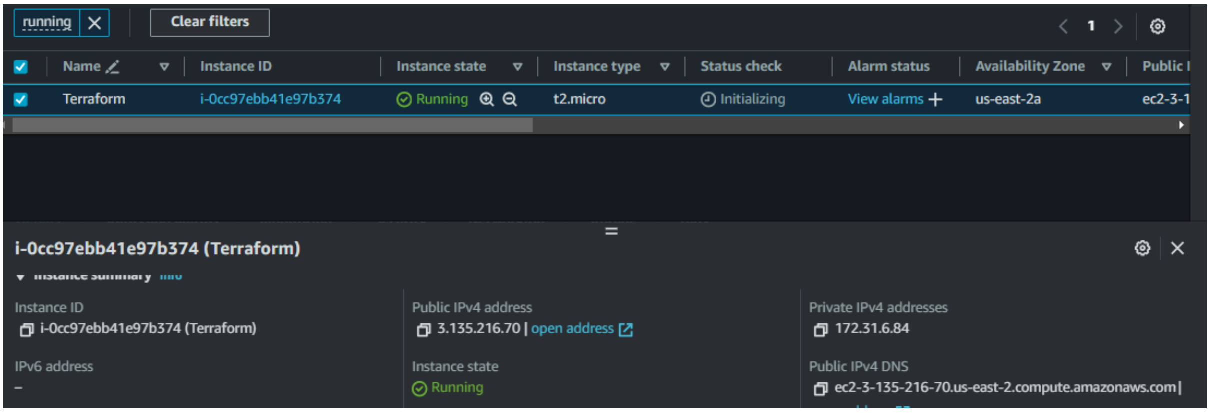
Terraform Assignment - 1

You have been asked to:

- Create an EC2 service in the default subnet in the ohio region

1. Launch Instances



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
☑	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	Initializing	View alarms +	us-east-2a	ec2-3-1

i-0cc97ebb41e97b374 (Terraform)

Instance summary Info

Instance ID
i-0cc97ebb41e97b374 (Terraform)

Public IPv4 address
3.135.216.70 | open address

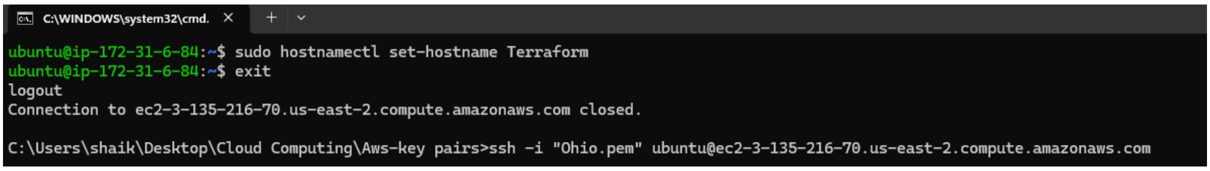
Private IPv4 addresses
172.31.6.84

IPv6 address
–

Instance state
Running

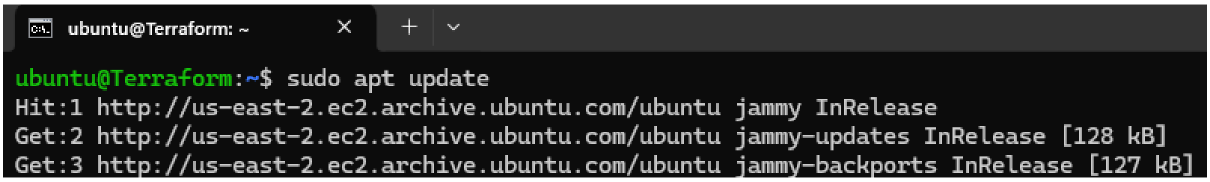
Public IPv4 DNS
ec2-3-135-216-70.us-east-2.compute.amazonaws.com |

2. change hostname



```
ubuntu@ip-172-31-6-84:~$ sudo hostnamectl set-hostname Terraform
ubuntu@ip-172-31-6-84:~$ exit
logout
Connection to ec2-3-135-216-70.us-east-2.compute.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Ohio.pem" ubuntu@ec2-3-135-216-70.us-east-2.compute.amazonaws.com
```

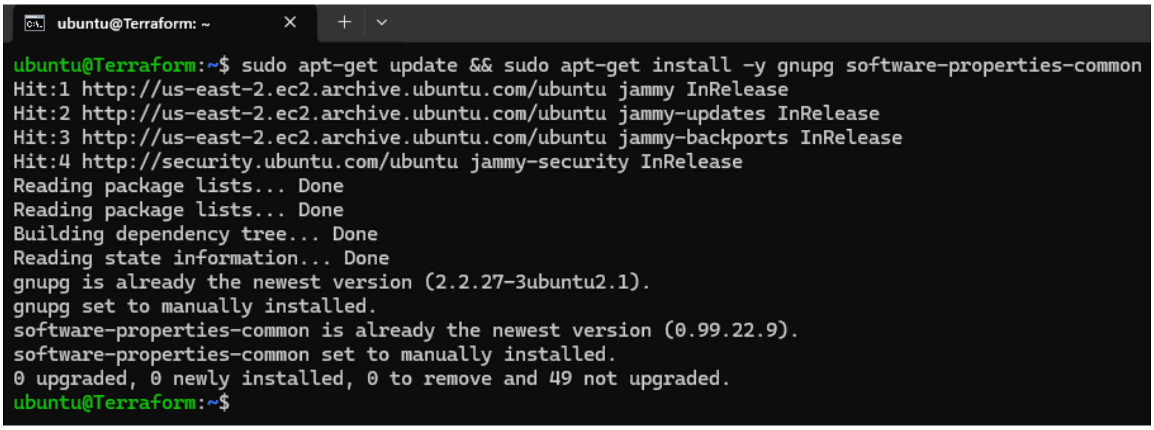
3. package updating



```
ubuntu@Terraform:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
```

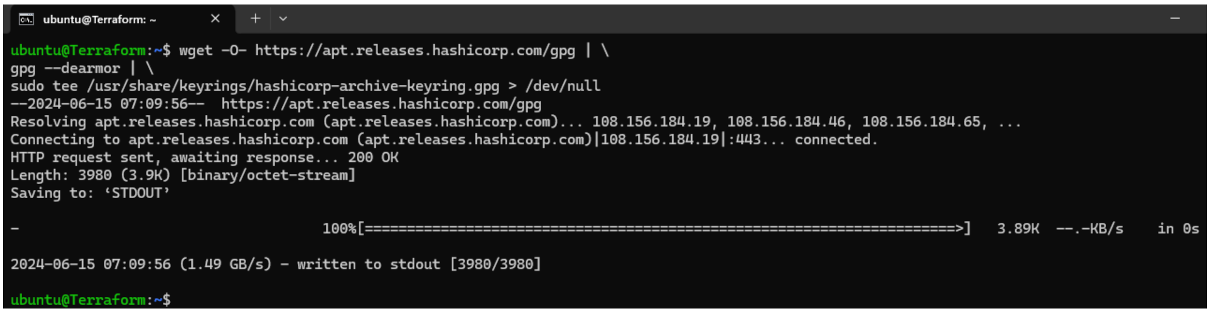
https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli

Ensure that your system is up to date and you have installed the gnupg, software-properties-common, and curl packages installed. You will use these packages to verify HashiCorp's GPG signature and install HashiCorp's Debian package repository.



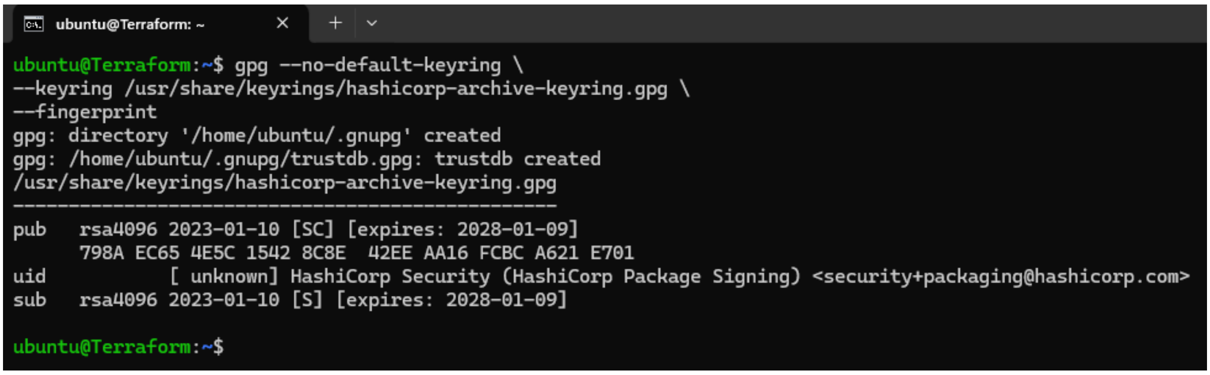
```
ubuntu@Terraform:~$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
ubuntu@Terraform:~$
```

Installing GPG gey



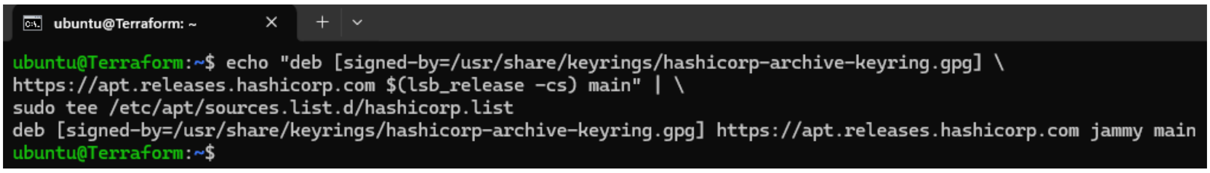
```
ubuntu@Terraform:~$ wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
--2024-06-15 07:09:56--  https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 108.156.184.19, 108.156.184.46, 108.156.184.65, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|108.156.184.19|: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

2024-06-15 07:09:56 (1.49 GB/s) - written to stdout [3980/3980]

ubuntu@Terraform:~$
```

Verify the keys fingerprint and it will report the key fingerprint



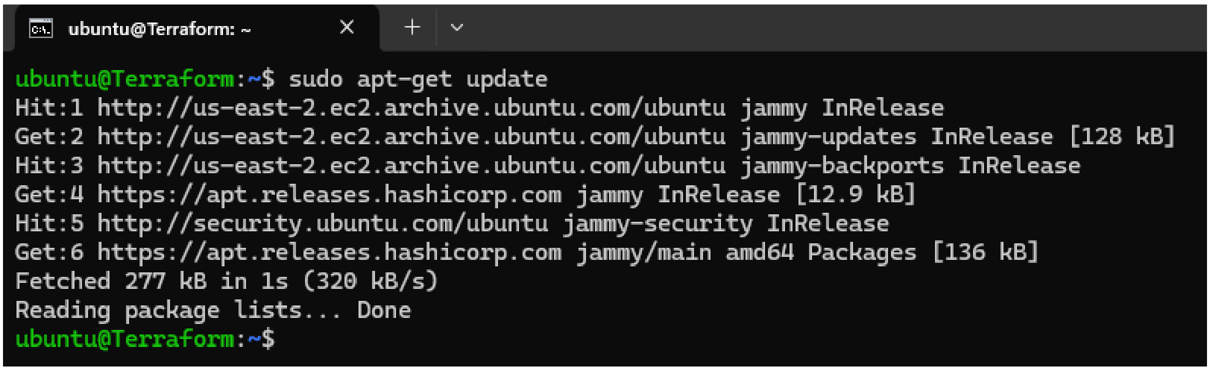
```
ubuntu@Terraform:~$ gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint
gpg: directory '/home/ubuntu/.gnupg' created
gpg: /home/ubuntu/.gnupg/trustdb.gpg: trustdb created
/usr/share/keyrings/hashicorp-archive-keyring.gpg
-------------------------------------------------
pub   rsa4096 2023-01-10 [SC] [expires: 2028-01-09]
      798A EC65 4E5C 1542 8C8E  42EE AA16 FCBC A621 E701
uid           [ unknown] HashiCorp Security (HashiCorp Package Signing) <security+packaging@hashicorp.com>
sub   rsa4096 2023-01-10 [S] [expires: 2028-01-09]

ubuntu@Terraform:~$
```

Add the official HashiCorp repository to your system. The lsb_release -cs command finds the distribution release codename for your current system, such as buster, groovy, or sid



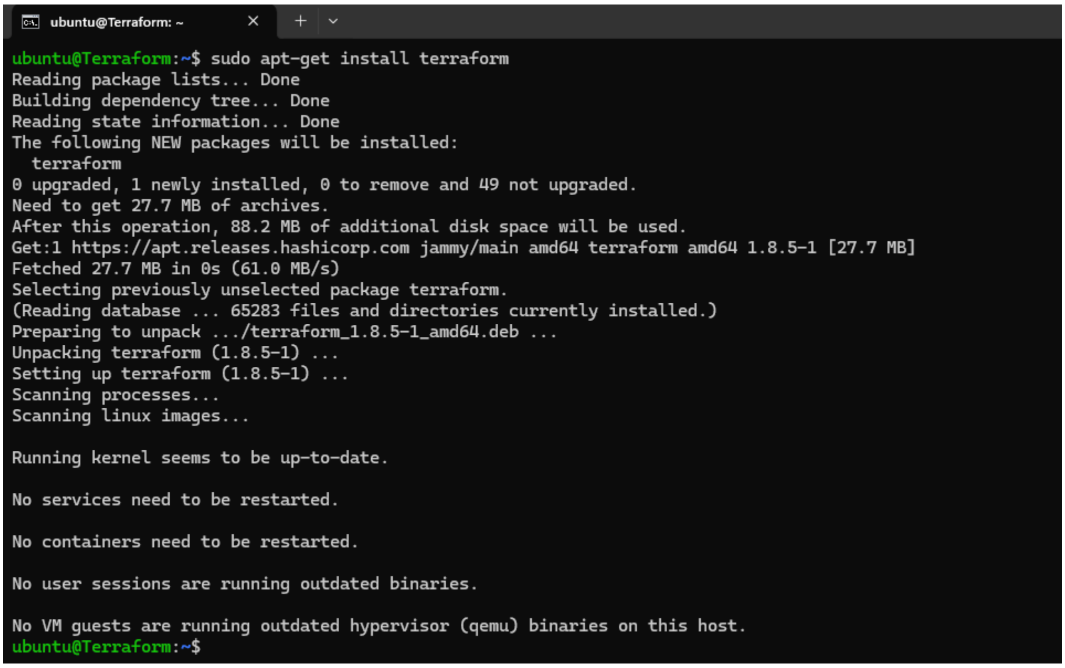
```
ubuntu@Terraform:~$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \
sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com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

Downloading the package information from hashicorp



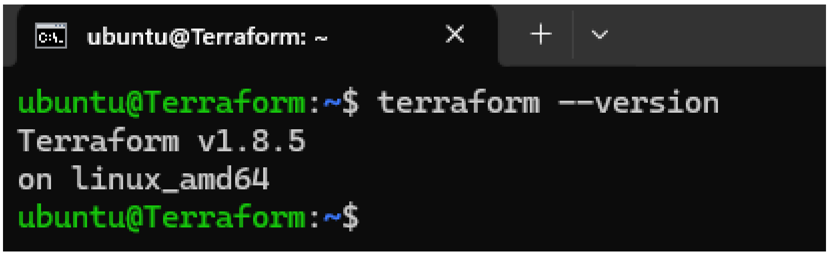
```
ubuntu@Terraform:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://apt.releases.hashicorp.com 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 [136 kB]
Fetched 277 kB in 1s (320 kB/s)
Reading package lists... Done
ubuntu@Terraform:~$
```

Install Terraform from the new repository.



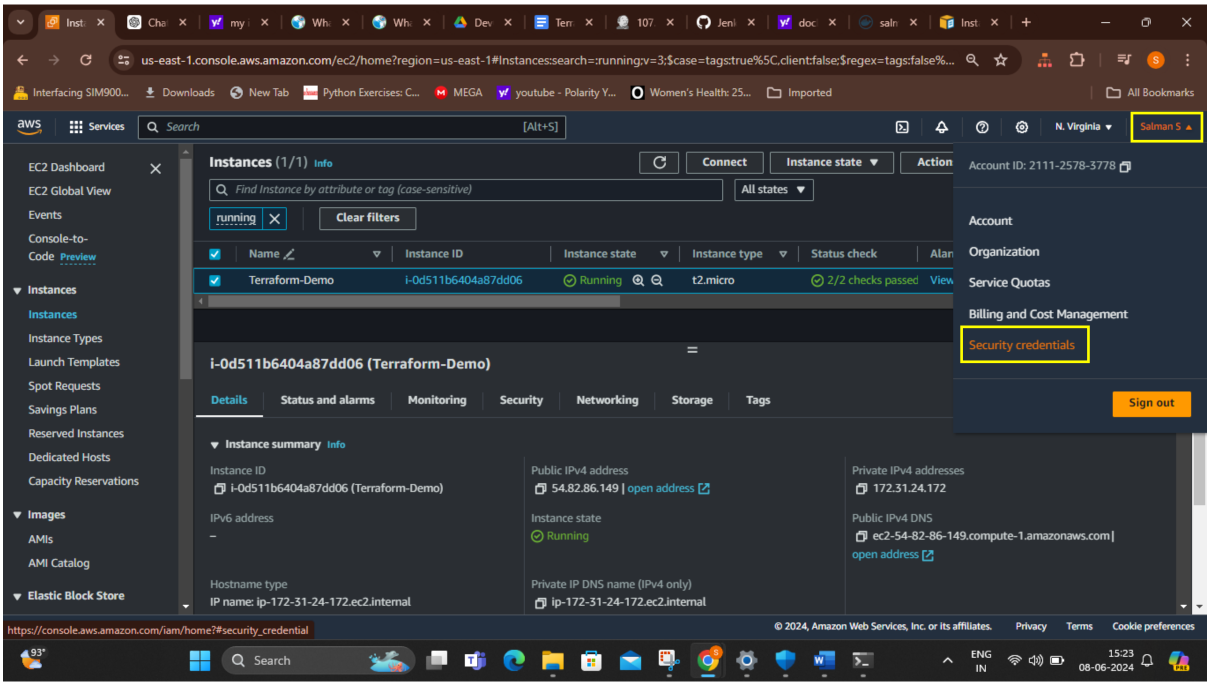
```
ubuntu@Terraform:~$ sudo apt-get install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 49 not upgraded.
Need to get 27.7 MB of archives.
After this operation, 88.2 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.8.5-1 [27.7 MB]
Fetched 27.7 MB in 0s (61.0 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 65283 files and directories currently installed.)
Preparing to unpack .../terraform_1.8.5-1_amd64.deb ...
Unpacking terraform (1.8.5-1) ...
Setting up terraform (1.8.5-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@Terraform:~$
```

Checking terraform --version



```
ubuntu@Terraform:~$ terraform --version
Terraform v1.8.5
on linux_amd64
ubuntu@Terraform:~$
```

Click on security credentials



Account ID: 2111-2578-3778

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check
Terraform-Demo	i-0d511b6404a87dd06	Running	t2.micro	2/2 checks passed

i-0d511b6404a87dd06 (Terraform-Demo)

Instance summary

Instance ID: i-0d511b6404a87dd06 (Terraform-Demo)

Public IPv4 address: 54.82.86.149 | open address

Private IPv4 addresses: 172.31.24.172

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-54-82-86-149.compute-1.amazonaws.com | open address

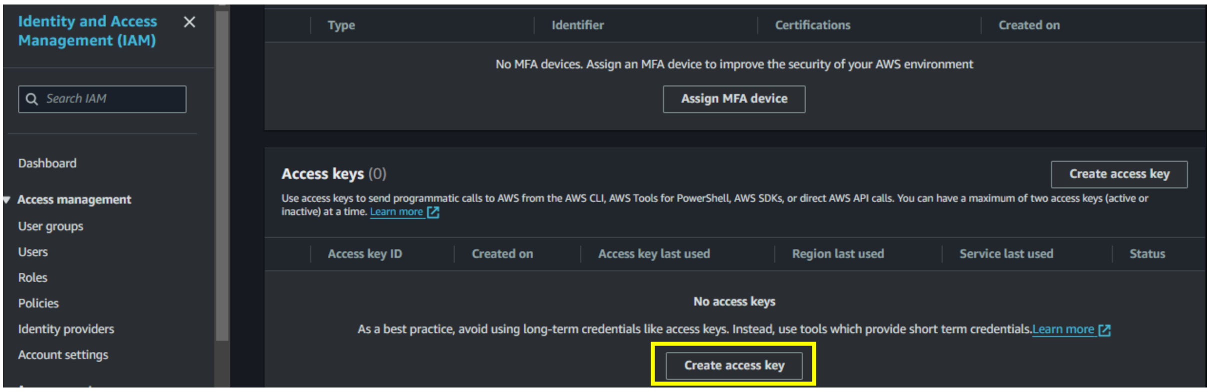
Hostname type: IP name: ip-172-31-24-172.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-24-172.ec2.internal

https://console.aws.amazon.com/iam/home?#security_credential

© 2024, Amazon Web Services, Inc. or its affiliates.

Generating Access key & secret access key



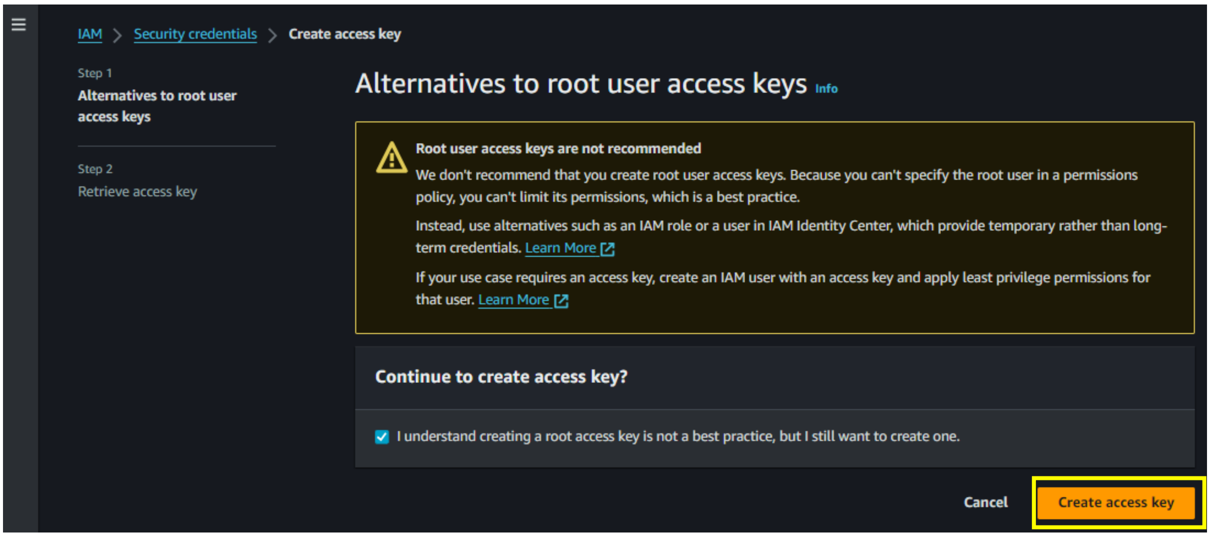
No MFA devices. Assign an MFA device to improve the security of your AWS environment

Access keys (0)

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

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


IAM > Security credentials > Create access key
Step 1
Alternatives to root user access keys
Step 2
Retrieve access key
Cancel
Create access key

Alternatives to root user access keys

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.

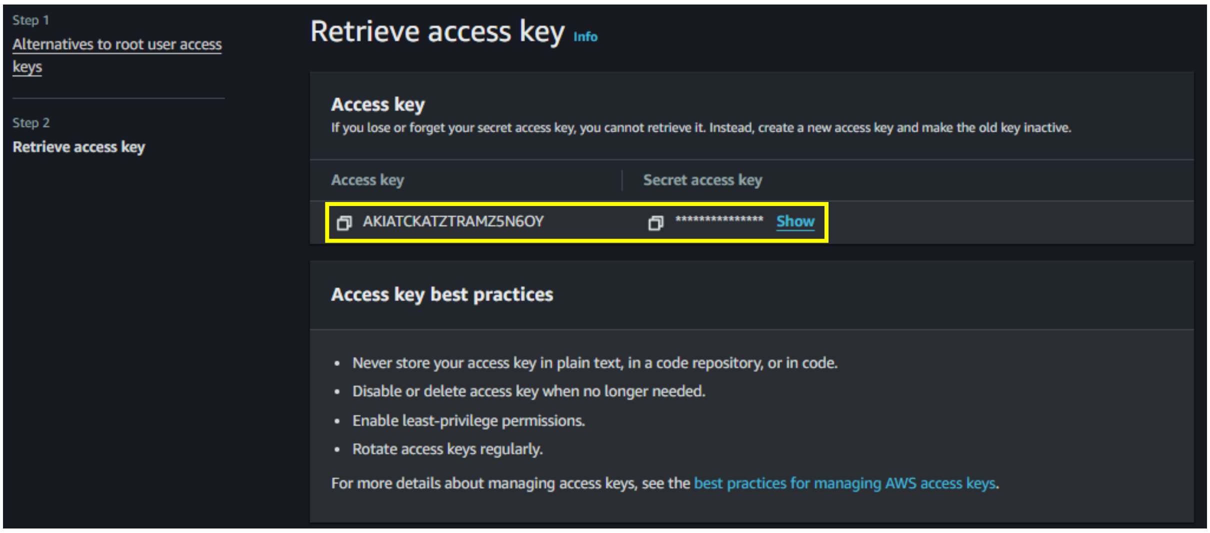
Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. Learn More

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.

Copy and save it.



Step 1

Alternatives to root user access keys

Step 2

Retrieve access key

Retrieve access key

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

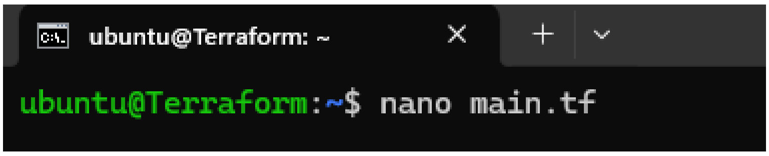
Access key	Secret access key
AKIATCKATZTRAMZ5N6OY	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

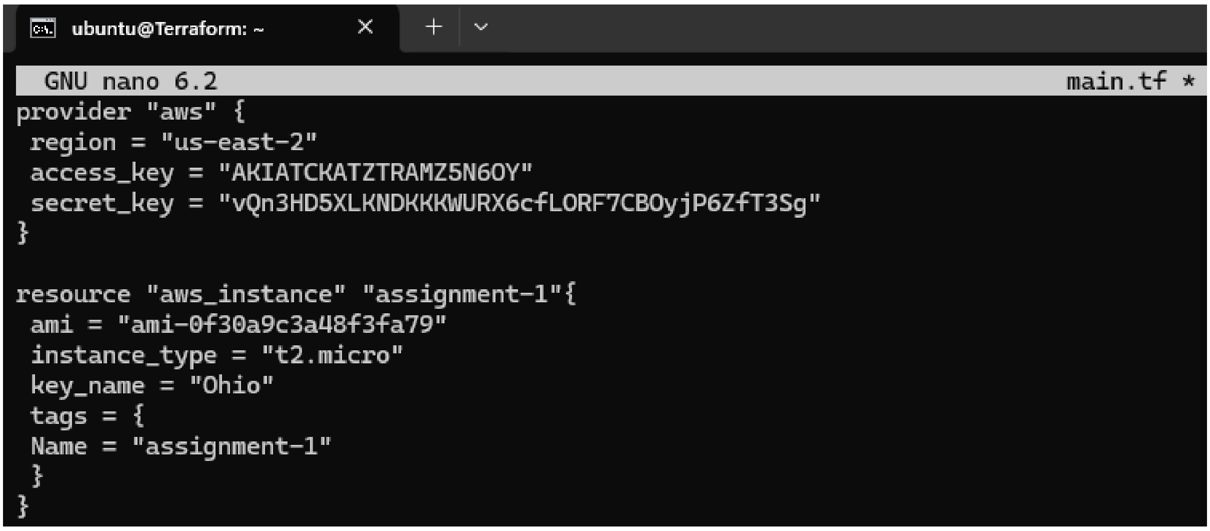
For more details about managing access keys, see the best practices for managing AWS access keys.

Nano main.tf and write the script to develop infrastructure



```
ubuntu@Terraform:~$ nano main.tf
```

Going install EC2 on AWS and with t2.micro and attached private key-pair



```
  GNU nano 6.2                                                            main.tf *
provider "aws" {
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

resource "aws_instance" "assignment-1"{
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 key_name = "Ohio"
 tags = {
 Name = "assignment-1"
 }
}
```

Script-1

provider "aws" {

 region = "us-east-2"

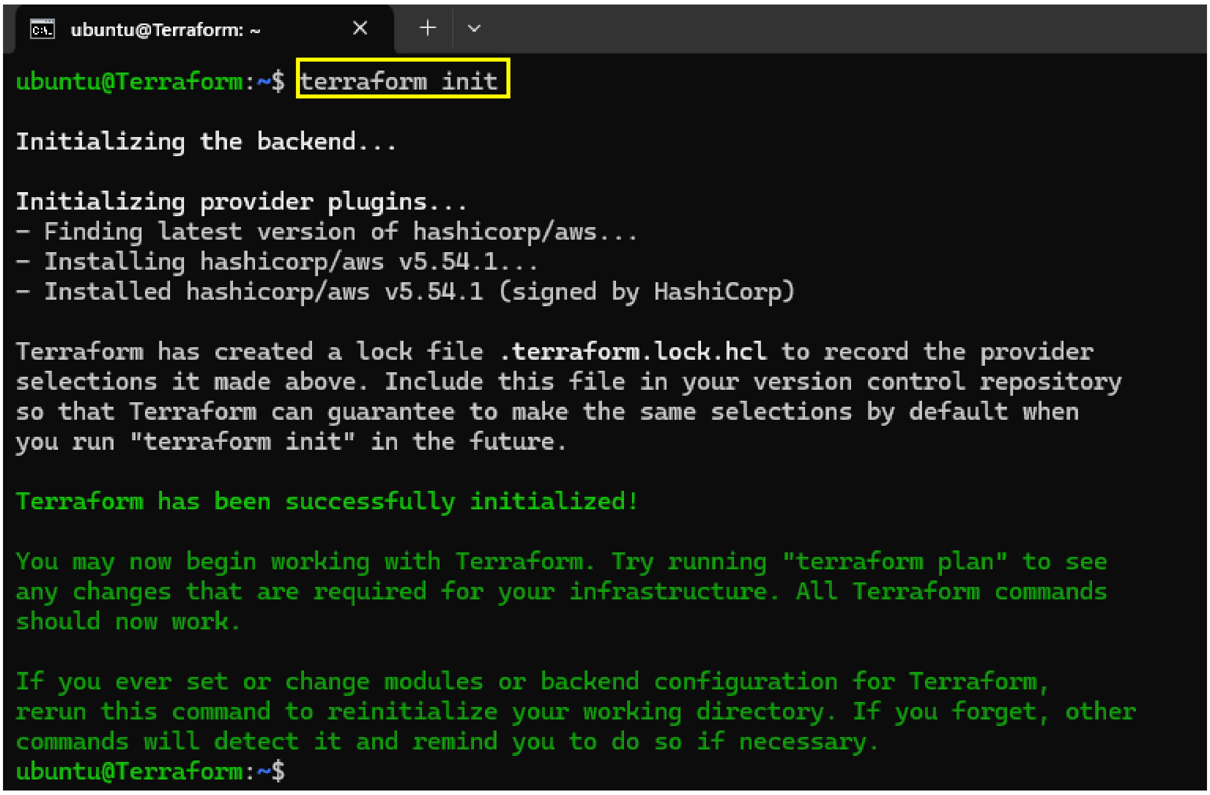
 access_key = "AKIATCKATZTRAMZ5N6OY"

 secret_key = "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"

}

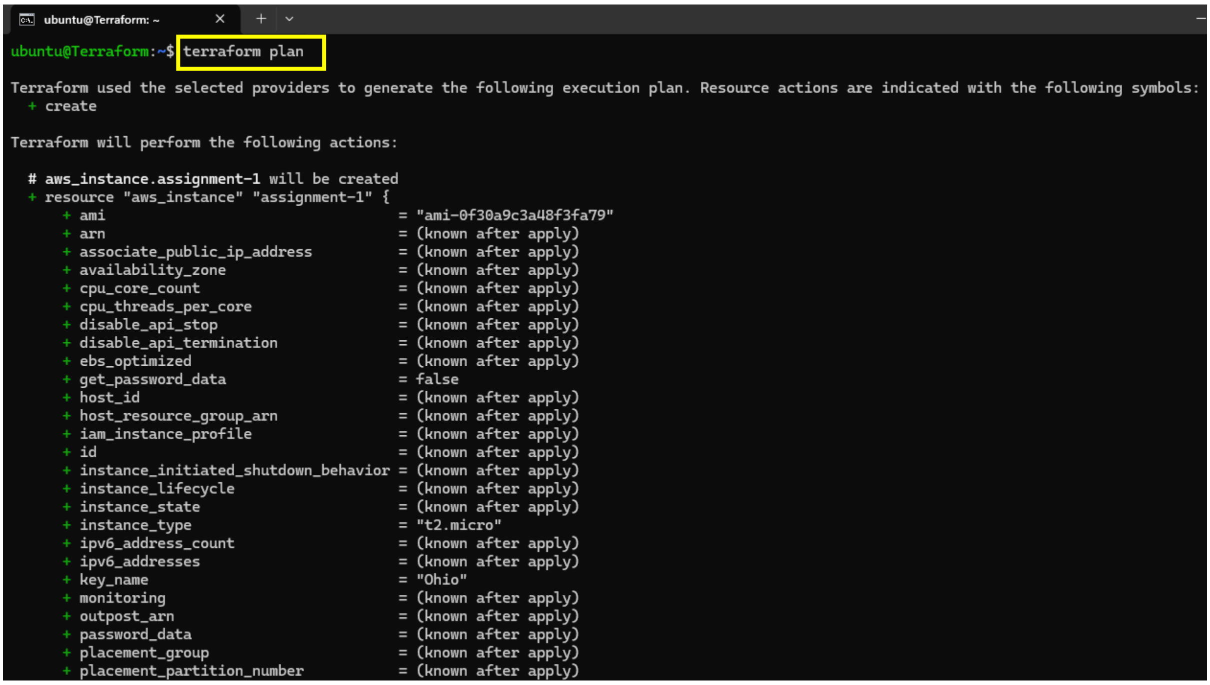
```
resource "aws_instance" "assignment-1"{
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 key_name = "Ohio"
 tags = {
 Name = "assignment-1"
 }
}
```

terraform init

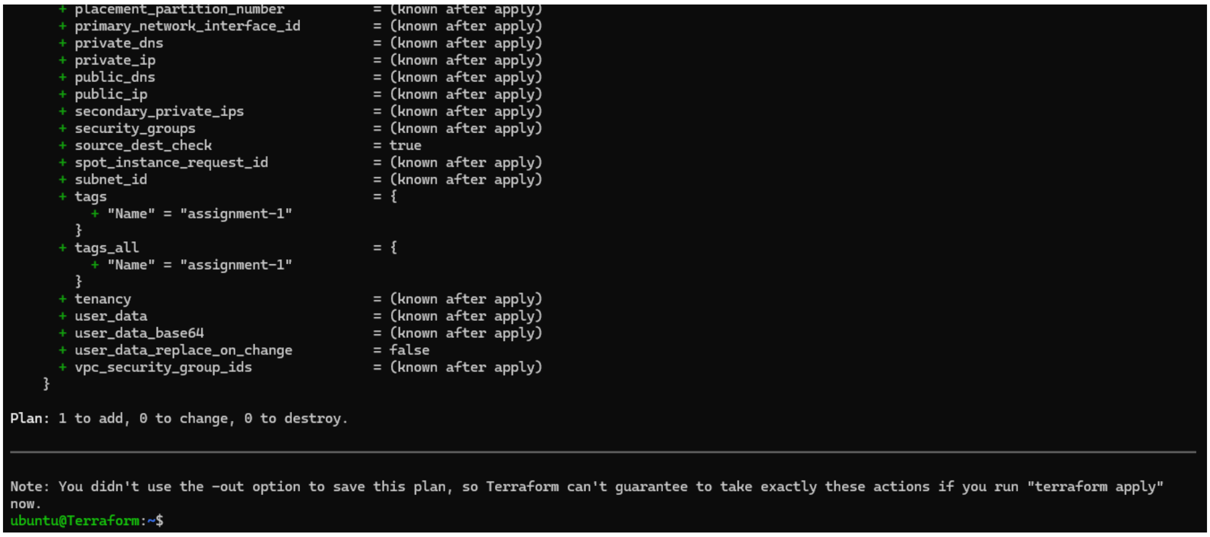


```
ubuntu@Terraform:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.54.1...
- Installed hashicorp/aws v5.54.1 (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:~$
```

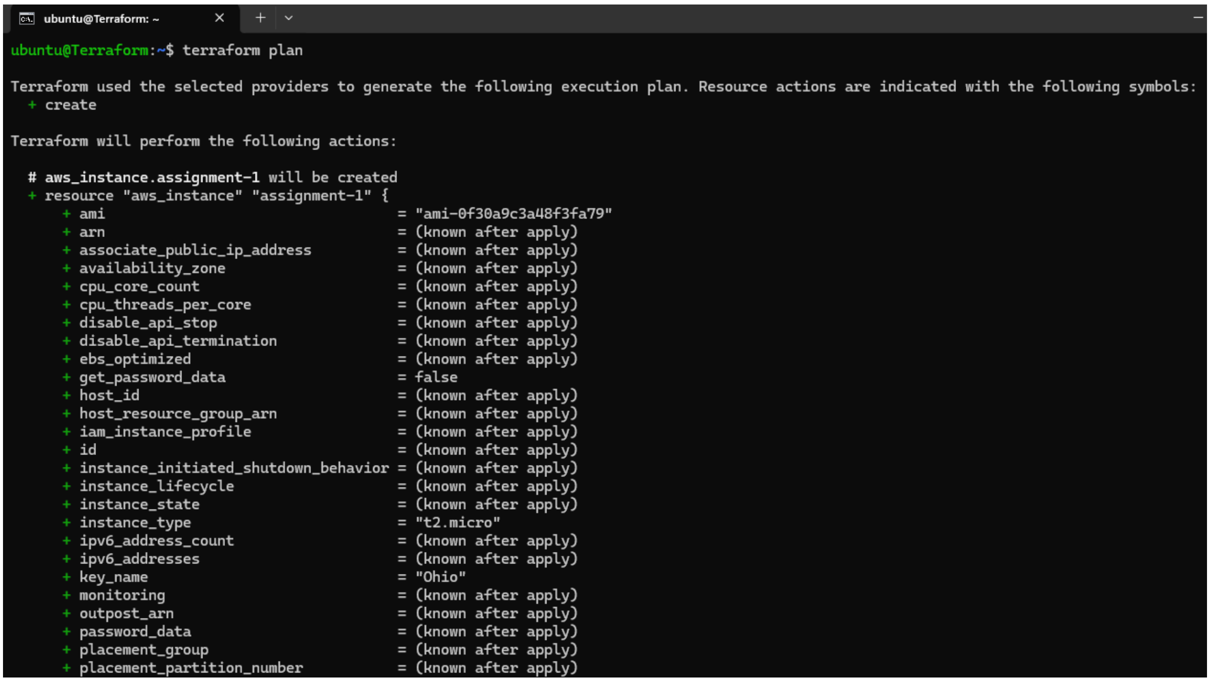
Terraform plan



```
ubuntu@Terraform:~$ 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.assignment-1 will be created
  + resource "aws_instance" "assignment-1" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
```

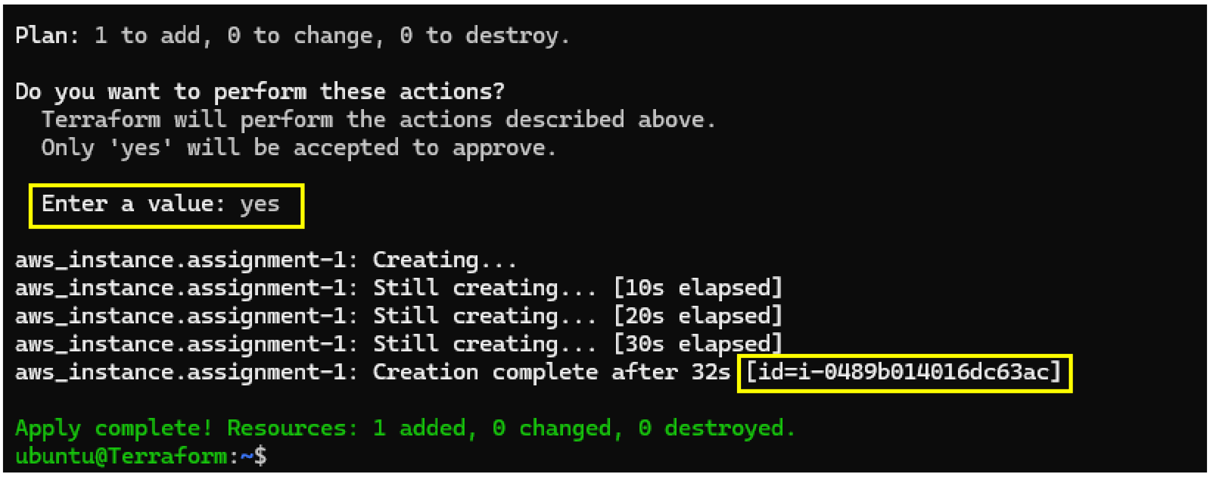


```
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "assignment-1"
        }
      + tags_all                             = {
          + "Name" = "assignment-1"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (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.
ubuntu@Terraform:~$
```



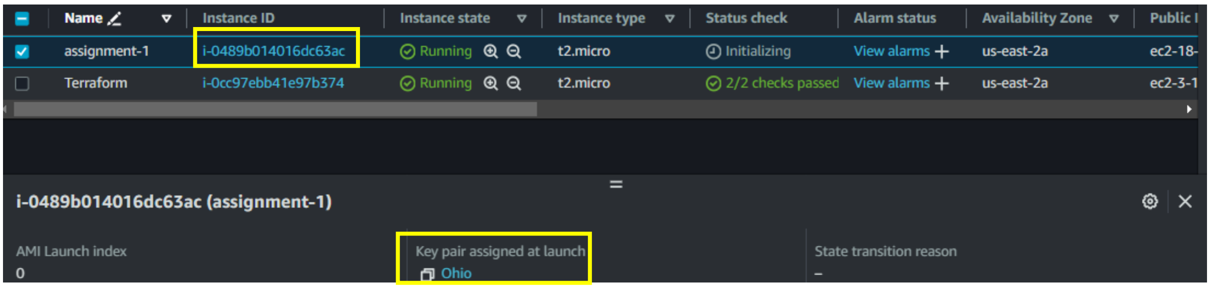
```
ubuntu@Terraform:~$ terraform plan

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.assignment-1 will be created
  + resource "aws_instance" "assignment-1" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
```

Terraform 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.assignment-1: Creating...
aws_instance.assignment-1: Still creating... [10s elapsed]
aws_instance.assignment-1: Still creating... [20s elapsed]
aws_instance.assignment-1: Still creating... [30s elapsed]
aws_instance.assignment-1: Creation complete after 32s [id=i-0489b014016dc63ac]

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

Launched successfully.



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
☑	assignment-1	i-0489b014016dc63ac	Running	t2.micro	Initializing	View alarms +	us-east-2a	ec2-18-
☐	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	ec2-3-1

i-0489b014016dc63ac (assignment-1)

AMI Launch index
0

Key pair assigned at launch
Ohio

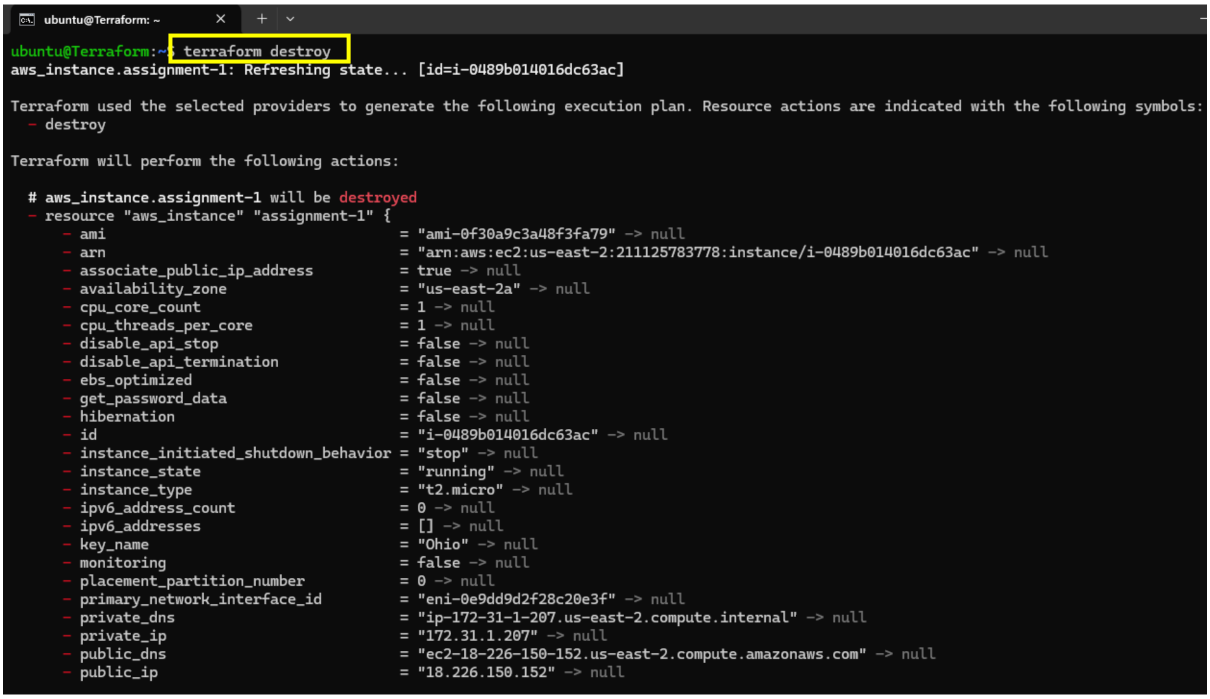
State transition reason
–

Terraform Assignment - 2

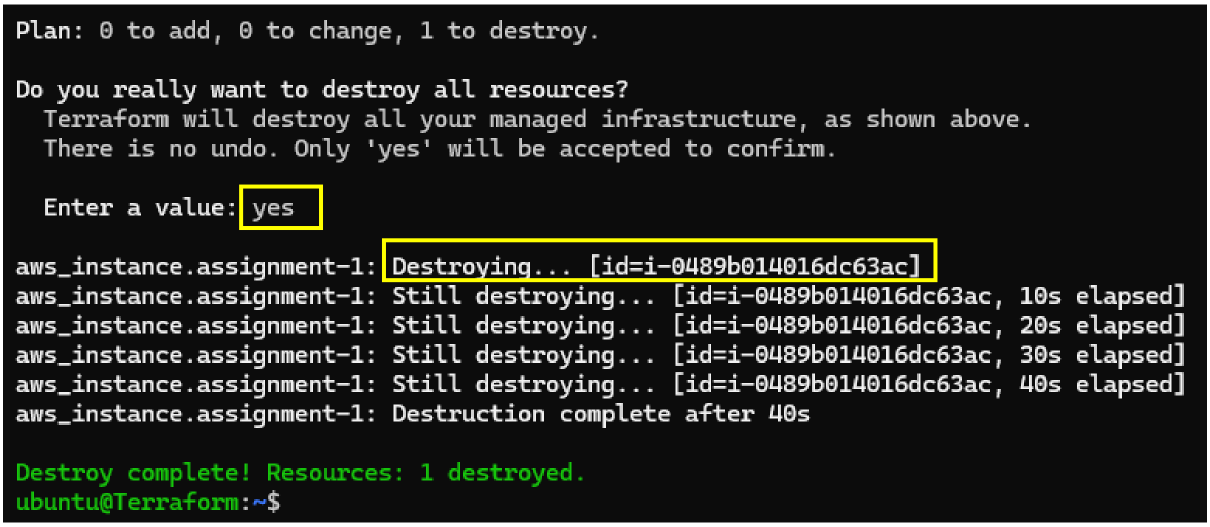
You have been asked to:

- Destroy the previous deployment
- Create a new EC2 instance with an Elastic IP

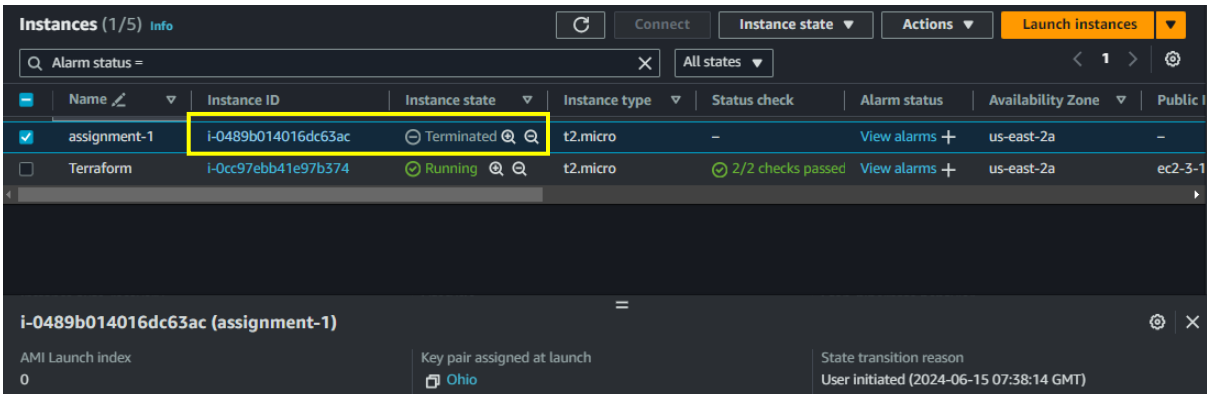
Destroy previous deployment



```
ubuntu@Terraform:~$ terraform destroy
aws_instance.assignment-1: Refreshing state... [id=i-0489b014016dc63ac]

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:

  # aws_instance.assignment-1 will be destroyed
  - resource "aws_instance" "assignment-1" {
      - ami                                  = "ami-0f30a9c3a48f3fa79" -> null
      - arn                                  = "arn:aws:ec2:us-east-2:211125783778:instance/i-0489b014016dc63ac" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-2a" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-0489b014016dc63ac" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
      - instance_type                        = "t2.micro" -> null
      - ipv6_address_count                   = 0 -> null
      - ipv6_addresses                       = [] -> null
      - key_name                             = "Ohio" -> null
      - monitoring                           = false -> null
      - placement_partition_number           = 0 -> null
      - primary_network_interface_id         = "eni-0e9dd9d2f28c20e3f" -> null
      - private_dns                          = "ip-172-31-1-207.us-east-2.compute.internal" -> null
      - private_ip                           = "172.31.1.207" -> null
      - public_dns                           = "ec2-18-226-150-152.us-east-2.compute.amazonaws.com" -> null
      - public_ip                            = "18.226.150.152" -> null
```



```
Plan: 0 to add, 0 to change, 1 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.assignment-1: Destroying... [id=i-0489b014016dc63ac]
aws_instance.assignment-1: Still destroying... [id=i-0489b014016dc63ac, 10s elapsed]
aws_instance.assignment-1: Still destroying... [id=i-0489b014016dc63ac, 20s elapsed]
aws_instance.assignment-1: Still destroying... [id=i-0489b014016dc63ac, 30s elapsed]
aws_instance.assignment-1: Still destroying... [id=i-0489b014016dc63ac, 40s elapsed]
aws_instance.assignment-1: Destruction complete after 40s

Destroy complete! Resources: 1 destroyed.
ubuntu@Terraform:~$
```

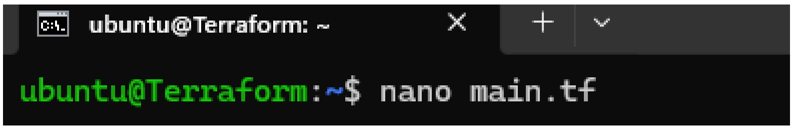


☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
☑	assignment-1	i-0489b014016dc63ac	Terminated	t2.micro	–	View alarms +	us-east-2a	–
☐	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	ec2-3-1

i-0489b014016dc63ac (assignment-1)

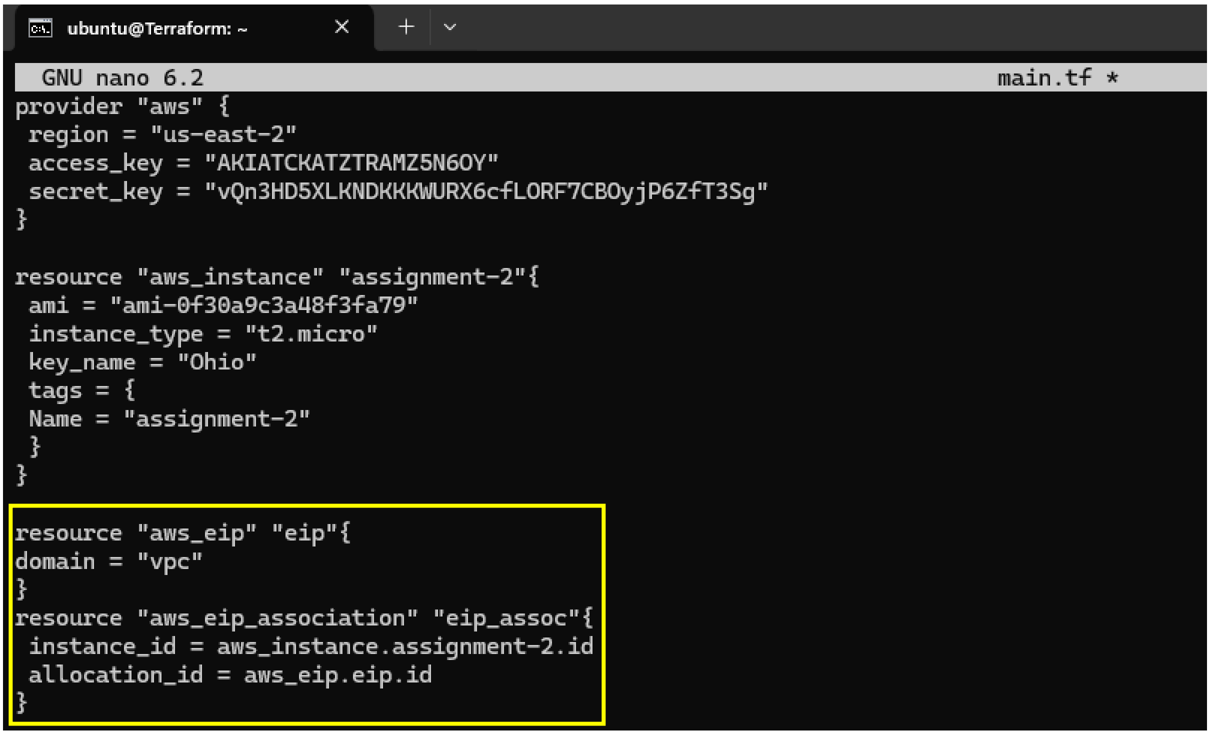
AMI Launch index	Key pair assigned at launch	State transition reason
0	Ohio	User initiated (2024-06-15 07:38:14 GMT)

Now write terraform script as per 2nd assignment



```
ubuntu@Terraform:~$ nano main.tf
```

Creating a New instance with Elastic Ip(Public-IP)



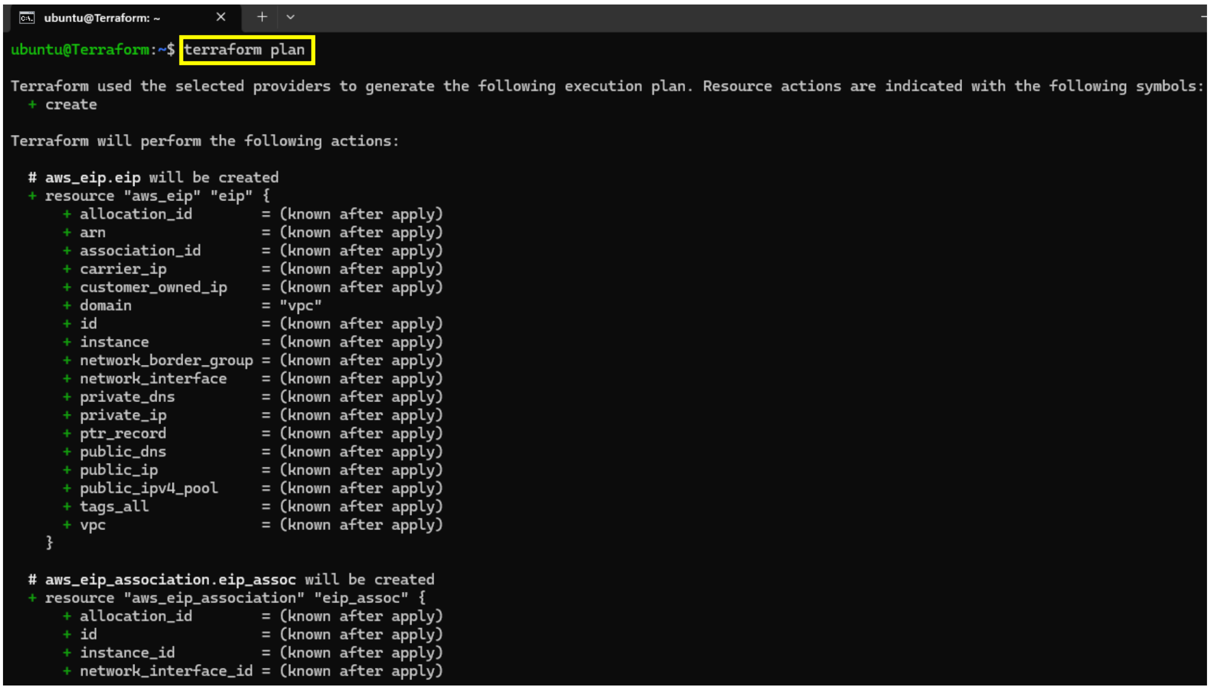
```
 GNU nano 6.2                                                          main.tf *
provider "aws" {
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

resource "aws_instance" "assignment-2"{
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 key_name = "Ohio"
 tags = {
 Name = "assignment-2"
 }
}

resource "aws_eip" "eip"{
domain = "vpc"
}
resource "aws_eip_association" "eip_assoc"{
 instance_id = aws_instance.assignment-2.id
 allocation_id = aws_eip.eip.id
}
```

Script-2t

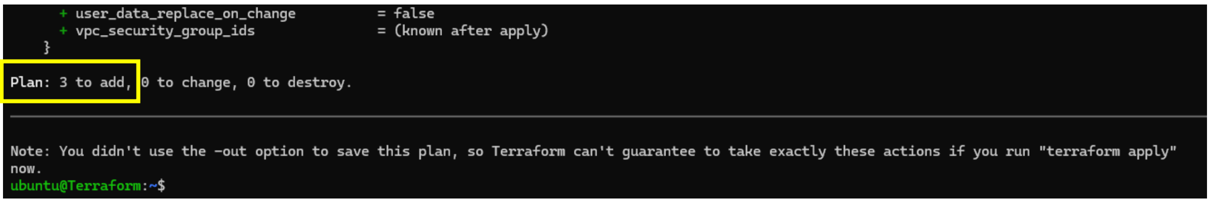
```
provider "aws" {
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}
```

```
resource "aws_instance" "assignment-2"{
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 key_name = "Ohio"
 tags = {
 Name = "assignment-2"
 }
}


resource "aws_eip" "eip"{
domain = "vpc"
}
resource "aws_eip_association" "eip_assoc"{
 instance_id = aws_instance.assignment-2.id
 allocation_id = aws_eip.eip.id
}
```

Terraform plan

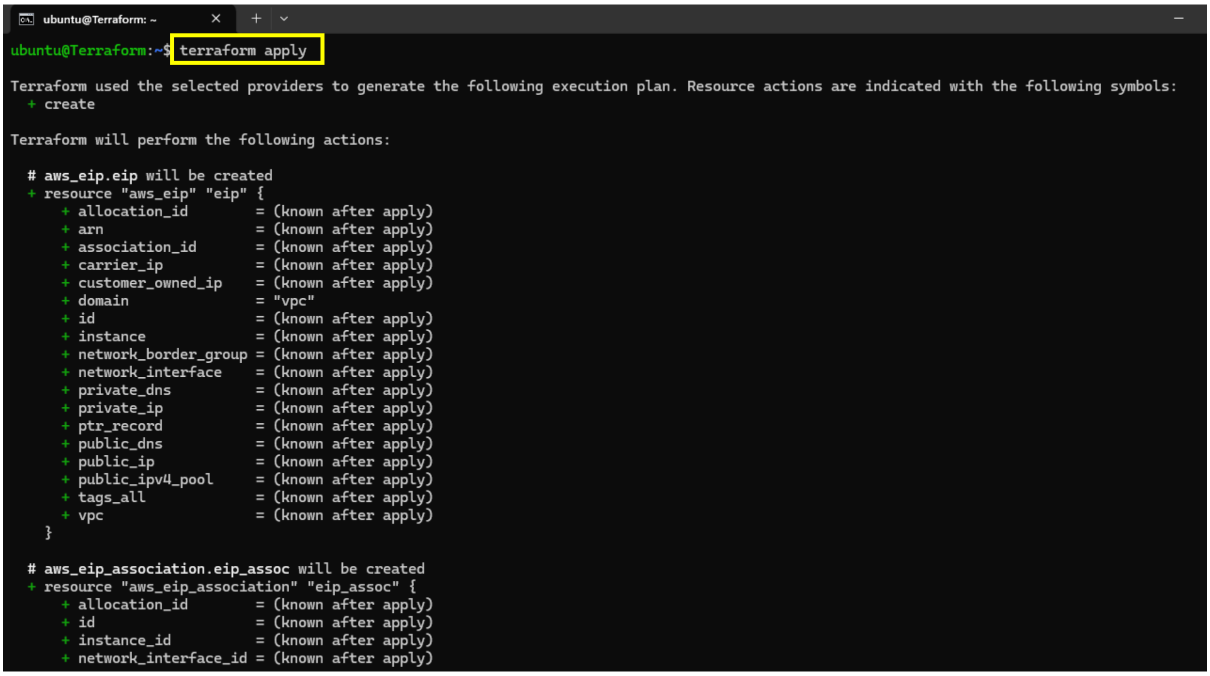


```
ubuntu@Terraform:~$ terraform plan

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.eip will be created
  + resource "aws_eip" "eip" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + 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_all             = (known after apply)
      + vpc                  = (known after apply)
    }

  # aws_eip_association.eip_assoc will be created
  + resource "aws_eip_association" "eip_assoc" {
      + allocation_id        = (known after apply)
      + id                   = (known after apply)
      + instance_id          = (known after apply)
      + network_interface_id = (known after apply)
```



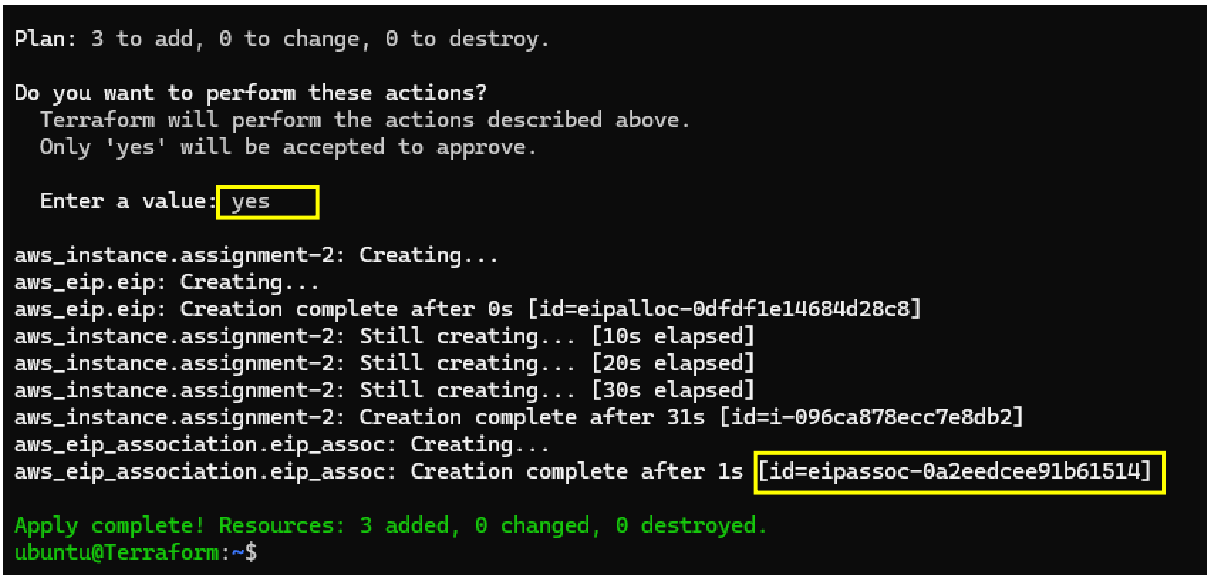
```
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 3 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:~$
```

Terraform apply



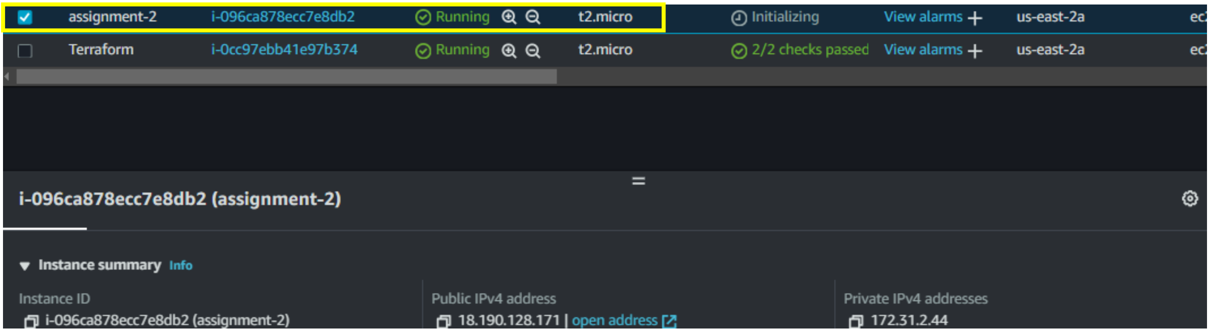
```
ubuntu@Terraform:~$ terraform apply

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.eip will be created
  + resource "aws_eip" "eip" {
      + allocation_id        = (known after apply)
      + arn                  = (known after apply)
      + 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_all             = (known after apply)
      + vpc                  = (known after apply)
    }

  # aws_eip_association.eip_assoc will be created
  + resource "aws_eip_association" "eip_assoc" {
      + allocation_id        = (known after apply)
      + id                   = (known after apply)
      + instance_id          = (known after apply)
      + network_interface_id = (known after apply)
```

Created the resources



```
Plan: 3 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.assignment-2: Creating...
aws_eip.eip: Creating...
aws_eip.eip: Creation complete after 0s [id=eipalloc-0dfdf1e14684d28c8]
aws_instance.assignment-2: Still creating... [10s elapsed]
aws_instance.assignment-2: Still creating... [20s elapsed]
aws_instance.assignment-2: Still creating... [30s elapsed]
aws_instance.assignment-2: Creation complete after 31s [id=i-096ca878ecc7e8db2]
aws_eip_association.eip_assoc: Creating...
aws_eip_association.eip_assoc: Creation complete after 1s [id=eipassoc-0a2eedcee91b61514]

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

We can see assignment-2 is running



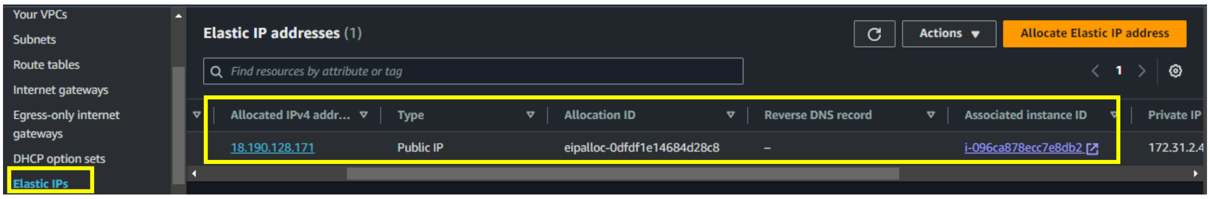
☑	assignment-2	i-096ca878ecc7e8db2	Running	t2.micro	Initializing	View alarms +	us-east-2a	ec
□	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	ec

i-096ca878ecc7e8db2 (assignment-2)

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-096ca878ecc7e8db2 (assignment-2)	18.190.128.171 | open address	172.31.2.44

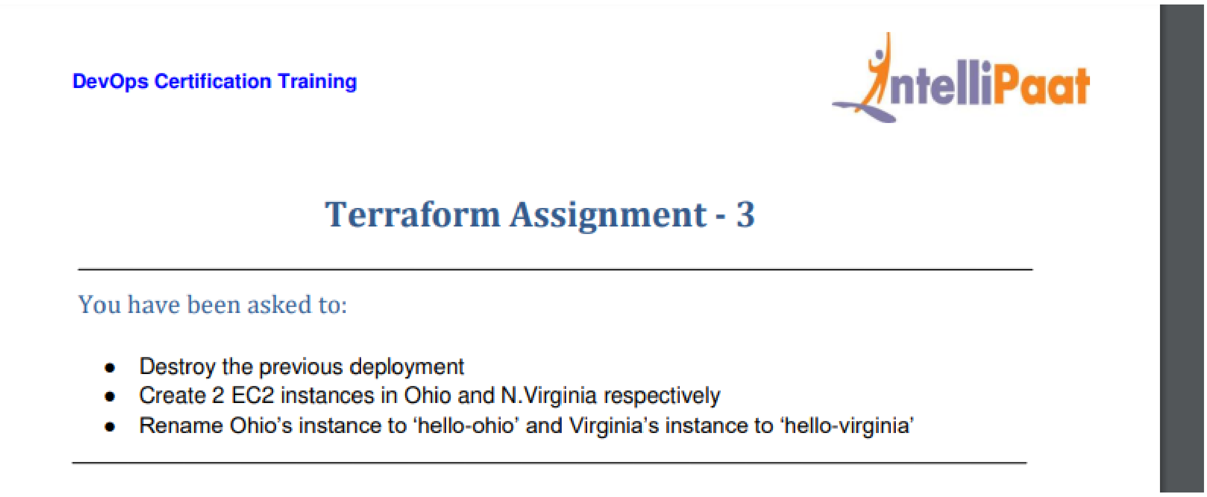
And its allotted public Ip too



Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Actions
Allocate Elastic IP address
Find resources by attribute or tag

Elastic IP addresses (1)

Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP
18.190.128.171	Public IP	eipalloc-0dfdf1e14684d28c8	–	i-096ca878ecc7e8db2	172.31.2.4



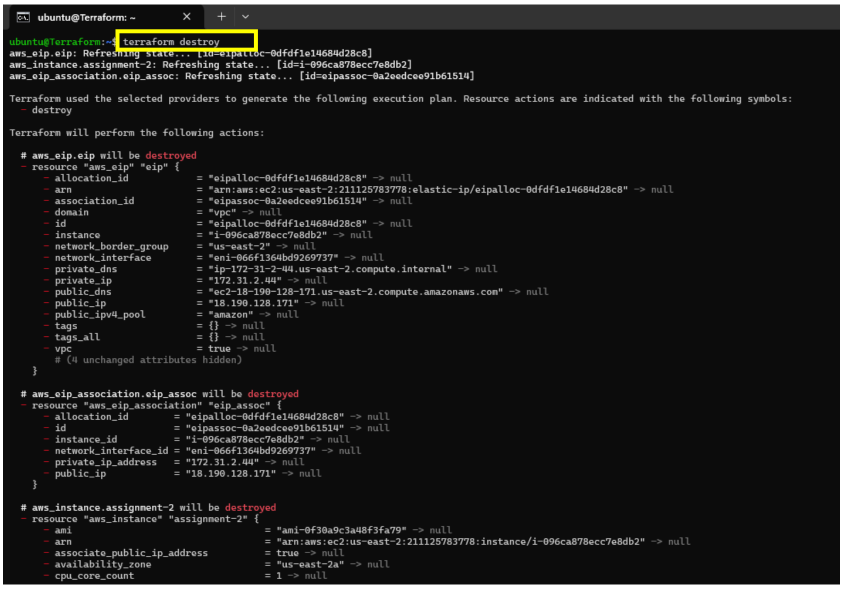
DevOps Certification Training

Terraform Assignment - 3

You have been asked to:

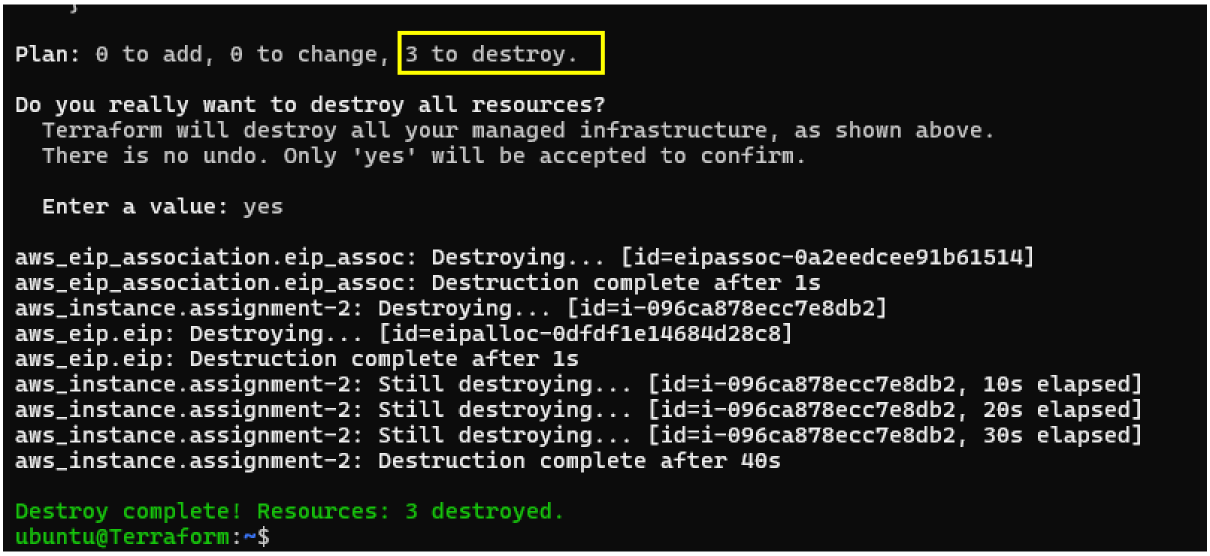
- Destroy the previous deployment
- Create 2 EC2 instances in Ohio and N.Virginia respectively
- Rename Ohio's instance to 'hello-ohio' and Virginia's instance to 'hello-virginia'

Destroying the previous deployment



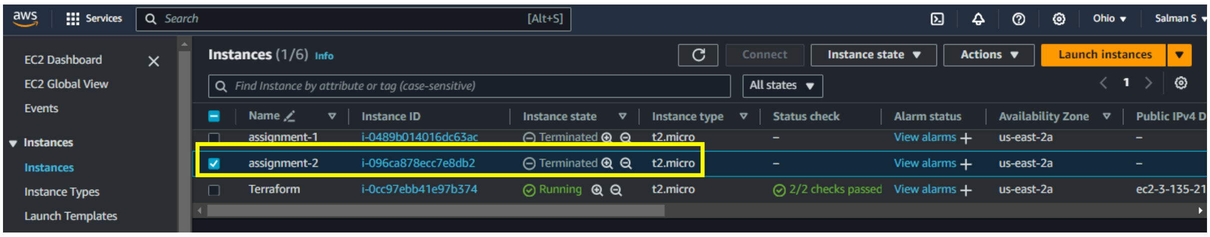
```
ubuntu@Terraform:~$ terraform destroy
aws_eip.eip: Refreshing state... [id=eipalloc-0dfdf1e14684d28c8]
aws_instance.assignment-2: Refreshing state... [id=i-096ca878ecc7e8db2]
aws_eip_association.eip_assoc: Refreshing state... [id=eipassoc-0a2eedcee91b61514]

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:

  # aws_eip.eip will be destroyed
  - resource "aws_eip" "eip" {
      - allocation_id            = "eipalloc-0dfdf1e14684d28c8" -> null
      - arn                      = "arn:aws:ec2:us-east-2:211125783778:elastic-ip/eipalloc-0dfdf1e14684d28c8" -> null
      - association_id           = "eipassoc-0a2eedcee91b61514" -> null
      - domain                   = "vpc" -> null
      - id                       = "eipalloc-0dfdf1e14684d28c8" -> null
      - instance                 = "i-096ca878ecc7e8db2" -> null
      - network_border_group     = "us-east-2" -> null
      - network_interface        = "eni-066f1364bd9269737" -> null
      - private_dns              = "ip-172-31-2-44.us-east-2.compute.internal" -> null
      - private_ip               = "172.31.2.44" -> null
      - public_dns               = "ec2-18-190-128-171.us-east-2.compute.amazonaws.com" -> null
      - public_ip                = "18.190.128.171" -> null
      - public_ipv4_pool         = "amazon" -> null
      - tags                     = {} -> null
      - tags_all                 = {} -> null
      - vpc                      = true -> null
        # (4 unchanged attributes hidden)
    }

  # aws_eip_association.eip_assoc will be destroyed
  - resource "aws_eip_association" "eip_assoc" {
      - allocation_id        = "eipalloc-0dfdf1e14684d28c8" -> null
      - id                   = "eipassoc-0a2eedcee91b61514" -> null
      - instance_id          = "i-096ca878ecc7e8db2" -> null
      - network_interface_id = "eni-066f1364bd9269737" -> null
      - private_ip_address   = "172.31.2.44" -> null
      - public_ip            = "18.190.128.171" -> null
    }

  # aws_instance.assignment-2 will be destroyed
  - resource "aws_instance" "assignment-2" {
      - ami                                  = "ami-0f30a9c3a48f3fa79" -> null
      - arn                                  = "arn:aws:ec2:us-east-2:211125783778:instance/i-096ca878ecc7e8db2" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-2a" -> null
      - cpu_core_count                       = 1 -> null
```

Its destroyed the resources



```
Plan: 0 to add, 0 to change, 3 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_association.eip_assoc: Destroying... [id=eipassoc-0a2eedcee91b61514]
aws_eip_association.eip_assoc: Destruction complete after 1s
aws_instance.assignment-2: Destroying... [id=i-096ca878ecc7e8db2]
aws_eip.eip: Destroying... [id=eipalloc-0dfdf1e14684d28c8]
aws_eip.eip: Destruction complete after 1s
aws_instance.assignment-2: Still destroying... [id=i-096ca878ecc7e8db2, 10s elapsed]
aws_instance.assignment-2: Still destroying... [id=i-096ca878ecc7e8db2, 20s elapsed]
aws_instance.assignment-2: Still destroying... [id=i-096ca878ecc7e8db2, 30s elapsed]
aws_instance.assignment-2: Destruction complete after 40s

Destroy complete! Resources: 3 destroyed.
ubuntu@Terraform:~$
```

We can see it is terminated

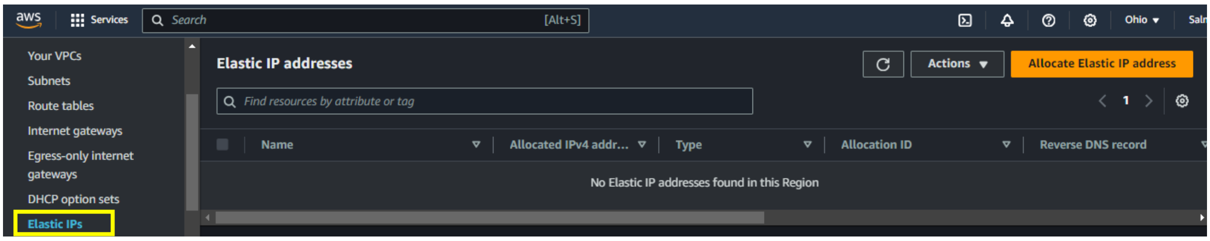


Instances (1/6) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
□	assignment-1	i-0489b014016dc63ac	Terminated	t2.micro	–	View alarms +	us-east-2a	–
☑	assignment-2	i-096ca878ecc7e8db2	Terminated	t2.micro	–	View alarms +	us-east-2a	–
□	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a	ec2-3-135-21

And also deleted elastic Ip



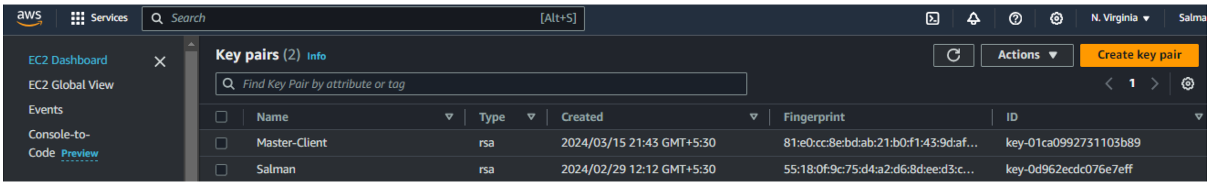
Elastic IP addresses

Find resources by attribute or tag

	Name	Allocated IPv4 addr...	Type	Allocation ID	Reverse DNS record

No Elastic IP addresses found in this Region

Remembering the key on N.virginia region

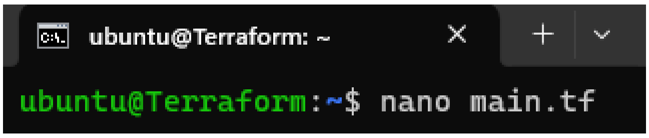


Key pairs (2) Info

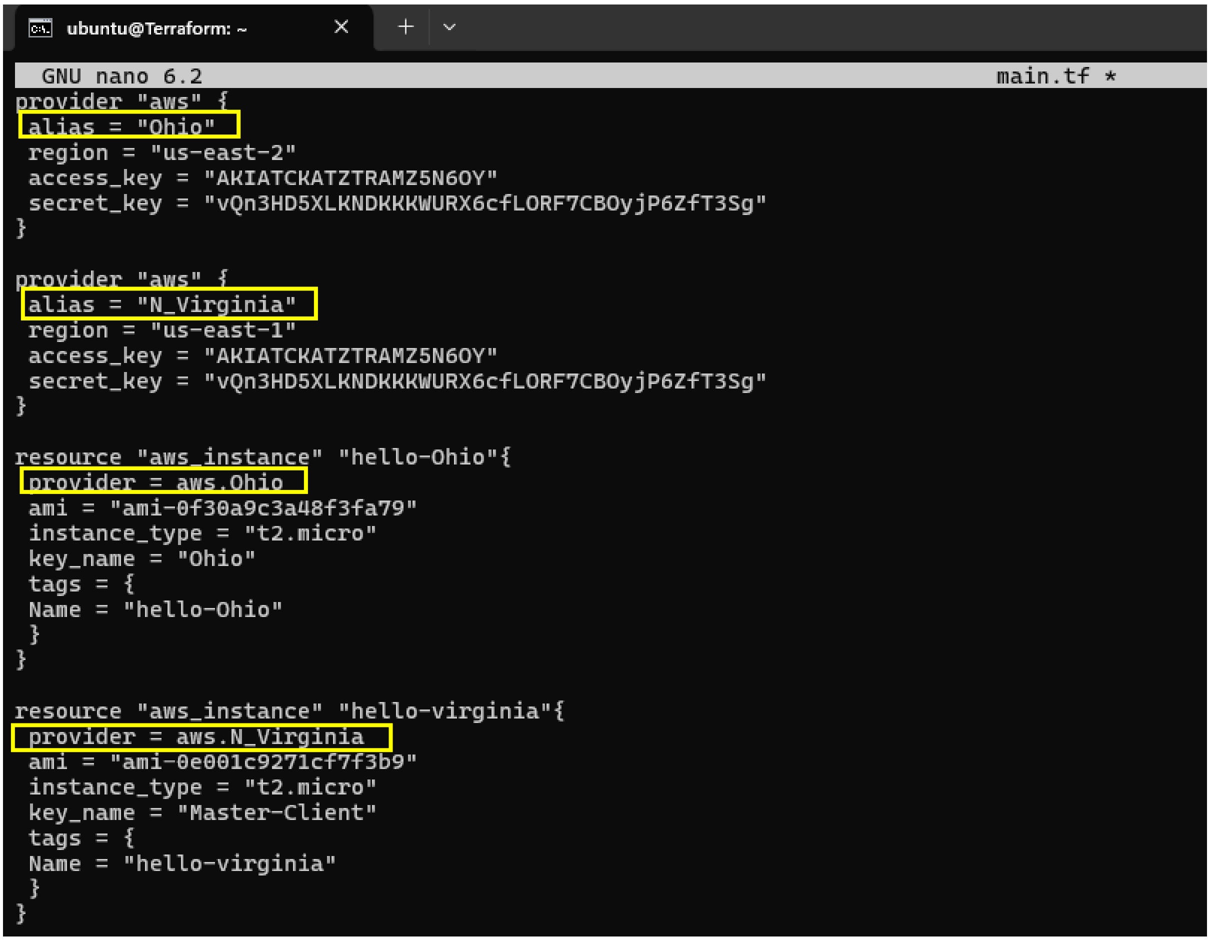
Find Key Pair by attribute or tag

	Name	Type	Created	Fingerprint	ID
□	Master-Client	rsa	2024/03/15 21:43 GMT+5:30	81:e0:cc:8e:bd:ab:21:b0:f1:43:9d:af...	key-01ca0992731103b89
□	Salman	rsa	2024/02/29 12:12 GMT+5:30	55:18:0f:9c:75:d4:a2:d6:8d:ee:d3:c...	key-0d962ecdc076e7eff

Now write terraform script as per 3rd assignment



```
ubuntu@Terraform:~$ nano main.tf
```



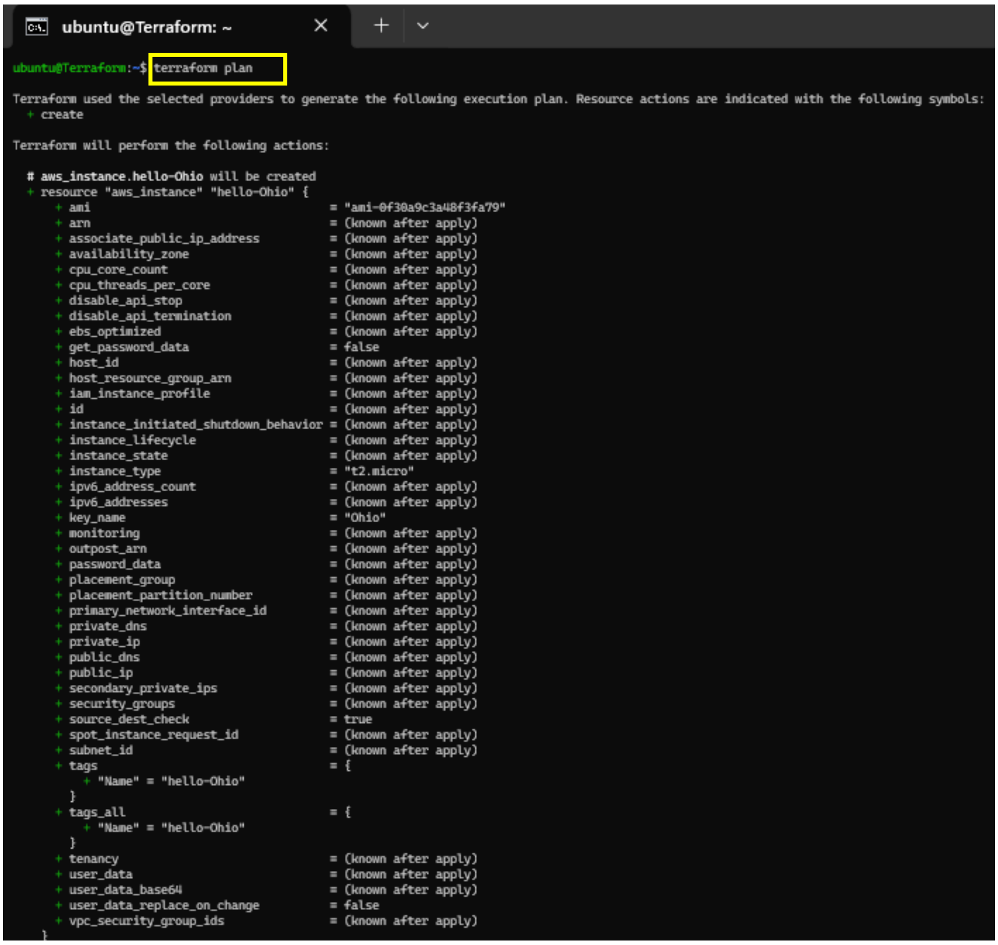
```
  GNU nano 6.2                                                        main.tf *
provider "aws" {
 alias = "Ohio"
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

provider "aws" {
 alias = "N_Virginia"
 region = "us-east-1"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

resource "aws_instance" "hello-Ohio"{
 provider = aws.Ohio
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 key_name = "Ohio"
 tags = {
 Name = "hello-Ohio"
 }
}

resource "aws_instance" "hello-virginia"{
 provider = aws.N_Virginia
 ami = "ami-0e001c9271cf7f3b9"
 instance_type = "t2.micro"
 key_name = "Master-Client"
 tags = {
 Name = "hello-virginia"
 }
}
```

Script-3

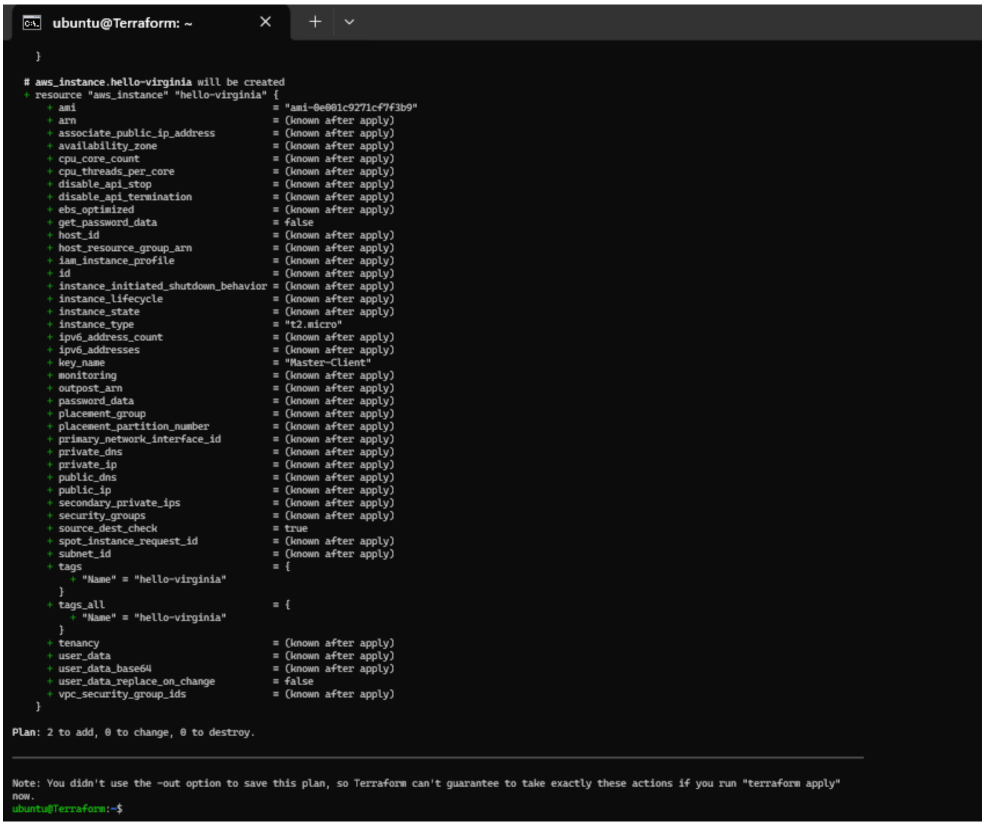
```
provider "aws" {
 alias = "Ohio"
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

provider "aws" {
 alias = "N_Virginia"
 region = "us-east-1"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}
```

```
resource "aws_instance" "hello-Ohio"{
 provider = aws.Ohio
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 key_name = "Ohio"
 tags = {
 Name = "hello-Ohio"
 }
}

resource "aws_instance" "hello-virginia"{
 provider = aws.N_Virginia
 ami = "ami-0e001c9271cf7f3b9"
 instance_type = "t2.micro"
 key_name = "Master-Client"
 tags = {
 Name = "hello-virginia"
 }
}
```

Terraform plan

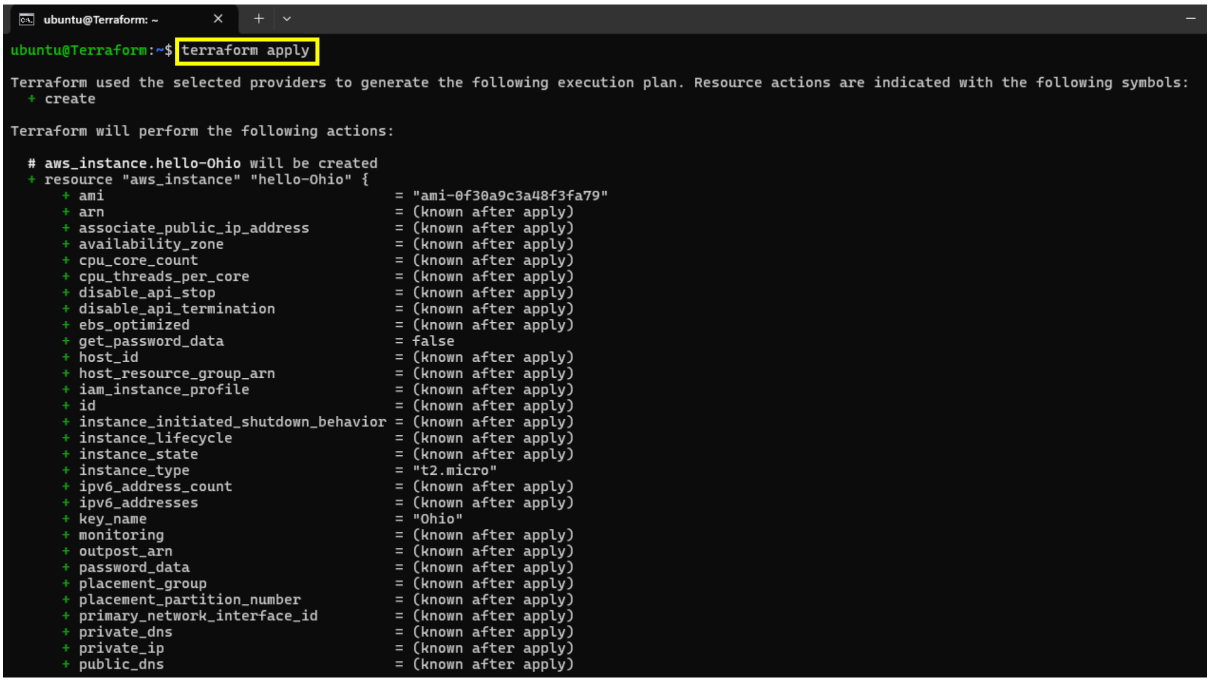


```
ubuntu@Terraform:~$ terraform plan

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.hello-Ohio will be created
  + resource "aws_instance" "hello-Ohio" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "hello-Ohio"
        }
      + tags_all                             = {
          + "Name" = "hello-Ohio"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }
```



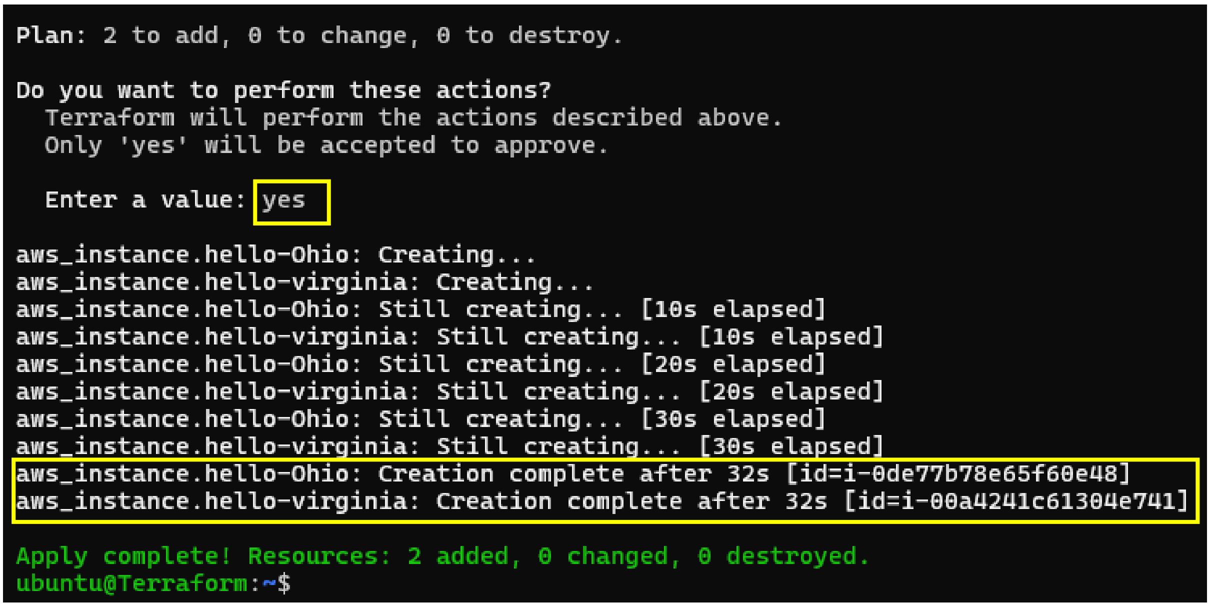
```
    }

  # aws_instance.hello-virginia will be created
  + resource "aws_instance" "hello-virginia" {
      + ami                                  = "ami-0e001c9271cf7f3b9"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Master-Client"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "hello-virginia"
        }
      + tags_all                             = {
          + "Name" = "hello-virginia"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

Plan: 2 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:~$
```

Terraform apply



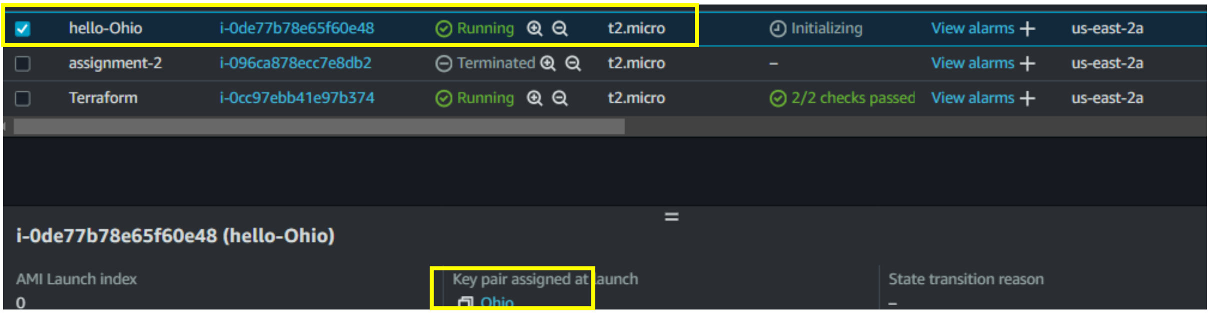
```
ubuntu@Terraform:~$ terraform apply

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.hello-Ohio will be created
  + resource "aws_instance" "hello-Ohio" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
```

Yes, it will start creating



```
Plan: 2 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.hello-Ohio: Creating...
aws_instance.hello-virginia: Creating...
aws_instance.hello-Ohio: Still creating... [10s elapsed]
aws_instance.hello-virginia: Still creating... [10s elapsed]
aws_instance.hello-Ohio: Still creating... [20s elapsed]
aws_instance.hello-virginia: Still creating... [20s elapsed]
aws_instance.hello-Ohio: Still creating... [30s elapsed]
aws_instance.hello-virginia: Still creating... [30s elapsed]
aws_instance.hello-Ohio: Creation complete after 32s [id=i-0de77b78e65f60e48]
aws_instance.hello-virginia: Creation complete after 32s [id=i-00a4241c61304e741]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@Terraform:~$
```

Now hello-Ohio Instances running on Ohio region



☑	hello-Ohio	i-0de77b78e65f60e48	Running	t2.micro	Initializing	View alarms +	us-east-2a
□	assignment-2	i-096ca878ecc7e8db2	Terminated	t2.micro	–	View alarms +	us-east-2a
□	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a

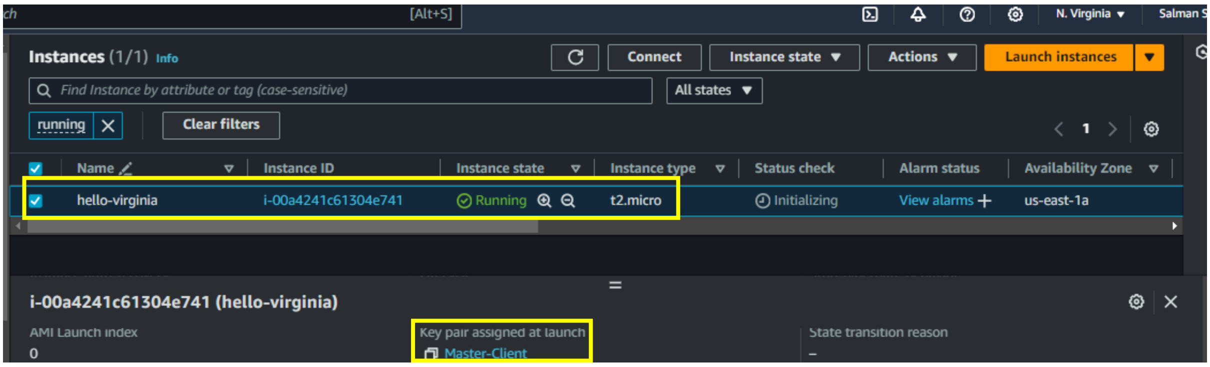
i-0de77b78e65f60e48 (hello-Ohio)

AMI Launch index
0

Key pair assigned at launch
Ohio

State transition reason
–

And Hello-virginia instances running on N.Virginia region



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	hello-virginia	i-00a4241c61304e741	✓ Running	t2.micro	Initializing	View alarms +	us-east-1a

i-00a4241c61304e741 (hello-virginia)

AMI Launch index: 0

Key pair assigned at launch: Master-Client

State transition reason: –

DevOps Certification Training

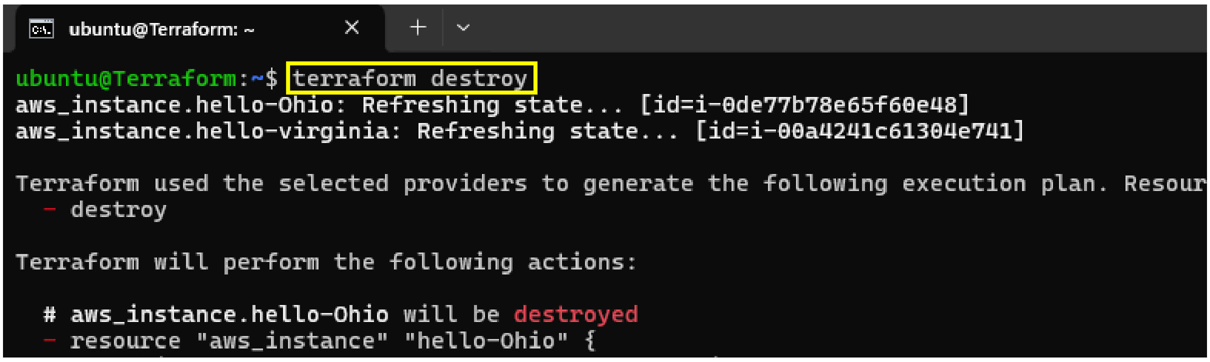
IntelliPaat

Terraform Assignment - 4

You have been asked to:

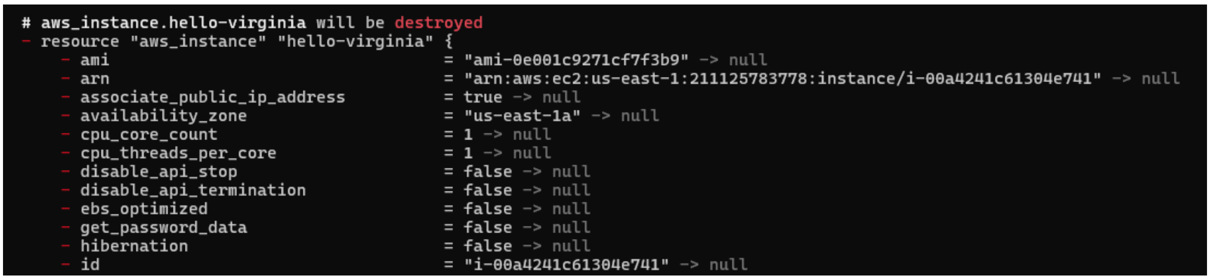
- Destroy the previous deployments
- Create a VPC with the required components using Terraform
- Deploy an EC2 instance inside the VPC

Terraform destroy



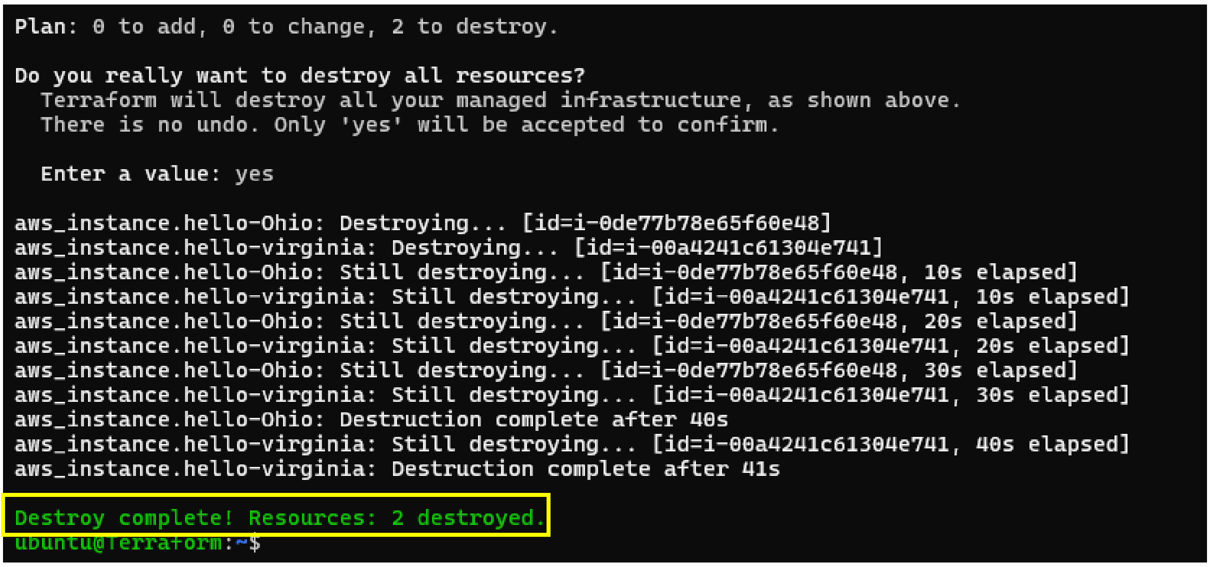
```
ubuntu@Terraform:~$ terraform destroy
aws_instance.hello-Ohio: Refreshing state... [id=i-0de77b78e65f60e48]
aws_instance.hello-virginia: Refreshing state... [id=i-00a4241c61304e741]

Terraform used the selected providers to generate the following execution plan. Resour
  - destroy

Terraform will perform the following actions:

  # aws_instance.hello-Ohio will be destroyed
  - resource "aws_instance" "hello-Ohio" {
```

destroyed



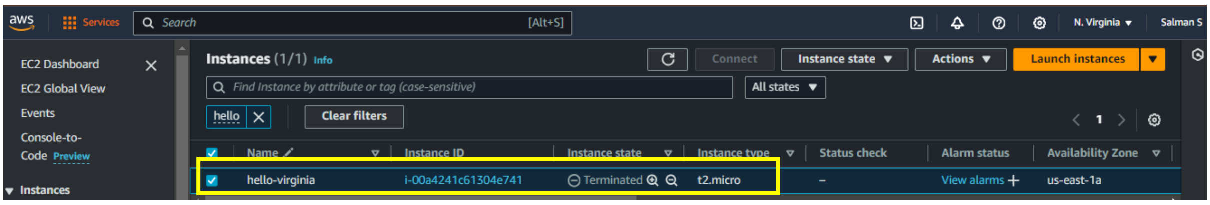
```
 # aws_instance.hello-virginia will be destroyed
 - resource "aws_instance" "hello-virginia" {
     - ami                                  = "ami-0e001c9271cf7f3b9" -> null
     - arn                                  = "arn:aws:ec2:us-east-1:211125783778:instance/i-00a4241c61304e741" -> null
     - associate_public_ip_address          = true -> null
     - availability_zone                    = "us-east-1a" -> null
     - cpu_core_count                       = 1 -> null
     - cpu_threads_per_core                 = 1 -> null
     - disable_api_stop                     = false -> null
     - disable_api_termination              = false -> null
     - ebs_optimized                        = false -> null
     - get_password_data                    = false -> null
     - hibernation                          = false -> null
     - id                                   = "i-00a4241c61304e741" -> null
```

Destroyed successfully

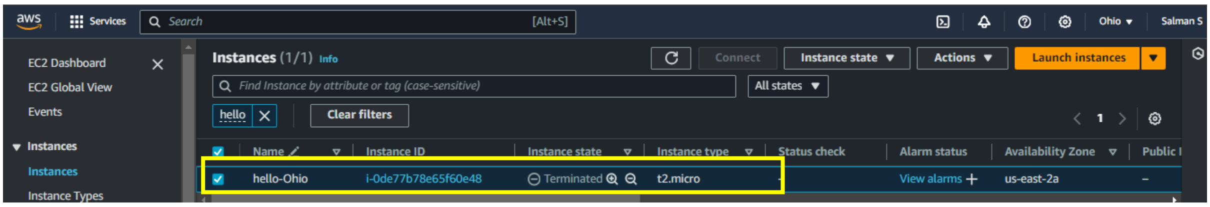


```
Plan: 0 to add, 0 to change, 2 to destroy.

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.hello-Ohio: Destroying... [id=i-0de77b78e65f60e48]
aws_instance.hello-virginia: Destroying... [id=i-00a4241c61304e741]
aws_instance.hello-Ohio: Still destroying... [id=i-0de77b78e65f60e48, 10s elapsed]
aws_instance.hello-virginia: Still destroying... [id=i-00a4241c61304e741, 10s elapsed]
aws_instance.hello-Ohio: Still destroying... [id=i-0de77b78e65f60e48, 20s elapsed]
aws_instance.hello-virginia: Still destroying... [id=i-00a4241c61304e741, 20s elapsed]
aws_instance.hello-Ohio: Still destroying... [id=i-0de77b78e65f60e48, 30s elapsed]
aws_instance.hello-virginia: Still destroying... [id=i-00a4241c61304e741, 30s elapsed]
aws_instance.hello-Ohio: Destruction complete after 40s
aws_instance.hello-virginia: Still destroying... [id=i-00a4241c61304e741, 40s elapsed]
aws_instance.hello-virginia: Destruction complete after 41s

Destroy complete! Resources: 2 destroyed.
ubuntu@Terraform:~$
```

Now, its successfully terminated

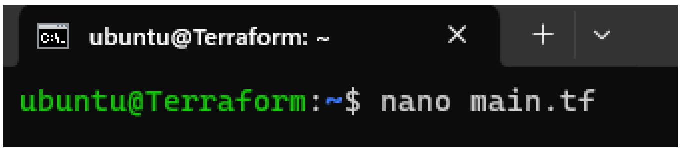


☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	hello-virginia	i-00a4241c61304e741	Terminated	t2.micro	–	View alarms +	us-east-1a

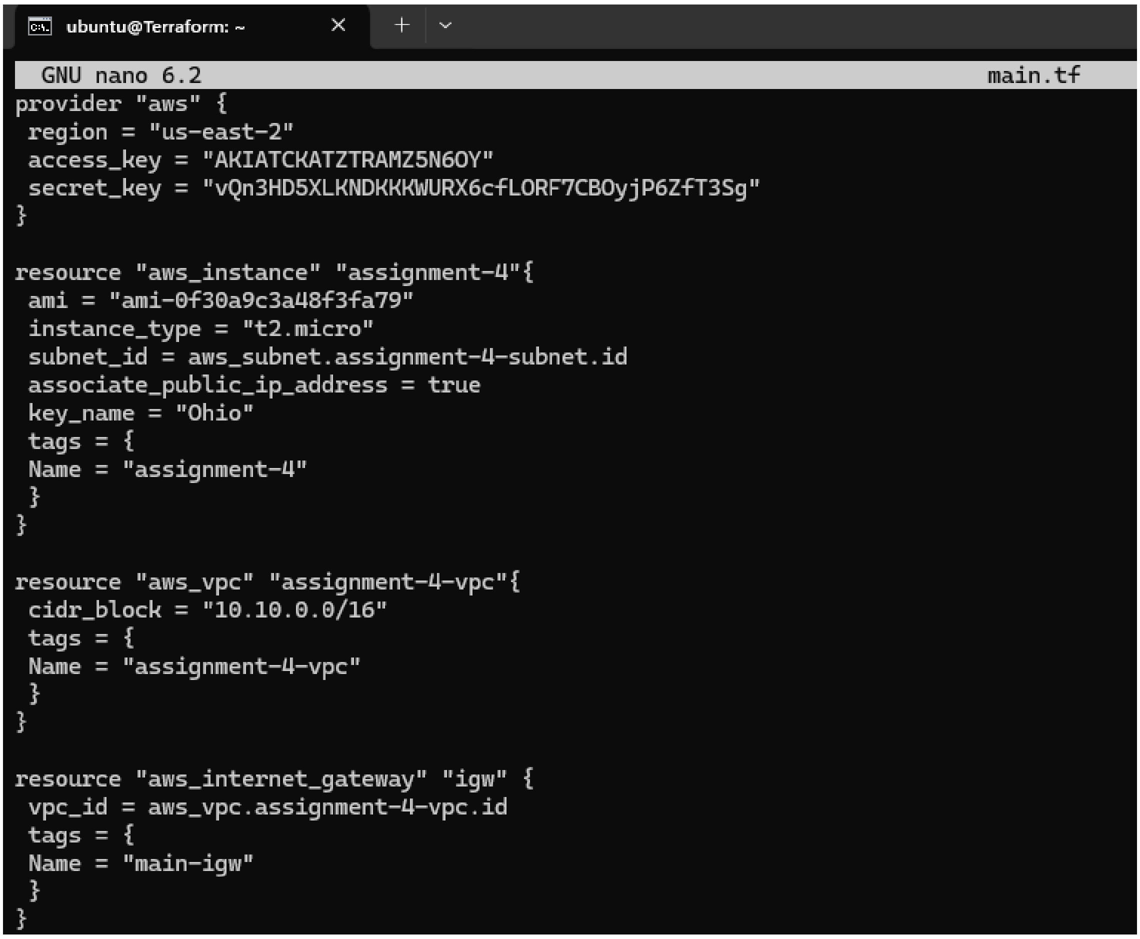


☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
☑	hello-Ohio	i-0de77b78e65f60e48	Terminated	t2.micro		View alarms +	us-east-2a	–

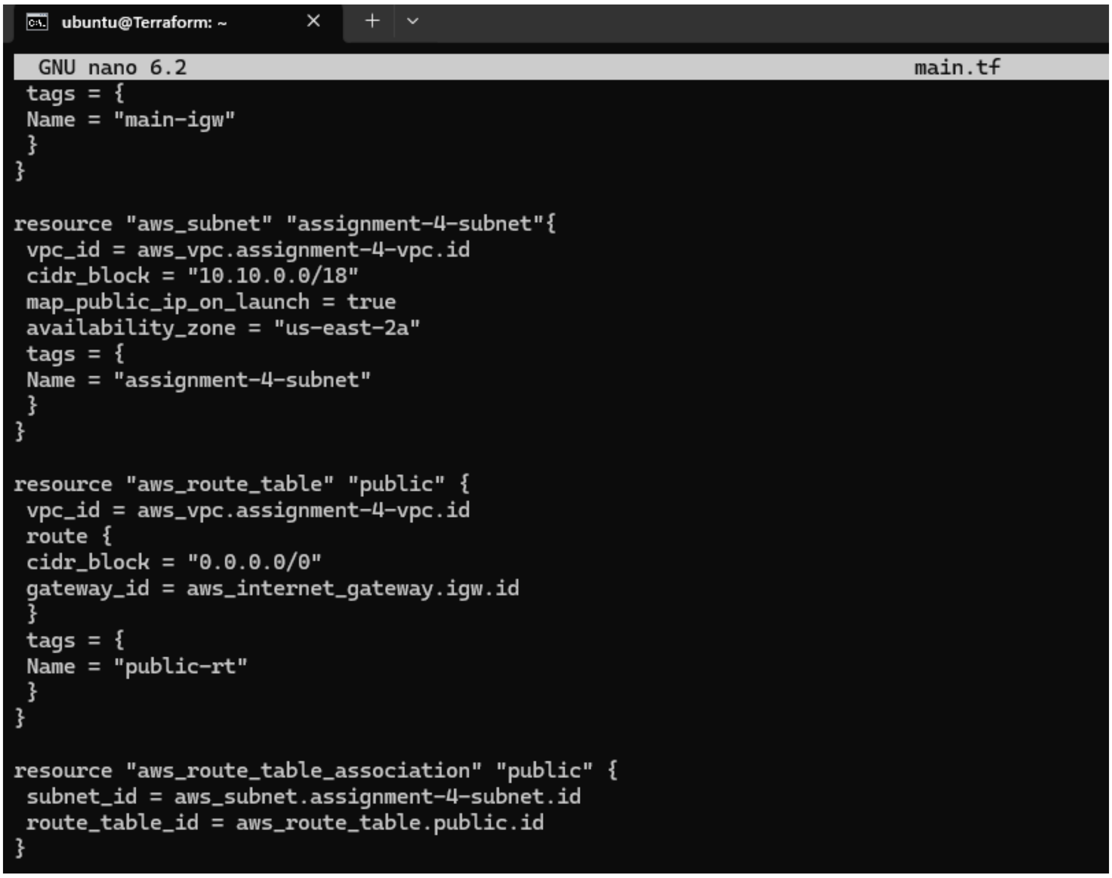
Now write terraform script as per 4th assignment



```
ubuntu@Terraform:~$ nano main.tf
```



```
  GNU nano 6.2                                                         main.tf
provider "aws" {
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

resource "aws_instance" "assignment-4"{
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 subnet_id = aws_subnet.assignment-4-subnet.id
 associate_public_ip_address = true
 key_name = "Ohio"
 tags = {
 Name = "assignment-4"
 }
}

resource "aws_vpc" "assignment-4-vpc"{
 cidr_block = "10.10.0.0/16"
 tags = {
 Name = "assignment-4-vpc"
 }
}

resource "aws_internet_gateway" "igw" {
 vpc_id = aws_vpc.assignment-4-vpc.id
 tags = {
 Name = "main-igw"
 }
}
```



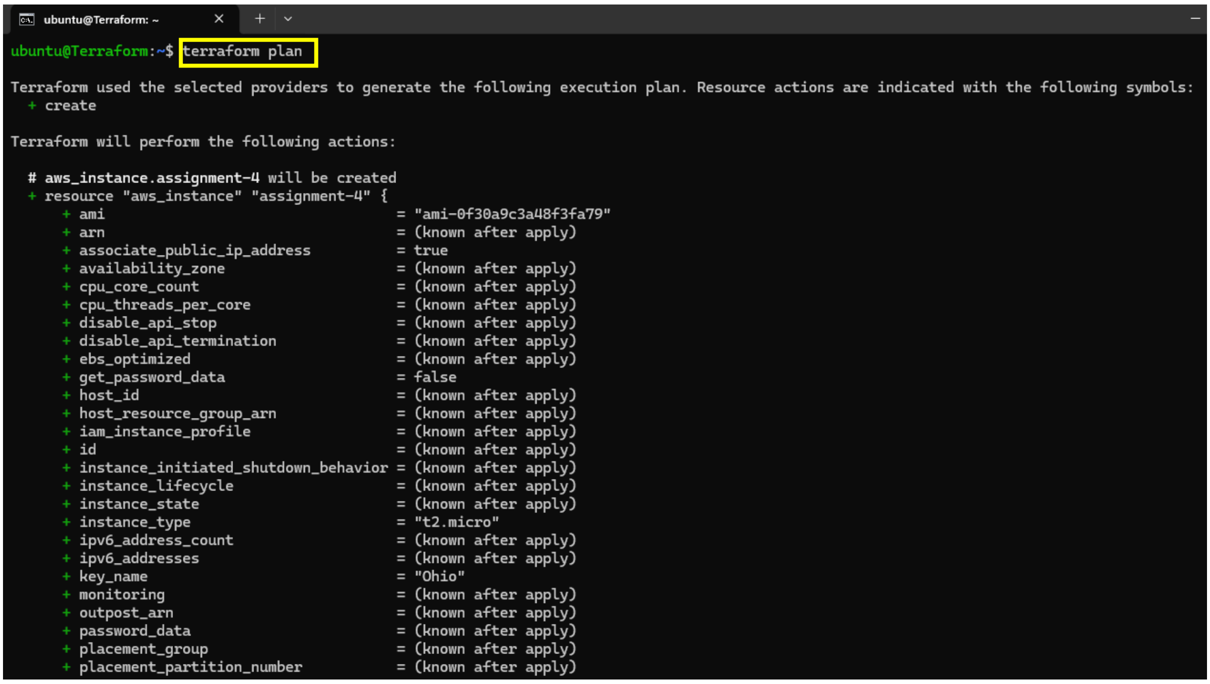
```
  GNU nano 6.2                                                         main.tf
 tags = {
 Name = "main-igw"
 }
}

resource "aws_subnet" "assignment-4-subnet"{
 vpc_id = aws_vpc.assignment-4-vpc.id
 cidr_block = "10.10.0.0/18"
 map_public_ip_on_launch = true
 availability_zone = "us-east-2a"
 tags = {
 Name = "assignment-4-subnet"
 }
}

resource "aws_route_table" "public" {
 vpc_id = aws_vpc.assignment-4-vpc.id
 route {
 cidr_block = "0.0.0.0/0"
 gateway_id = aws_internet_gateway.igw.id
 }
 tags = {
 Name = "public-rt"
 }
}

resource "aws_route_table_association" "public" {
 subnet_id = aws_subnet.assignment-4-subnet.id
 route_table_id = aws_route_table.public.id
}
```

Script-4

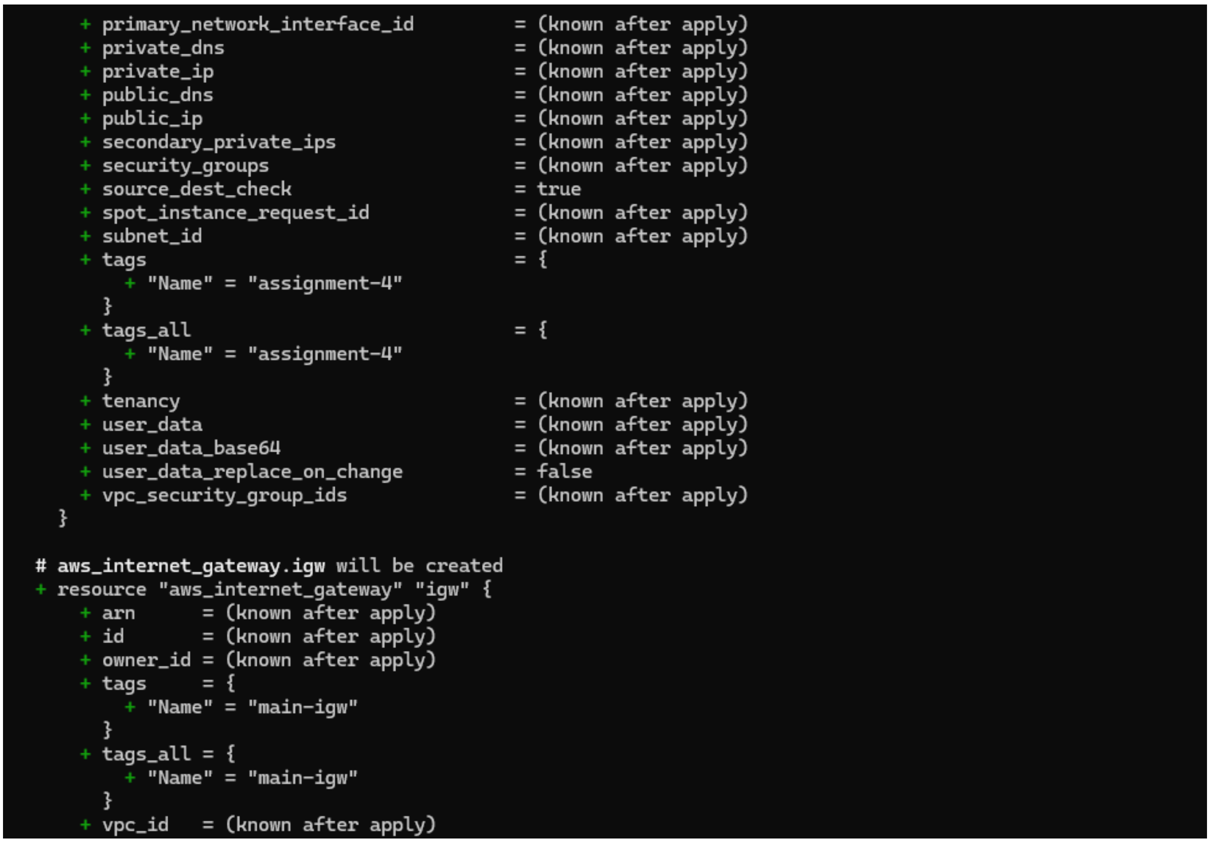
```
provider "aws" {
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

resource "aws_instance" "assignment-4"{
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 subnet_id = aws_subnet.assignment-4-subnet.id
 associate_public_ip_address = true
 key_name = "Ohio"
 tags = {
 Name = "assignment-4"
 }
}

resource "aws_vpc" "assignment-4-vpc"{
 cidr_block = "10.10.0.0/16"
 tags = {
 Name = "assignment-4-vpc"
 }
}

resource "aws_internet_gateway" "igw" {
 vpc_id = aws_vpc.assignment-4-vpc.id
 tags = {
 Name = "main-igw"
 }
}
```

```
resource "aws_subnet" "assignment-4-subnet"{
 vpc_id = aws_vpc.assignment-4-vpc.id
 cidr_block = "10.10.0.0/18"
 map_public_ip_on_launch = true
 availability_zone = "us-east-2a"
 tags = {
 Name = "assignment-4-subnet"
 }
}

resource "aws_route_table" "public" {
 vpc_id = aws_vpc.assignment-4-vpc.id
 route {
 cidr_block = "0.0.0.0/0"
 gateway_id = aws_internet_gateway.igw.id
 }
 tags = {
 Name = "public-rt"
 }
}

resource "aws_route_table_association" "public" {
 subnet_id = aws_subnet.assignment-4-subnet.id
 route_table_id = aws_route_table.public.id
}
```

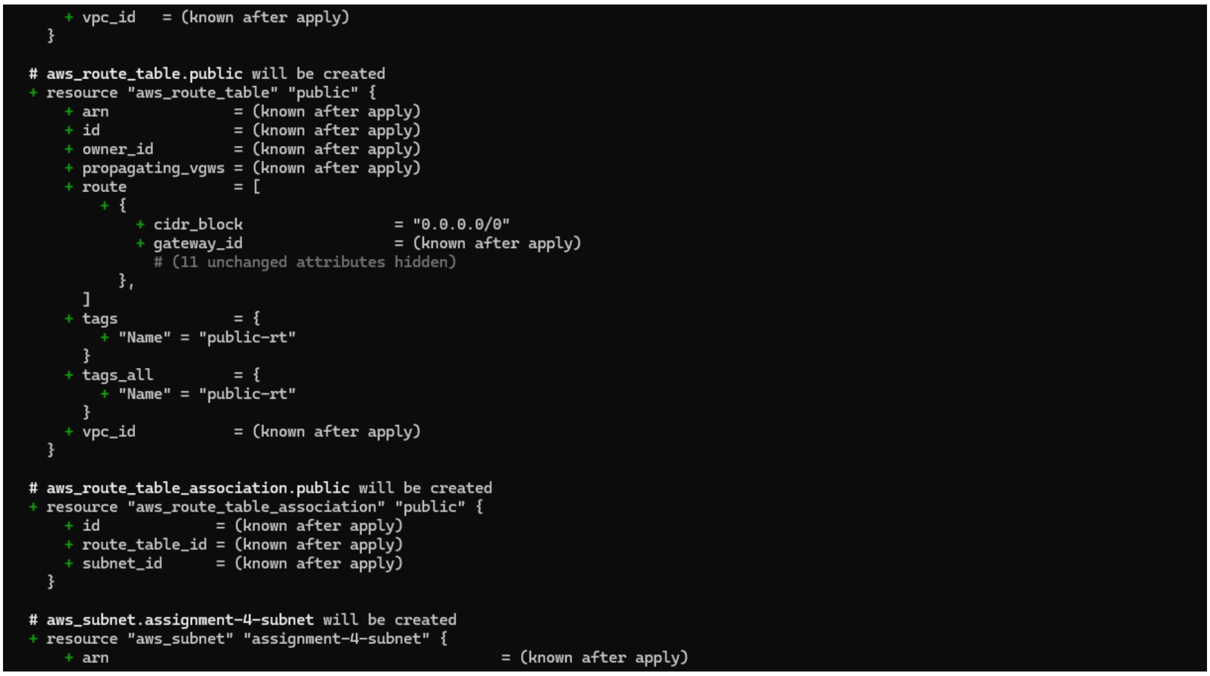
Terraform plan



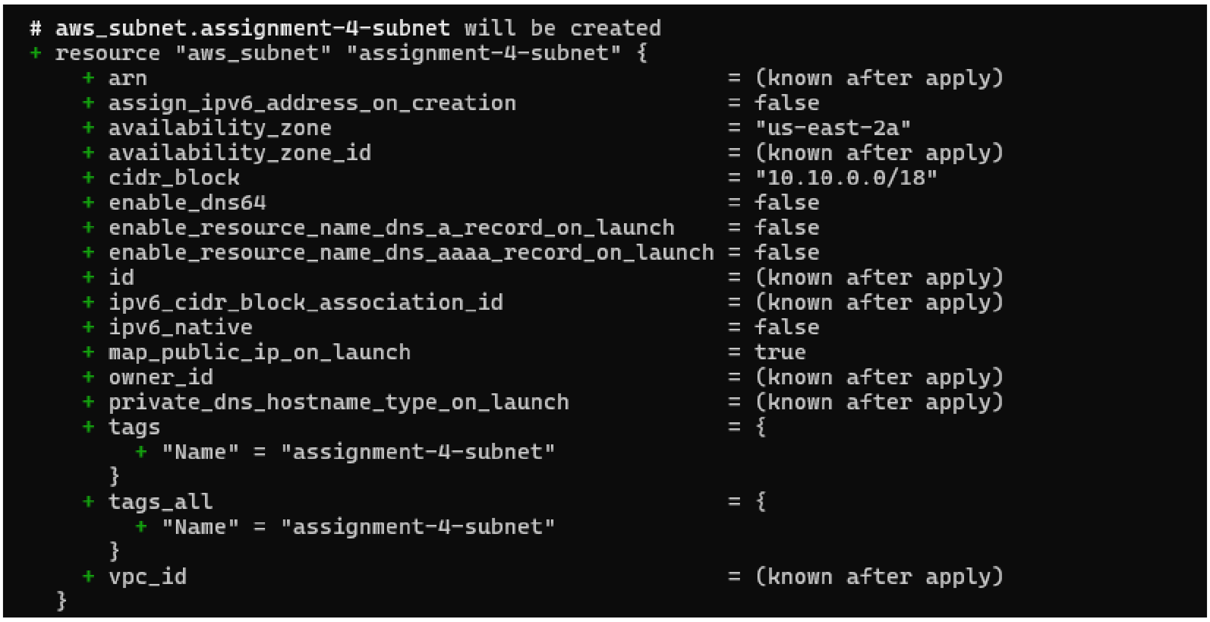
```
ubuntu@Terraform:~$ terraform plan

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.assignment-4 will be created
  + resource "aws_instance" "assignment-4" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
```



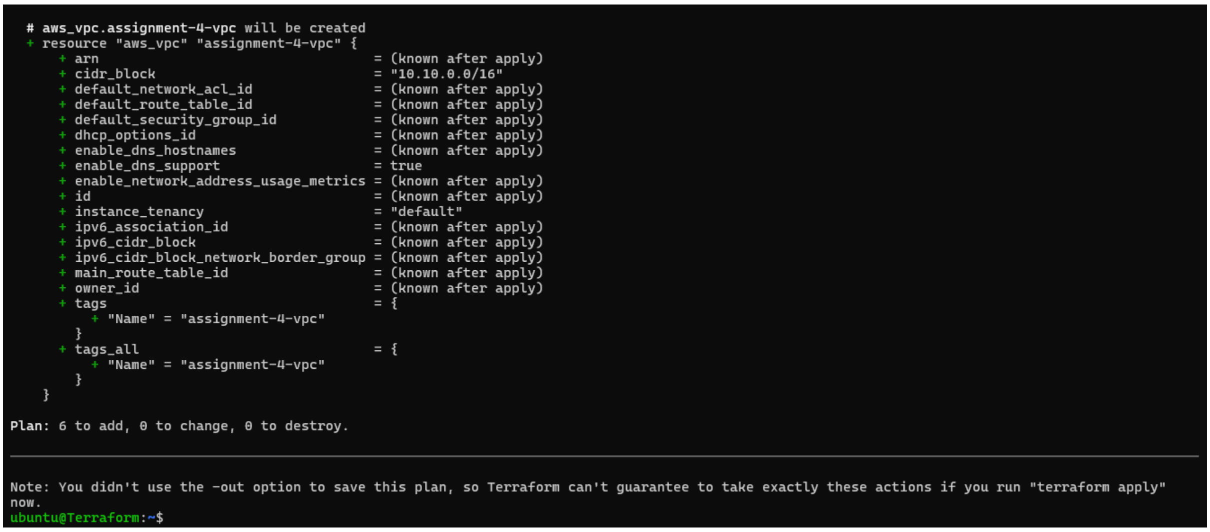
```
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "assignment-4"
        }
      + tags_all                             = {
          + "Name" = "assignment-4"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

  # aws_internet_gateway.igw will be created
  + resource "aws_internet_gateway" "igw" {
      + arn      = (known after apply)
      + id       = (known after apply)
      + owner_id = (known after apply)
      + tags     = {
          + "Name" = "main-igw"
        }
      + tags_all = {
          + "Name" = "main-igw"
        }
      + vpc_id   = (known after apply)
```



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

  # aws_route_table.public will be created
  + resource "aws_route_table" "public" {
      + arn              = (known after apply)
      + id               = (known after apply)
      + owner_id         = (known after apply)
      + propagating_vgws = (known after apply)
      + route            = [
          + {
              + cidr_block                 = "0.0.0.0/0"
              + gateway_id                 = (known after apply)
                # (11 unchanged attributes hidden)
            },
        ]
      + tags             = {
          + "Name" = "public-rt"
        }
      + tags_all         = {
          + "Name" = "public-rt"
        }
      + vpc_id           = (known after apply)
    }

  # aws_route_table_association.public will be created
  + resource "aws_route_table_association" "public" {
      + id             = (known after apply)
      + route_table_id = (known after apply)
      + subnet_id      = (known after apply)
    }

  # aws_subnet.assignment-4-subnet will be created
  + resource "aws_subnet" "assignment-4-subnet" {
      + arn                                            = (known after apply)
```

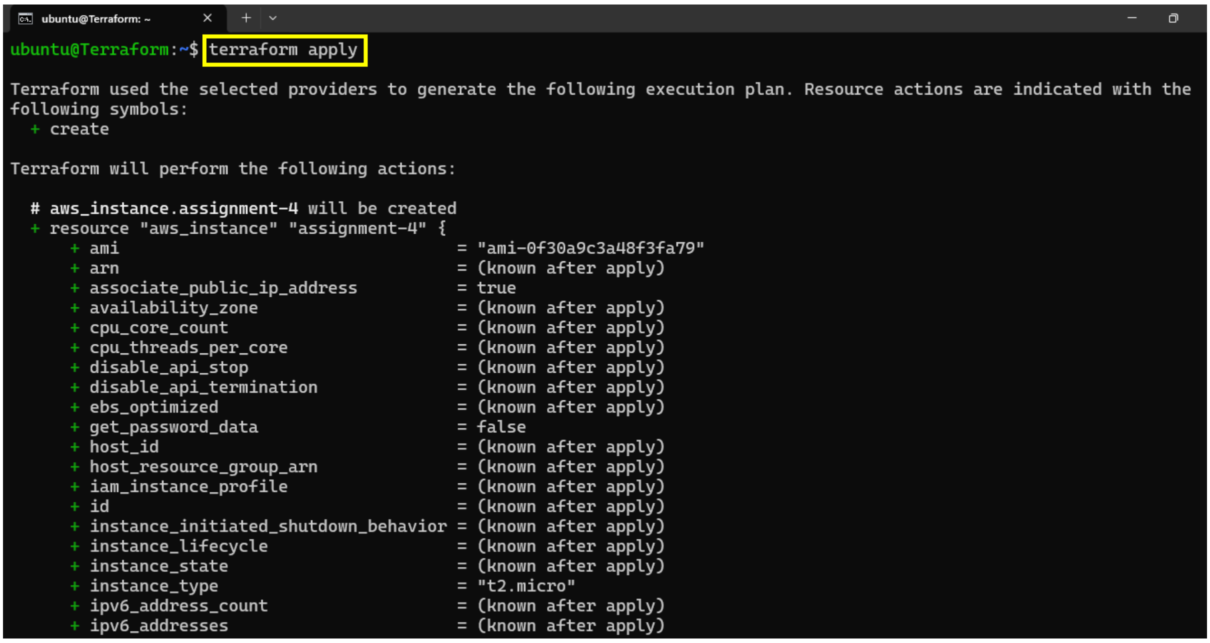


```
  # aws_subnet.assignment-4-subnet will be created
  + resource "aws_subnet" "assignment-4-subnet" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "us-east-2a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.10.0.0/18"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "assignment-4-subnet"
        }
      + tags_all                                       = {
          + "Name" = "assignment-4-subnet"
        }
      + vpc_id                                         = (known after apply)
    }
```

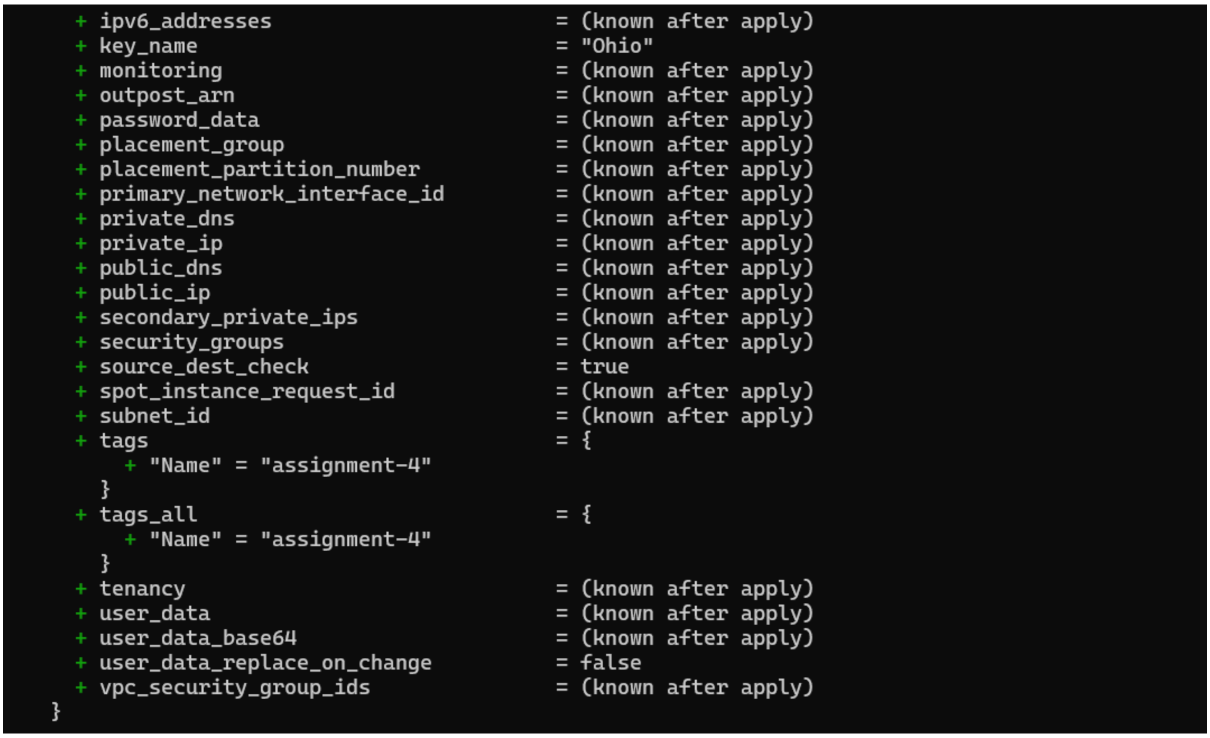


```
  # aws_vpc.assignment-4-vpc will be created
  + resource "aws_vpc" "assignment-4-vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.10.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "assignment-4-vpc"
        }
      + tags_all                             = {
          + "Name" = "assignment-4-vpc"
        }
    }

Plan: 6 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:~$
```

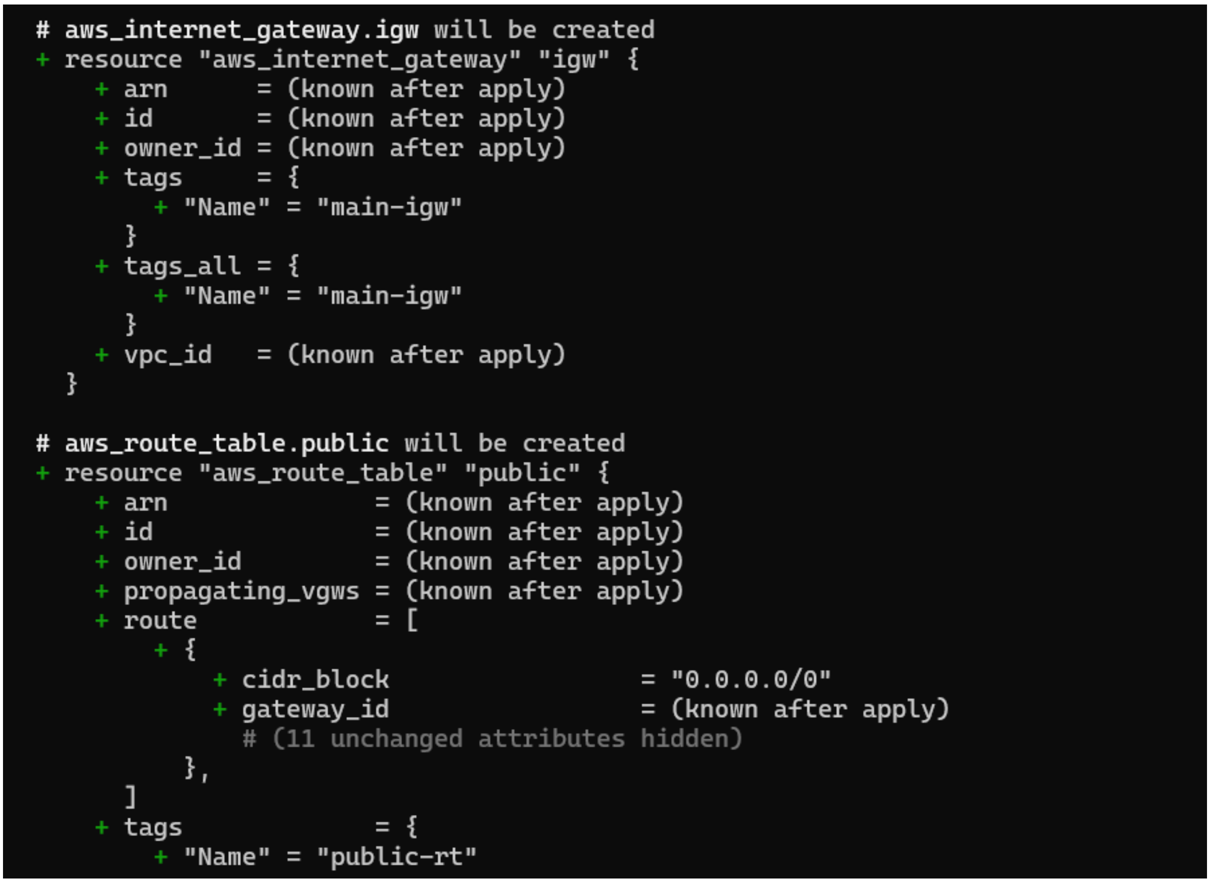
Terraform apply



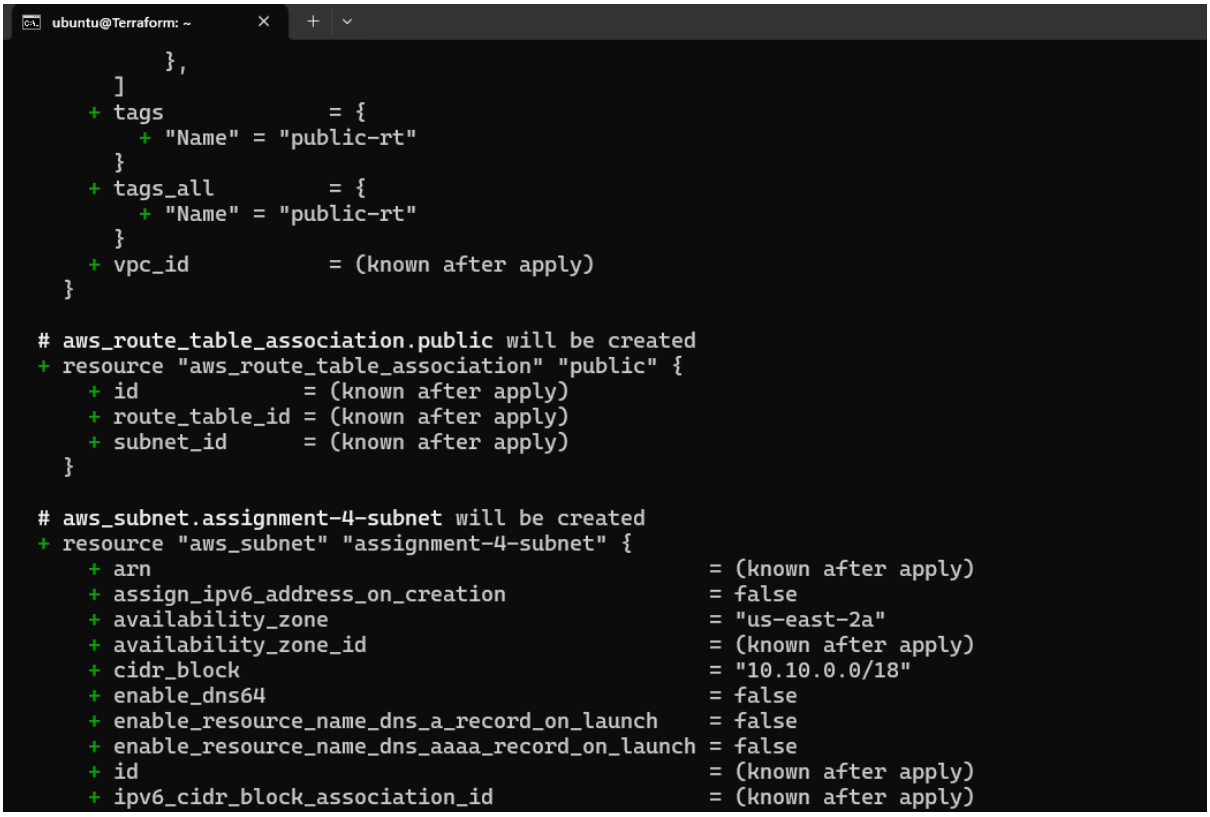
```
ubuntu@Terraform:~$ terraform apply

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.assignment-4 will be created
  + resource "aws_instance" "assignment-4" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
```



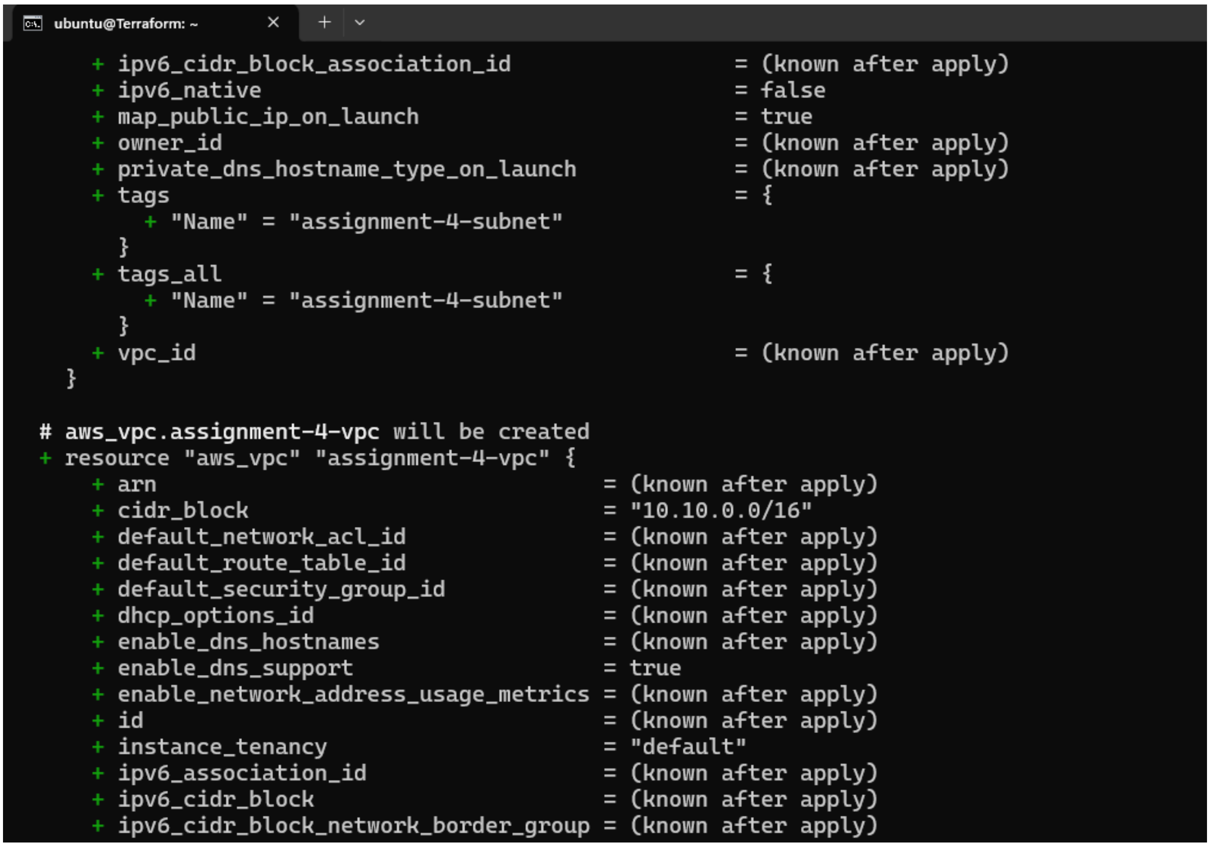
```
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "assignment-4"
        }
      + tags_all                             = {
          + "Name" = "assignment-4"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }
```



```
# aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
        + "Name" = "main-igw"
      }
    + tags_all = {
        + "Name" = "main-igw"
      }
    + vpc_id   = (known after apply)
  }

# aws_route_table.public will be created
+ resource "aws_route_table" "public" {
    + arn              = (known after apply)
    + id               = (known after apply)
    + owner_id         = (known after apply)
    + propagating_vgws = (known after apply)
    + route            = [
        + {
            + cidr_block                 = "0.0.0.0/0"
            + gateway_id                 = (known after apply)
              # (11 unchanged attributes hidden)
          },
      ]
    + tags             = {
        + "Name" = "public-rt"
```

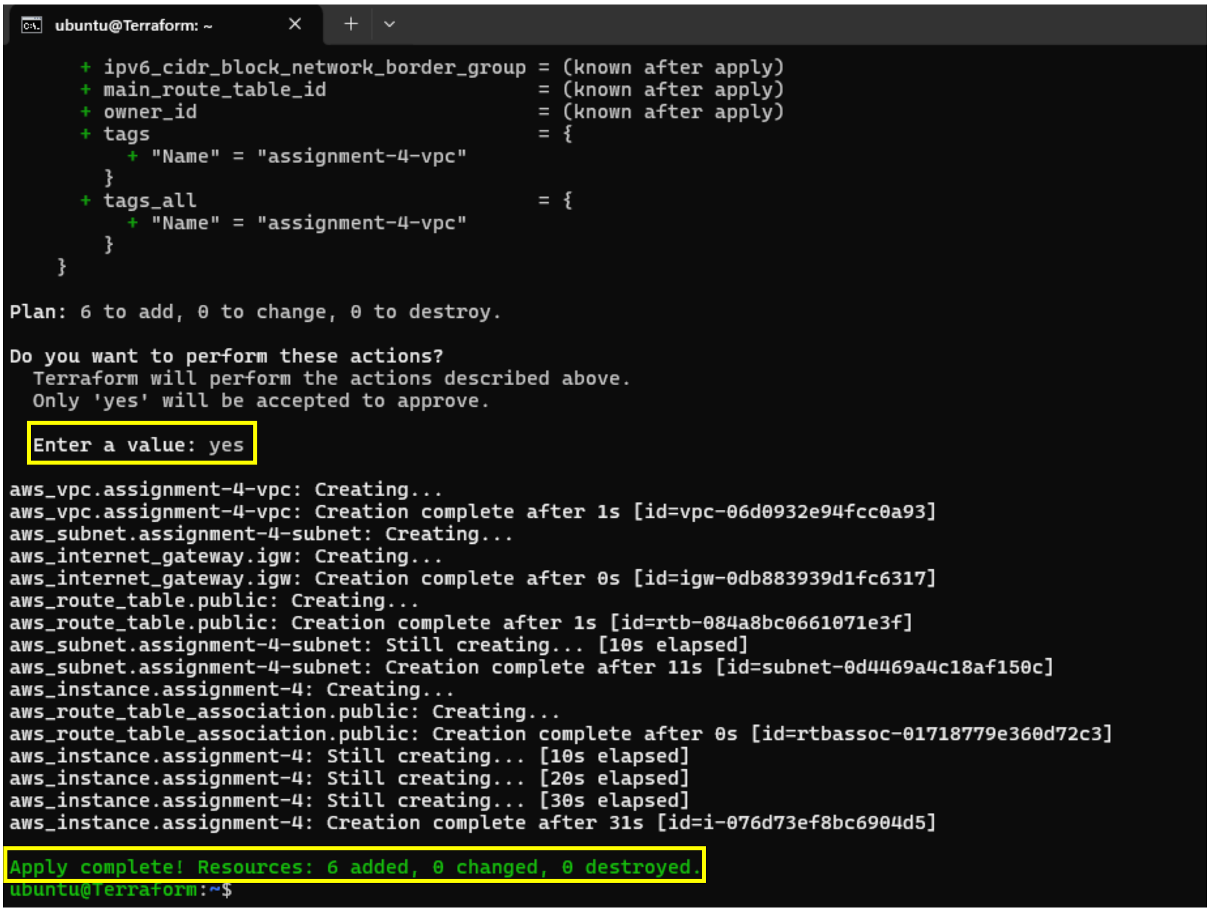


```
          },
      ]
    + tags             = {
        + "Name" = "public-rt"
      }
    + tags_all         = {
        + "Name" = "public-rt"
      }
    + vpc_id           = (known after apply)
  }

# aws_route_table_association.public will be created
+ resource "aws_route_table_association" "public" {
    + id             = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id      = (known after apply)
  }

# aws_subnet.assignment-4-subnet will be created
+ resource "aws_subnet" "assignment-4-subnet" {
    + arn                                            = (known after apply)
    + assign_ipv6_address_on_creation                = false
    + availability_zone                              = "us-east-2a"
    + availability_zone_id                           = (known after apply)
    + cidr_block                                     = "10.10.0.0/18"
    + enable_dns64                                   = false
    + enable_resource_name_dns_a_record_on_launch    = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                                             = (known after apply)
    + ipv6_cidr_block_association_id                 = (known after apply)
```



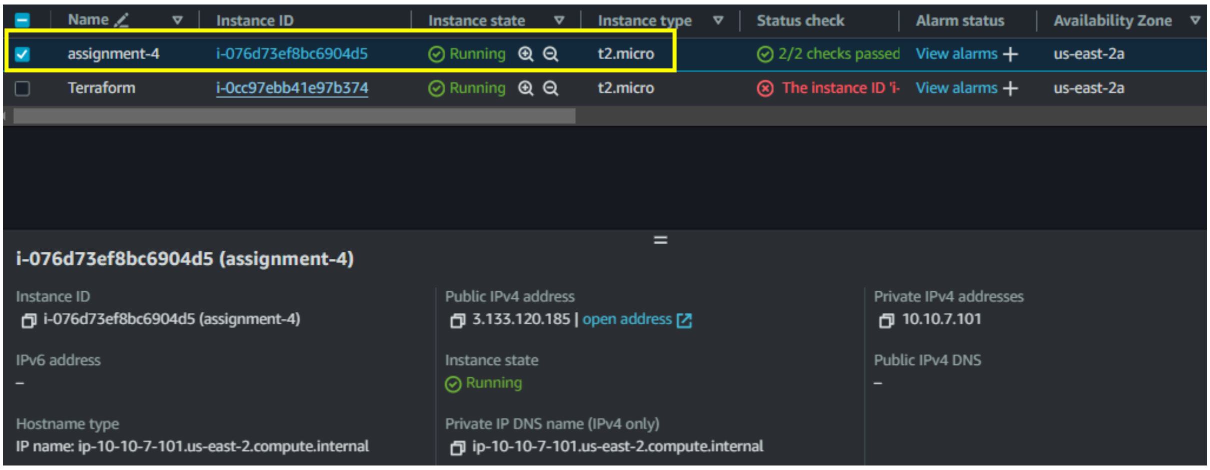
```
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = true
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "assignment-4-subnet"
        }
      + tags_all                                       = {
          + "Name" = "assignment-4-subnet"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.assignment-4-vpc will be created
  + resource "aws_vpc" "assignment-4-vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.10.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
```

Yes, it will create the resources/infrastructure



```
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "assignment-4-vpc"
        }
      + tags_all                             = {
          + "Name" = "assignment-4-vpc"
        }
    }

Plan: 6 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.assignment-4-vpc: Creating...
aws_vpc.assignment-4-vpc: Creation complete after 1s [id=vpc-06d0932e94fcc0a93]
aws_subnet.assignment-4-subnet: Creating...
aws_internet_gateway.igw: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-0db883939d1fc6317]
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 1s [id=rtb-084a8bc0661071e3f]
aws_subnet.assignment-4-subnet: Still creating... [10s elapsed]
aws_subnet.assignment-4-subnet: Creation complete after 11s [id=subnet-0d4469a4c18af150c]
aws_instance.assignment-4: Creating...
aws_route_table_association.public: Creating...
aws_route_table_association.public: Creation complete after 0s [id=rtbassoc-01718779e360d72c3]
aws_instance.assignment-4: Still creating... [10s elapsed]
aws_instance.assignment-4: Still creating... [20s elapsed]
aws_instance.assignment-4: Still creating... [30s elapsed]
aws_instance.assignment-4: Creation complete after 31s [id=i-076d73ef8bc6904d5]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
ubuntu@Terraform:~$
```

Now Running assignment-4 instances, Public ip is assigned to the instance



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☑	assignment-4	i-076d73ef8bc6904d5	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a
☐	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	The instance ID 'i-	View alarms +	us-east-2a

i-076d73ef8bc6904d5 (assignment-4)

Instance ID
i-076d73ef8bc6904d5 (assignment-4)

Public IPv4 address
3.133.120.185 | open address

Private IPv4 addresses
10.10.7.101

IPv6 address
–

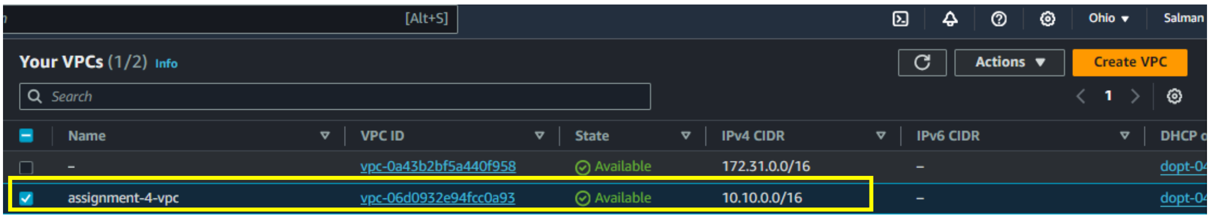
Instance state
Running

Public IPv4 DNS
–

Hostname type
IP name: ip-10-10-7-101.us-east-2.compute.internal

Private IP DNS name (IPv4 only)
ip-10-10-7-101.us-east-2.compute.internal

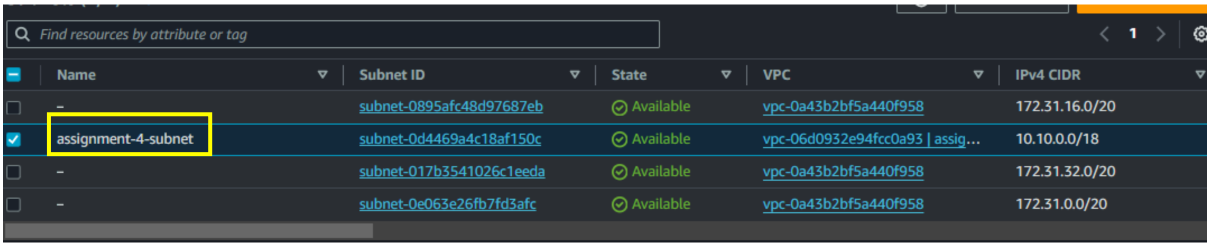
VPC created



Your VPCs (1/2) Info

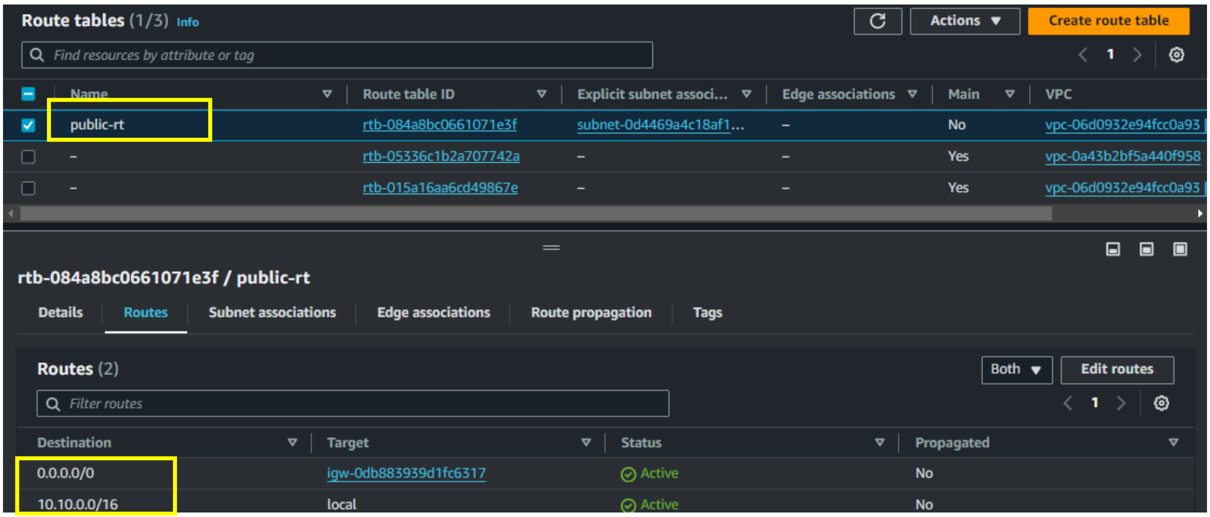
☐	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o
☐	–	vpc-0a43b2bf5a440f958	Available	172.31.0.0/16	–	dopt-0
☑	assignment-4-vpc	vpc-06d0932e94fcc0a93	Available	10.10.0.0/16	–	dopt-0

Subnet created



☐	Name	Subnet ID	State	VPC	IPv4 CIDR
☐	–	subnet-0895afc48d97687eb	Available	vpc-0a43b2bf5a440f958	172.31.16.0/20
☑	assignment-4-subnet	subnet-0d4469a4c18af150c	Available	vpc-06d0932e94fcc0a93 | assig...	10.10.0.0/18
☐	–	subnet-017b3541026c1eeda	Available	vpc-0a43b2bf5a440f958	172.31.32.0/20
☐	–	subnet-0e063e26fb7fd3afc	Available	vpc-0a43b2bf5a440f958	172.31.0.0/20

Route table created and routes



Route tables (1/3) Info

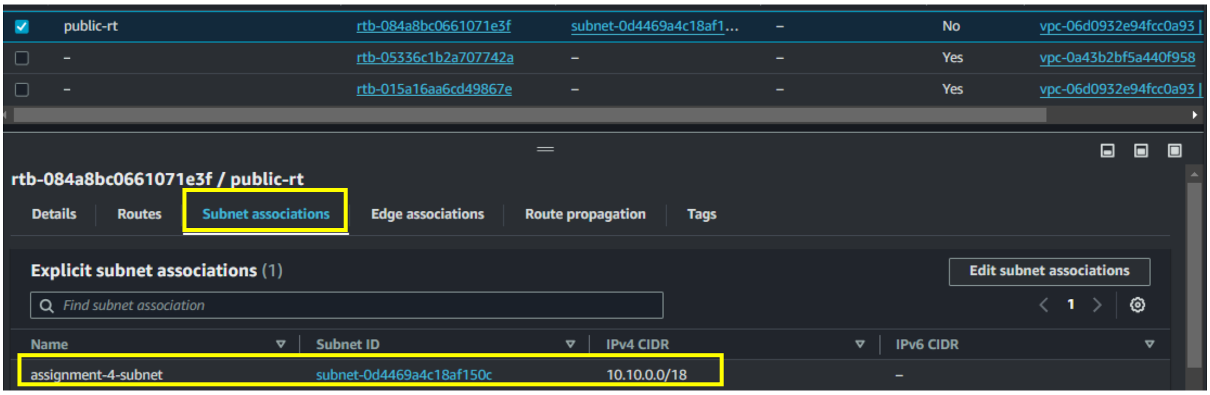
☐	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
☑	public-rt	rtb-084a8bc0661071e3f	subnet-0d4469a4c18af1...	–	No	vpc-06d0932e94fcc0a93
☐	–	rtb-05336c1b2a707742a	–	–	Yes	vpc-0a43b2bf5a440f958
☐	–	rtb-015a16aa6cd49867e	–	–	Yes	vpc-06d0932e94fcc0a93

rtb-084a8bc0661071e3f / public-rt

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0db883939d1fc6317	Active	No
10.10.0.0/16	local	Active	No

subnet association



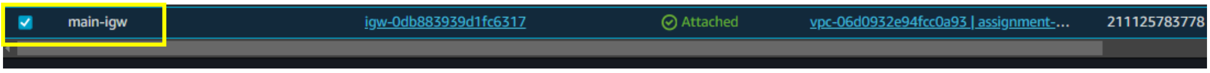
☑	public-rt	rtb-084a8bc0661071e3f	subnet-0d4469a4c18af1...	–	No	vpc-06d0932e94fcc0a93 |
□	–	rtb-05336c1b2a707742a	–	–	Yes	vpc-0a43b2bf5a440f958
□	–	rtb-015a16aa6cd49867e	–	–	Yes	vpc-06d0932e94fcc0a93 |

rtb-084a8bc0661071e3f / public-rt

Explicit subnet associations (1)

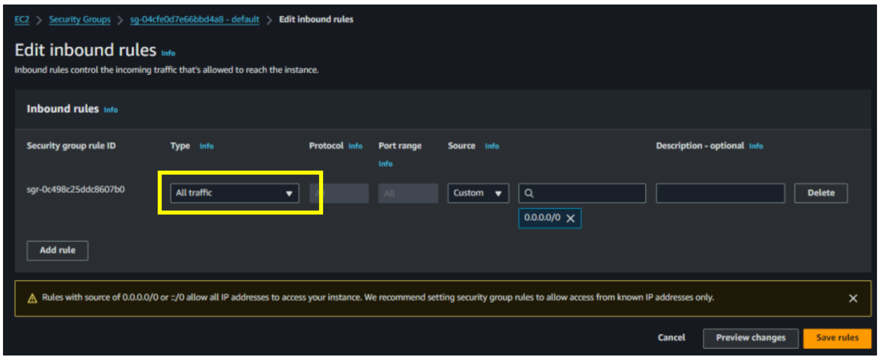
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
assignment-4-subnet	subnet-0d4469a4c18af150c	10.10.0.0/18	–

Internet gateway created & attached to our VPC



☑	main-igw	igw-0db883939d1fc6317	✓ Attached	vpc-06d0932e94fcc0a93 | assignment-...	211125783778

Allowing all traffic to assignment 4 instances Security Group



EC2 > Security Groups > sg-04cfe0d7e66bbd4a8 - default > Edit inbound rules

Edit inbound rules

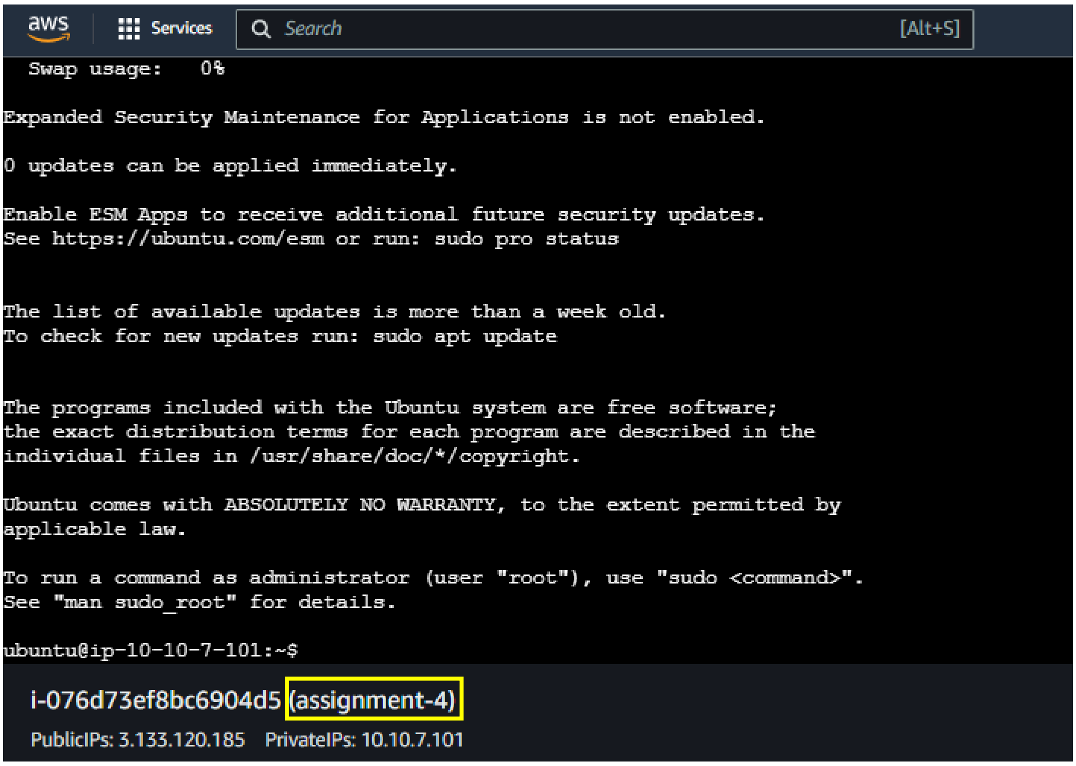
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-0c498c25ddc8607b0	All traffic	A	All	Custom	0.0.0.0/0		Delete

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

successfully connected to the ec2 instance



```
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-10-7-101:~$
```

i-076d73ef8bc6904d5 (assignment-4)

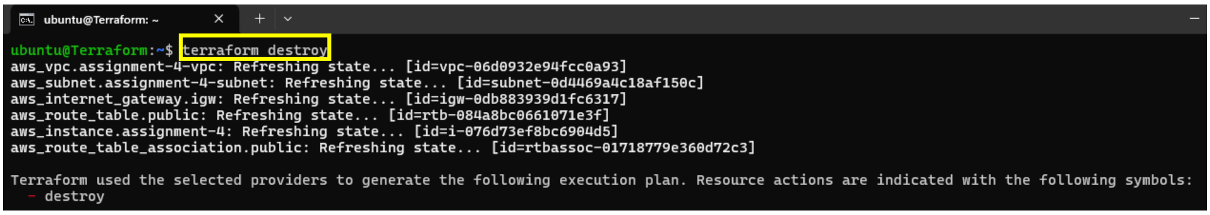
PublicIPs: 3.133.120.185 PrivateIPs: 10.10.7.101

Terraform Assignment - 5

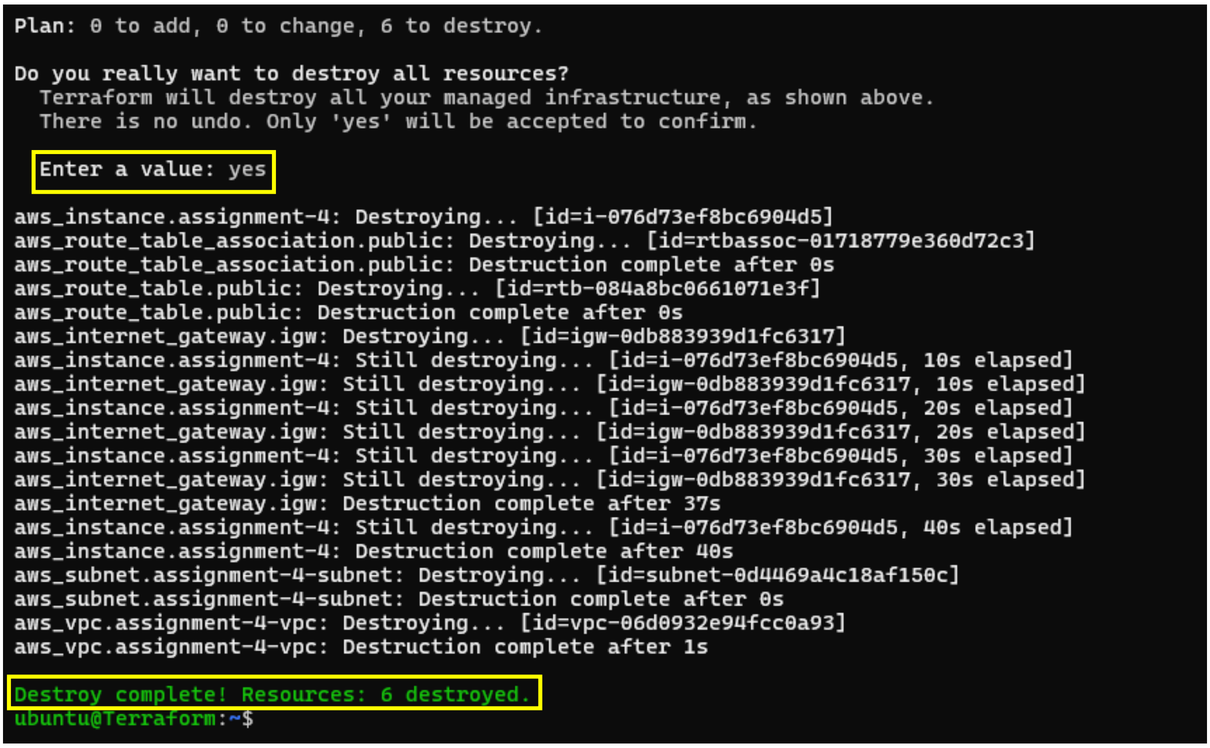
You have been asked to:

- Destroy the previous deployments
- Create a script to install apache2
- Run this script on a newly created EC2 instance
- Print the IP address of the instance in a file on the local, once deployed

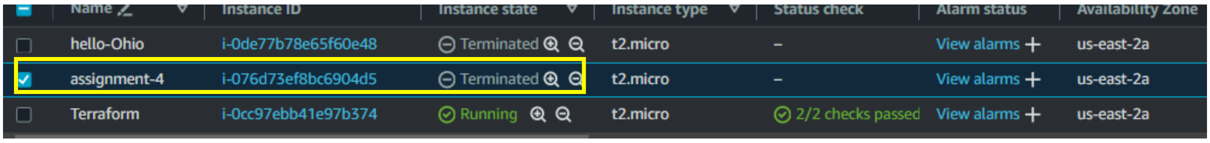
terraform destroy



```
ubuntu@Terraform:~$ terraform destroy
aws_vpc.assignment-4-vpc: Refreshing state... [id=vpc-06d0932e94fcc0a93]
aws_subnet.assignment-4-subnet: Refreshing state... [id=subnet-0d4469a4c18af150c]
aws_internet_gateway.igw: Refreshing state... [id=igw-0db883939d1fc6317]
aws_route_table.public: Refreshing state... [id=rtb-084a8bc0661071e3f]
aws_instance.assignment-4: Refreshing state... [id=i-076d73ef8bc6904d5]
aws_route_table_association.public: Refreshing state... [id=rtbassoc-01718779e360d72c3]

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

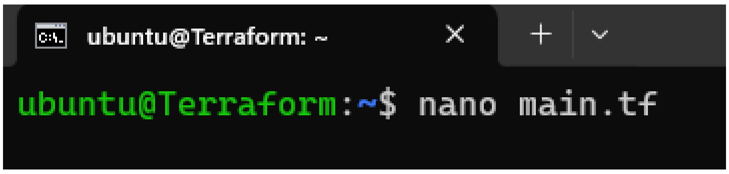


```
Plan: 0 to add, 0 to change, 6 to destroy.

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.assignment-4: Destroying... [id=i-076d73ef8bc6904d5]
aws_route_table_association.public: Destroying... [id=rtbassoc-01718779e360d72c3]
aws_route_table_association.public: Destruction complete after 0s
aws_route_table.public: Destroying... [id=rtb-084a8bc0661071e3f]
aws_route_table.public: Destruction complete after 0s
aws_internet_gateway.igw: Destroying... [id=igw-0db883939d1fc6317]
aws_instance.assignment-4: Still destroying... [id=i-076d73ef8bc6904d5, 10s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-0db883939d1fc6317, 10s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-076d73ef8bc6904d5, 20s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-0db883939d1fc6317, 20s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-076d73ef8bc6904d5, 30s elapsed]
aws_internet_gateway.igw: Still destroying... [id=igw-0db883939d1fc6317, 30s elapsed]
aws_internet_gateway.igw: Destruction complete after 37s
aws_instance.assignment-4: Still destroying... [id=i-076d73ef8bc6904d5, 40s elapsed]
aws_instance.assignment-4: Destruction complete after 40s
aws_subnet.assignment-4-subnet: Destroying... [id=subnet-0d4469a4c18af150c]
aws_subnet.assignment-4-subnet: Destruction complete after 0s
aws_vpc.assignment-4-vpc: Destroying... [id=vpc-06d0932e94fcc0a93]
aws_vpc.assignment-4-vpc: Destruction complete after 1s

Destroy complete! Resources: 6 destroyed.
ubuntu@Terraform:~$
```

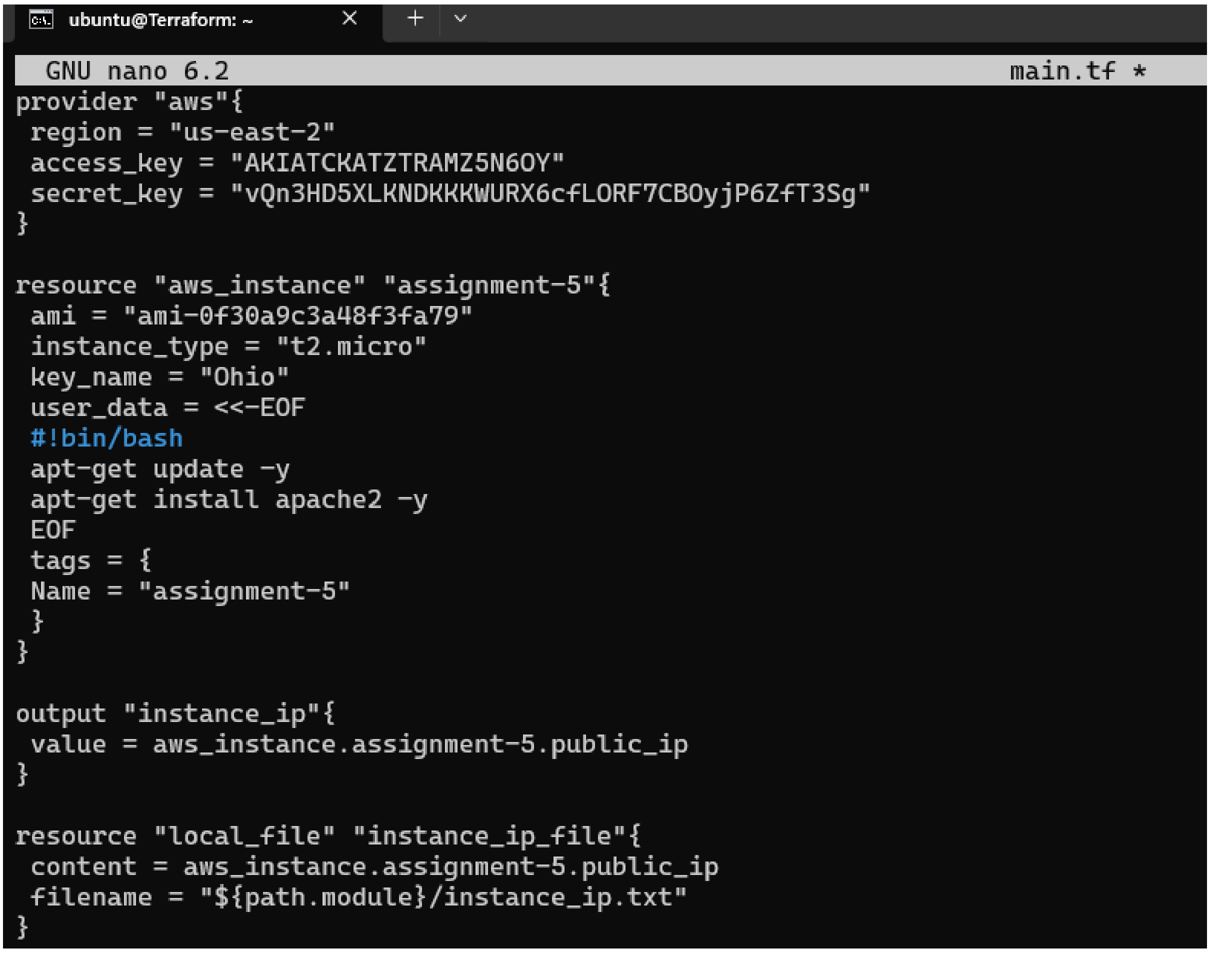


	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	hello-Ohio	i-0de77b78e65f60e48	Terminated	t2.micro	–	View alarms +	us-east-2a
☑	assignment-4	i-076d73ef8bc6904d5	Terminated	t2.micro	–	View alarms +	us-east-2a
□	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a

Now write terraform script as per 5th assignment



```
ubuntu@Terraform:~$ nano main.tf
```

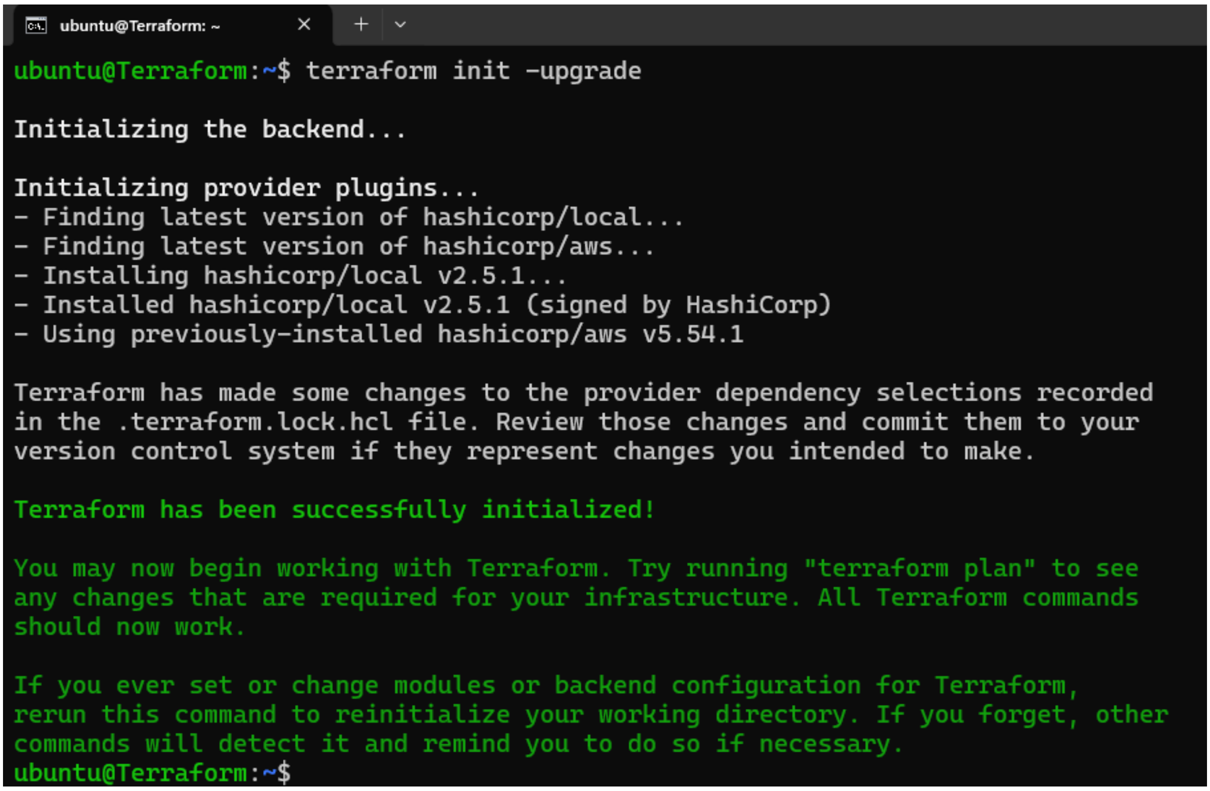


```
 GNU nano 6.2                                                       main.tf *
provider "aws"{
 region = "us-east-2"
 access_key = "AKIATCKATZTRAMZ5N6OY"
 secret_key = "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"
}

resource "aws_instance" "assignment-5"{
 ami = "ami-0f30a9c3a48f3fa79"
 instance_type = "t2.micro"
 key_name = "Ohio"
 user_data = <<-EOF
 #!bin/bash
 apt-get update -y
 apt-get install apache2 -y
 EOF
 tags = {
 Name = "assignment-5"
 }
}

output "instance_ip"{
 value = aws_instance.assignment-5.public_ip
}

resource "local_file" "instance_ip_file"{
 content = aws_instance.assignment-5.public_ip
 filename = "${path.module}/instance_ip.txt"
}
```

Script-5

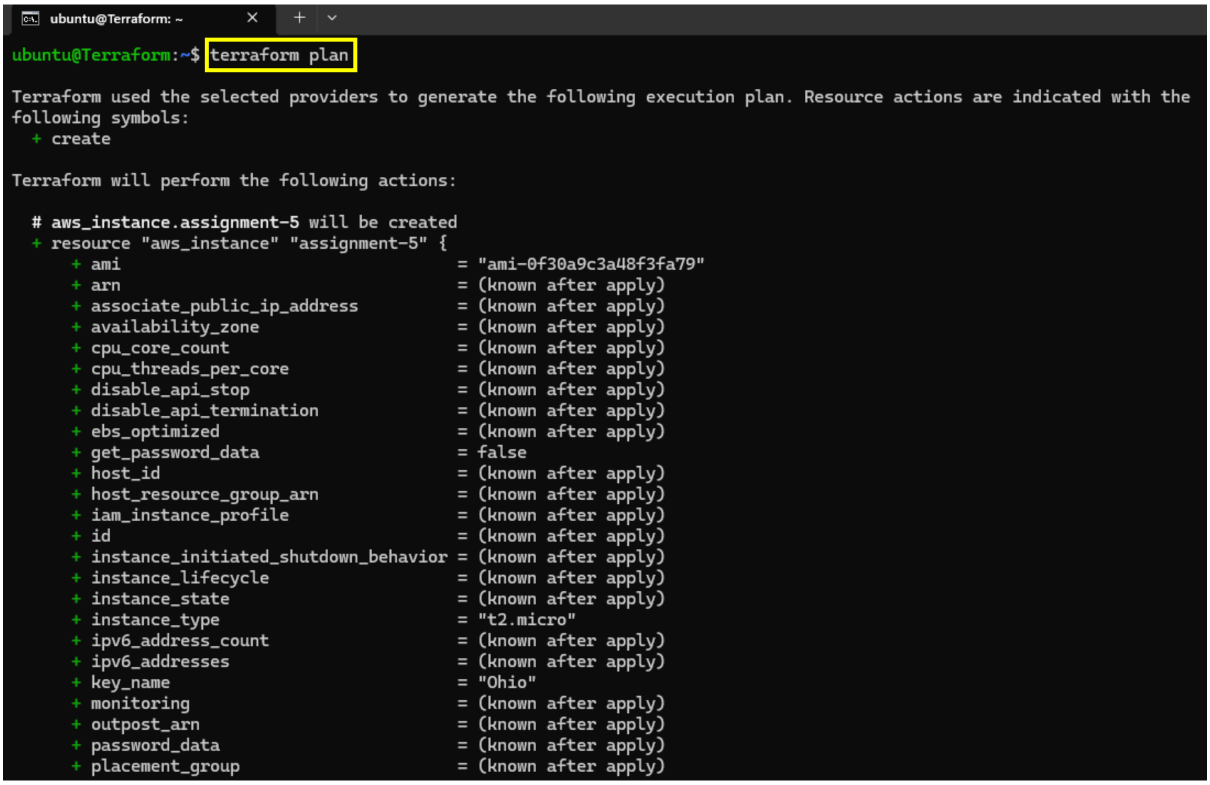
```
provider "aws"{

 region = "us-east-2"

 access_key = "AKIATCKATZTRAMZ5N6OY"

 secret_key = "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"

}


resource "aws_instance" "assignment-5"{

 ami = "ami-0f30a9c3a48f3fa79"

 instance_type = "t2.micro"

 key_name = "Ohio"
 user_data = <<-EOF
 #!bin/bash
 apt-get update -y
 apt-get install apache2 -y
 EOF
 tags = {
 Name = "assignment-5"
 }
}

output "instance_ip"{
 value = aws_instance.assignment-5.public_ip
}

resource "local_file" "instance_ip_file"{
 content = aws_instance.assignment-5.public_ip
 filename = "${path.module}/instance_ip.txt"
}
```

Terraform init -upgrade

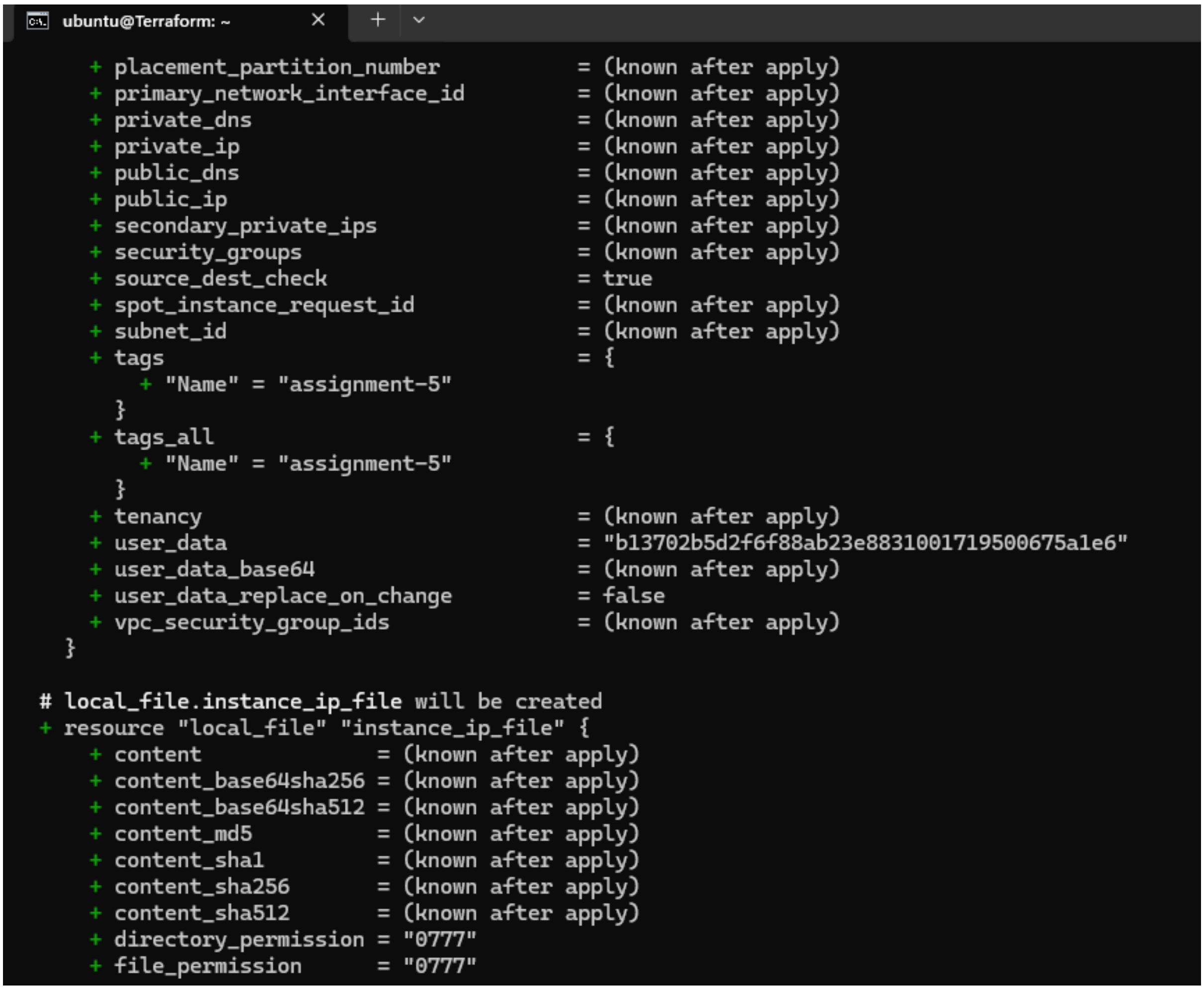


```
ubuntu@Terraform:~$ terraform init -upgrade

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/local...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/local v2.5.1...
- Installed hashicorp/local v2.5.1 (signed by HashiCorp)
- Using previously-installed hashicorp/aws v5.54.1

Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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:~$
```

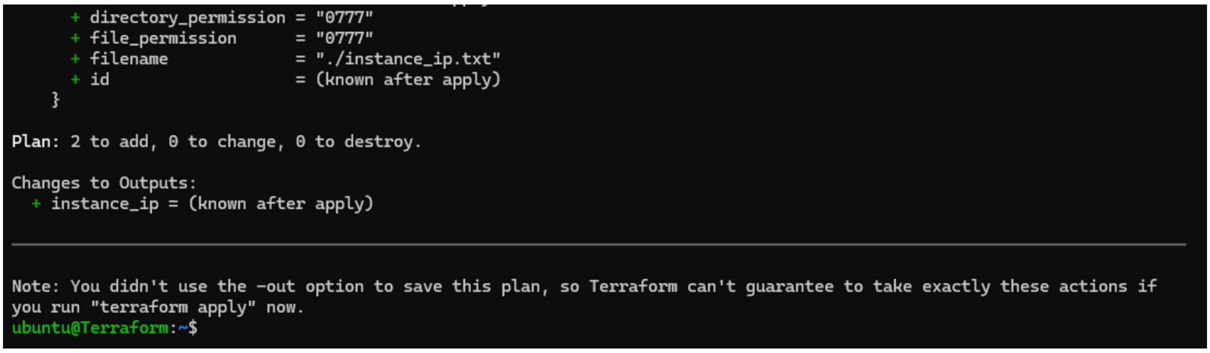
Terraform plan



```
ubuntu@Terraform:~$ 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.assignment-5 will be created
  + resource "aws_instance" "assignment-5" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
```

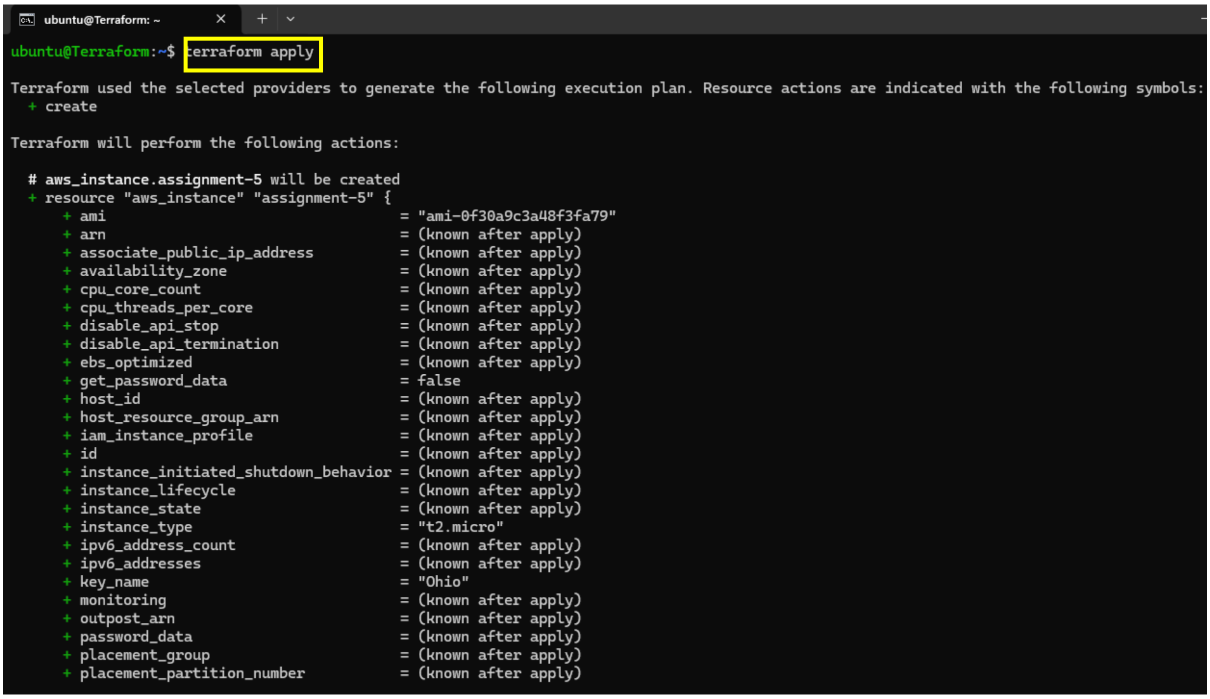


```
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "assignment-5"
        }
      + tags_all                             = {
          + "Name" = "assignment-5"
        }
      + tenancy                              = (known after apply)
      + user_data                            = "b13702b5d2f6f88ab23e8831001719500675a1e6"
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

  # local_file.instance_ip_file will be created
  + resource "local_file" "instance_ip_file" {
      + content              = (known after apply)
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
```

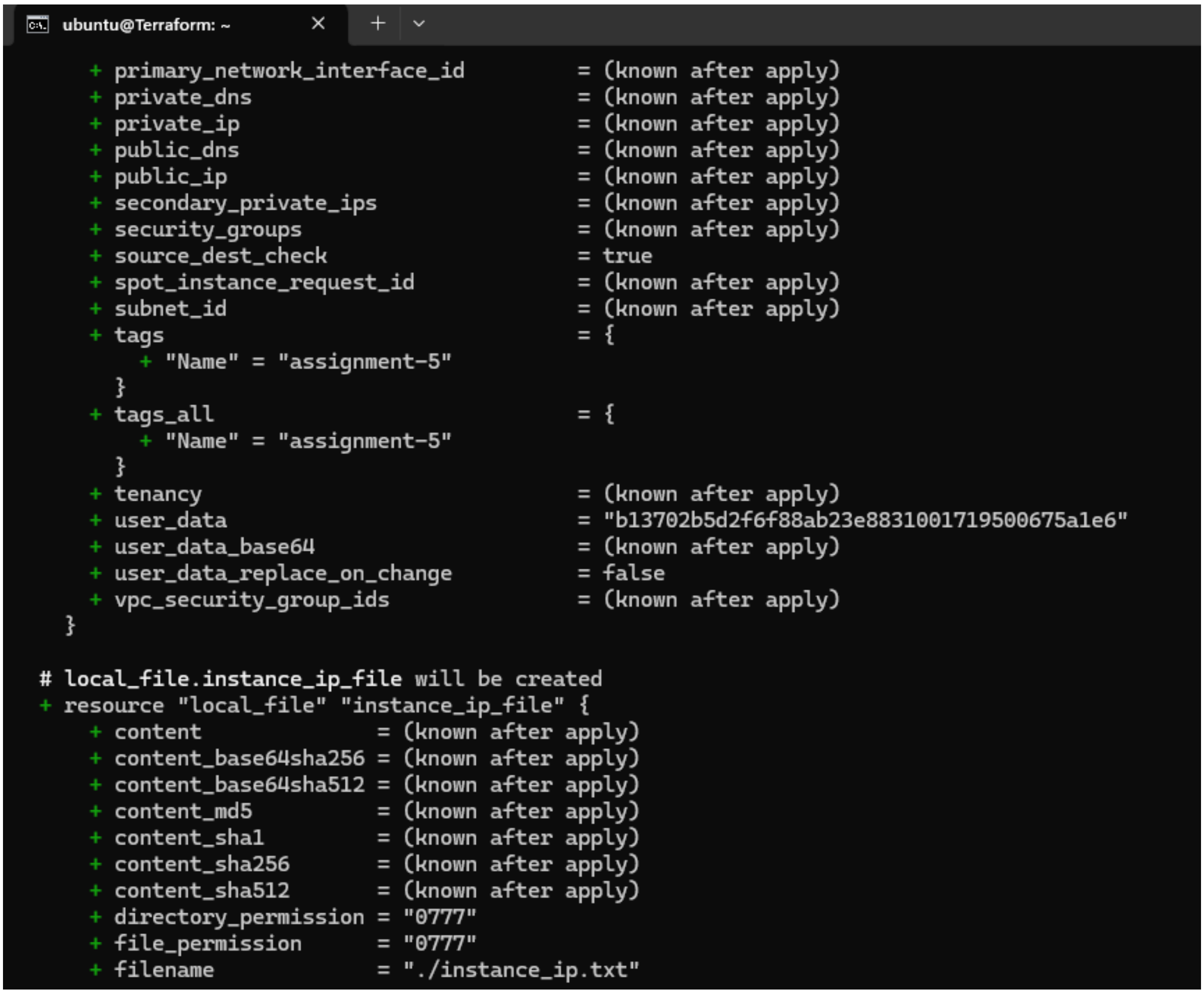


```
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "./instance_ip.txt"
      + id                   = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_ip = (known after apply)

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

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:~$
```

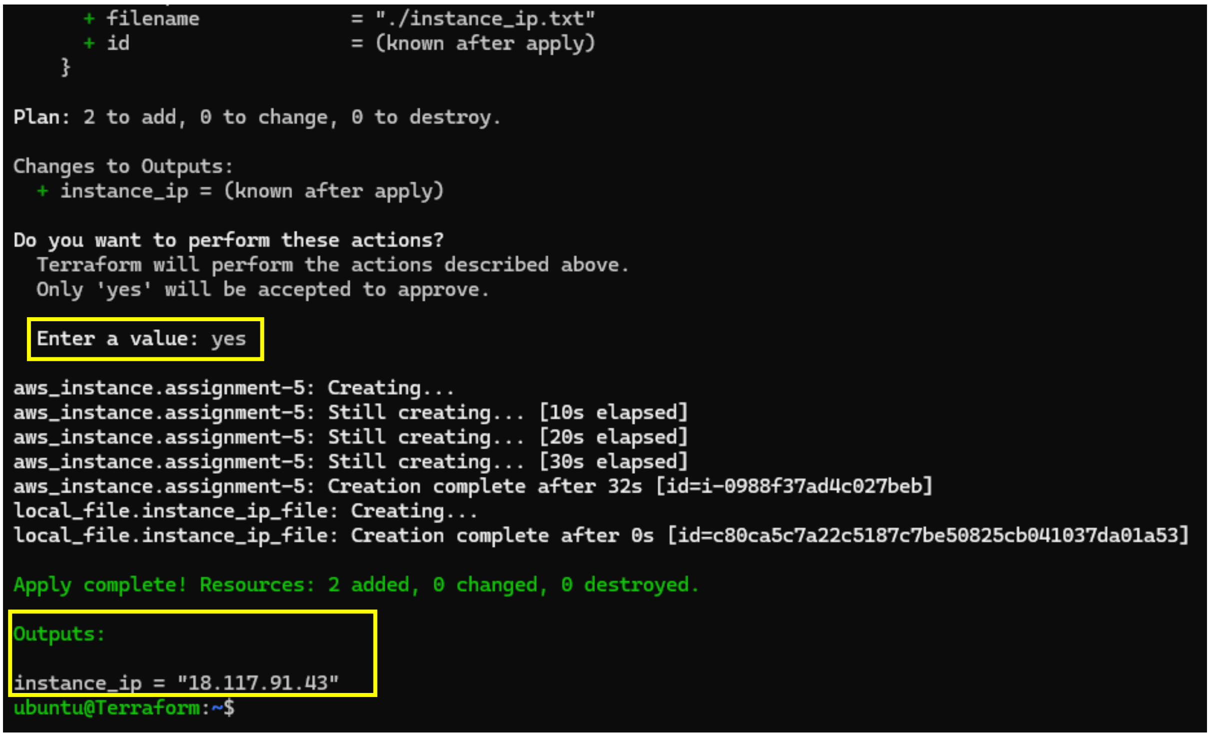
Terraform apply



```
ubuntu@Terraform:~$ 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.assignment-5 will be created
  + resource "aws_instance" "assignment-5" {
      + ami                                  = "ami-0f30a9c3a48f3fa79"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + 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                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "Ohio"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
```

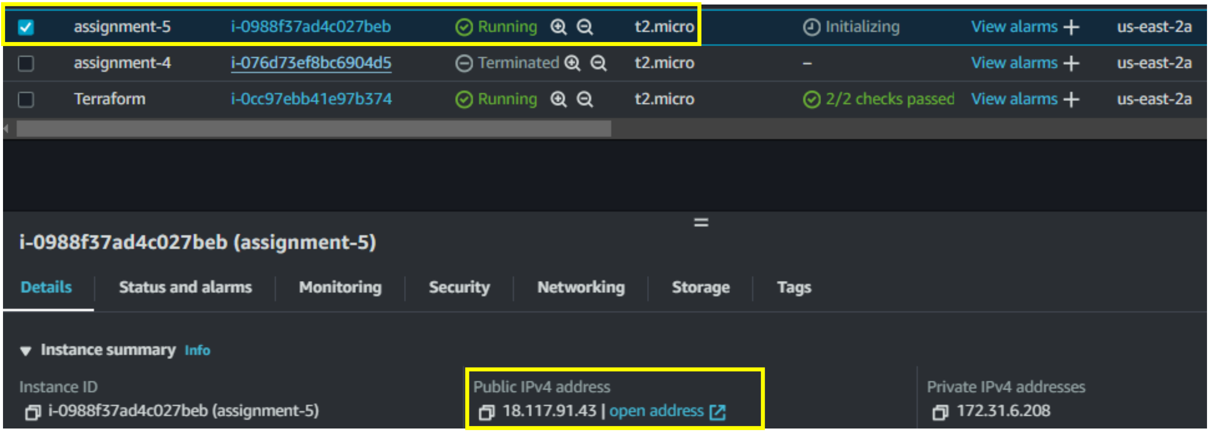


```
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "Name" = "assignment-5"
        }
      + tags_all                             = {
          + "Name" = "assignment-5"
        }
      + tenancy                              = (known after apply)
      + user_data                            = "b13702b5d2f6f88ab23e8831001719500675a1e6"
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

  # local_file.instance_ip_file will be created
  + resource "local_file" "instance_ip_file" {
      + content              = (known after apply)
      + content_base64sha256 = (known after apply)
      + content_base64sha512 = (known after apply)
      + content_md5          = (known after apply)
      + content_sha1         = (known after apply)
      + content_sha256       = (known after apply)
      + content_sha512       = (known after apply)
      + directory_permission = "0777"
      + file_permission      = "0777"
      + filename             = "./instance_ip.txt"
```



```
      + filename             = "./instance_ip.txt"
      + id                   = (known after apply)
    }

Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_ip = (known after apply)

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.assignment-5: Creating...
aws_instance.assignment-5: Still creating... [10s elapsed]
aws_instance.assignment-5: Still creating... [20s elapsed]
aws_instance.assignment-5: Still creating... [30s elapsed]
aws_instance.assignment-5: Creation complete after 32s [id=i-0988f37ad4c027beb]
local_file.instance_ip_file: Creating...
local_file.instance_ip_file: Creation complete after 0s [id=c80ca5c7a22c5187c7be50825cb041037da01a53]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

instance_ip = "18.117.91.43"
ubuntu@Terraform:~$
```

Now its running state



☑	assignment-5	i-0988f37ad4c027beb	Running	t2.micro	Initializing	View alarms +	us-east-2a
□	assignment-4	i-076d73ef8bc6904d5	Terminated	t2.micro	–	View alarms +	us-east-2a
□	Terraform	i-0cc97ebb41e97b374	Running	t2.micro	2/2 checks passed	View alarms +	us-east-2a

i-0988f37ad4c027beb (assignment-5)

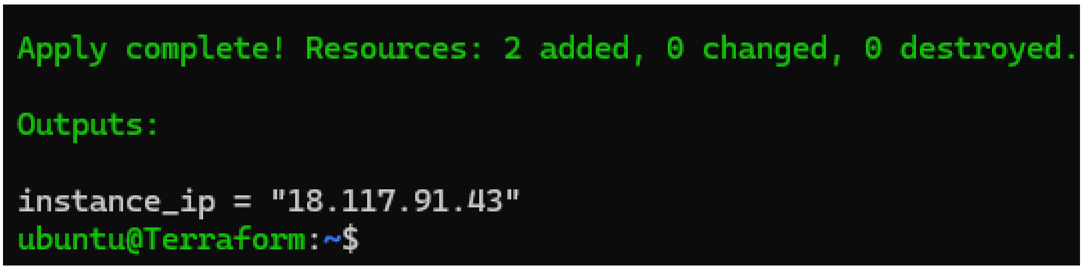
▼ Instance summary Info

Instance ID: i-0988f37ad4c027beb (assignment-5)

Public IPv4 address: 18.117.91.43 | open address

Private IPv4 addresses: 172.31.6.208

Output, According to our assignment task



```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

instance_ip = "18.117.91.43"
ubuntu@Terraform:~$
```