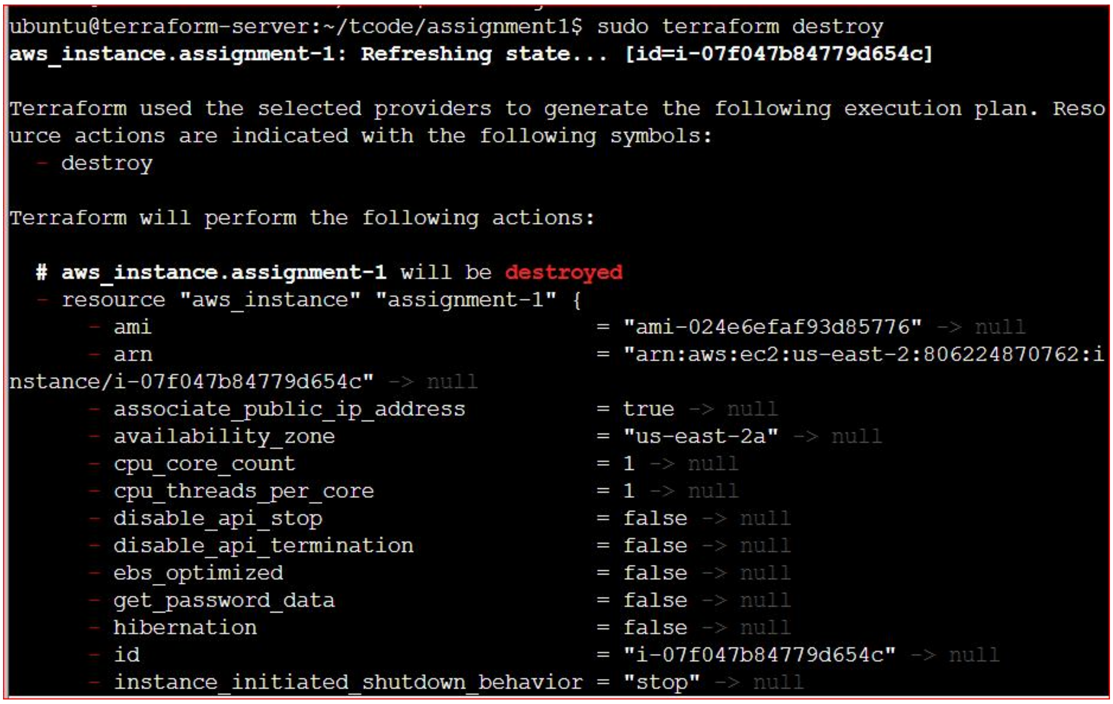
Module 8: Terraform Assignment - 2

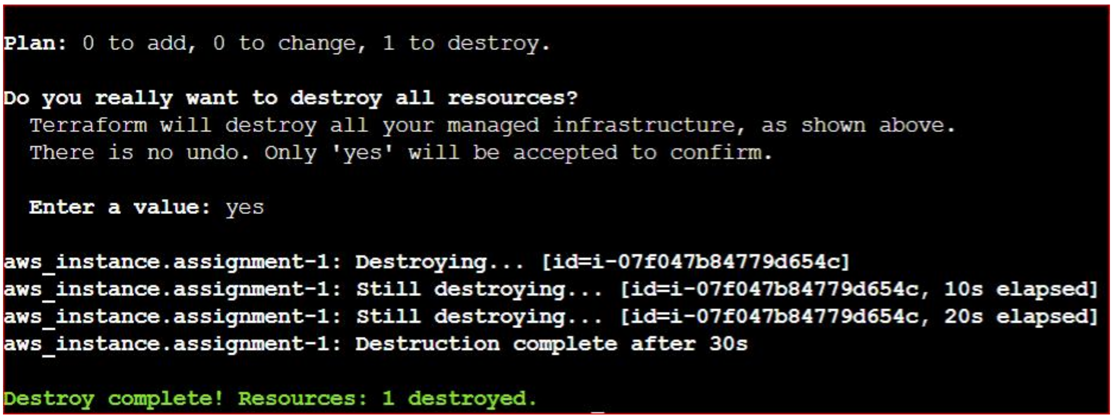
1. Destroy the previous deployment
2. Create a new EC2 instance with an Elastic IP

Solution:-

```
$ sudo terraform destroy
```

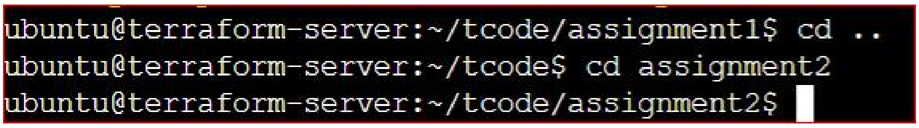


```
ubuntu@terraform-server:~/tcode/assignment1$ sudo terraform destroy
aws_instance.assignment-1: Refreshing state... [id=i-07f047b84779d654c]

Terraform used the selected providers to generate the following execution plan. Reso
urce 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-024e6efaf93d85776" -> null
      - arn                                  = "arn:aws:ec2:us-east-2:806224870762:i
nstance/i-07f047b84779d654c" -> 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-07f047b84779d654c" -> null
      - instance_initiated_shutdown_behavior = "stop" -> 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-07f047b84779d654c]
aws_instance.assignment-1: Still destroying... [id=i-07f047b84779d654c, 10s elapsed]
aws_instance.assignment-1: Still destroying... [id=i-07f047b84779d654c, 20s elapsed]
aws_instance.assignment-1: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
```

```
$ cd ..
$ cd assignment2
```



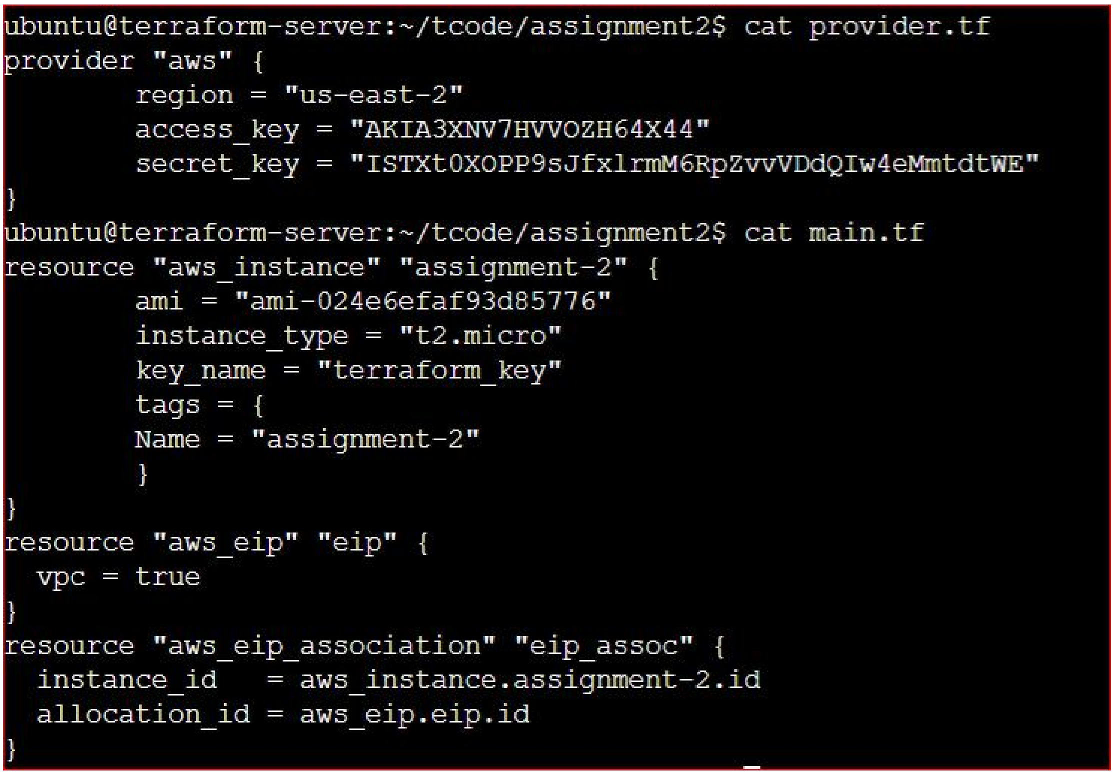
```
ubuntu@terraform-server:~/tcode/assignment1$ cd ..
ubuntu@terraform-server:~/tcode$ cd assignment2
ubuntu@terraform-server:~/tcode/assignment2$
```

```
$ vi provider.tf
```

```
provider "aws" {
        region = "us-east-2"
        access_key = "AKIA3XNV7HVVOZH64X44"
        secret_key =
"ISTXt0XOPP9sJfxlrmM6RpZvvVDdQlw4eMmtd
tWE"
}
```

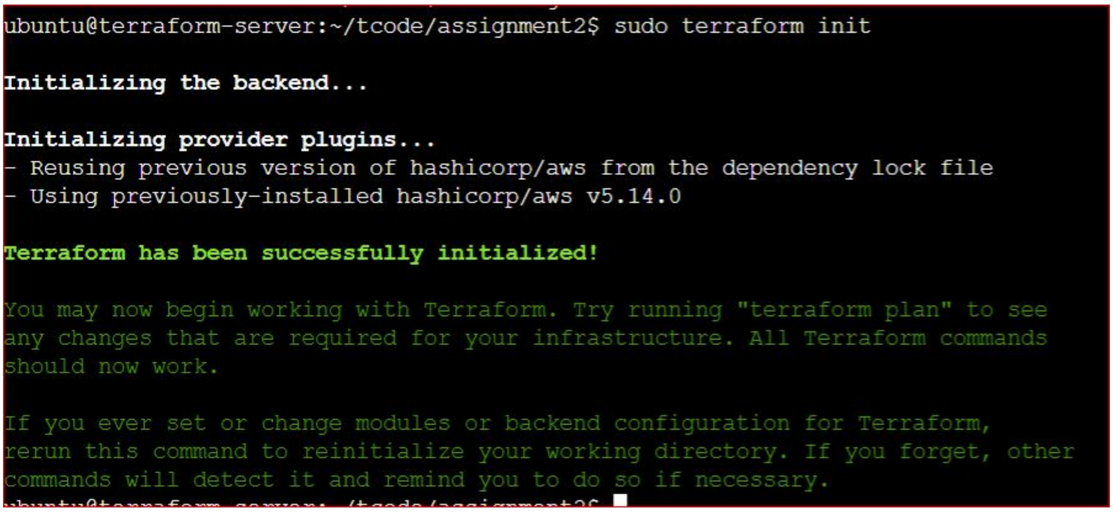
```
$ vi main.tf
```

```
resource "aws_instance" "assignment-2" {
         ami = "ami-024e6efaf93d85776"
         instance_type = "t2.micro"
         key_name = "terraform_key"
         tags = {
         Name = "assignment-2”
         }
}
resource "aws_eip" "eip" {
  vpc = true
}
resource "aws_eip_association" "eip_assoc" {
 instance_id = aws_instance.assignment-2.id
 allocation_id = aws_eip.eip.id
}
```



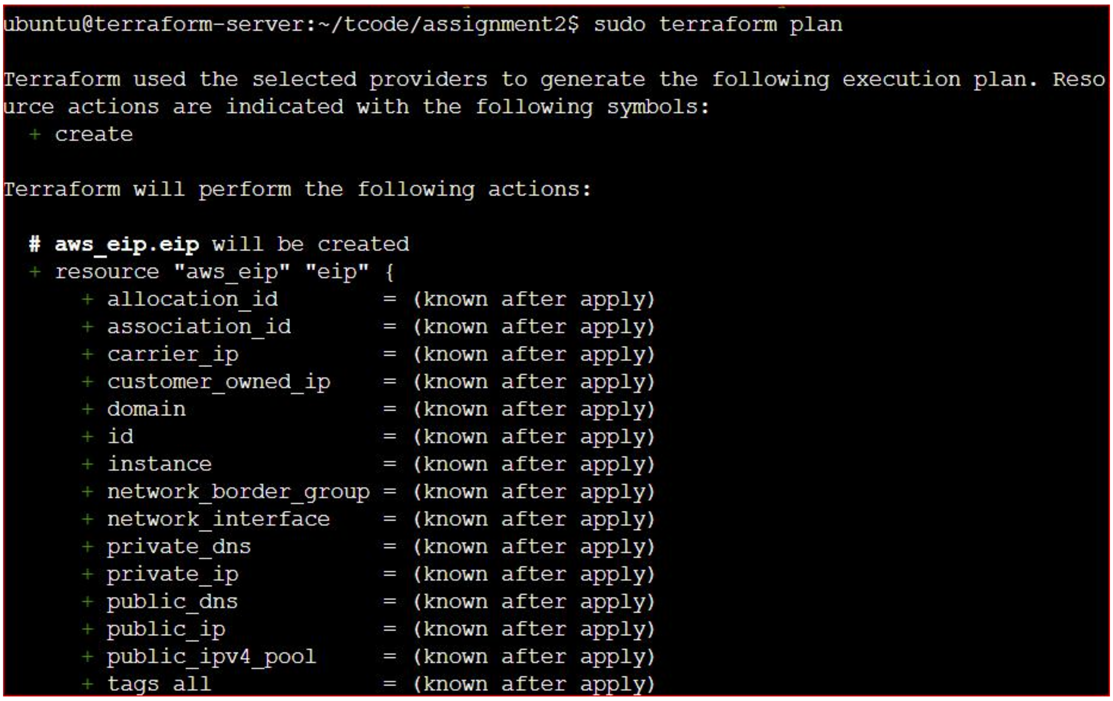
```
ubuntu@terraform-server:~/tcode/assignment2$ cat provider.tf
provider "aws" {
        region = "us-east-2"
        access_key = "AKIA3XNV7HVVOZH64X44"
        secret_key = "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"
}
ubuntu@terraform-server:~/tcode/assignment2$ cat main.tf
resource "aws_instance" "assignment-2" {
        ami = "ami-024e6efaf93d85776"
        instance_type = "t2.micro"
        key_name = "terraform_key"
        tags = {
        Name = "assignment-2"
        }
}
resource "aws_eip" "eip" {
  vpc = true
}
resource "aws_eip_association" "eip_assoc" {
  instance_id   = aws_instance.assignment-2.id
  allocation_id = aws_eip.eip.id
}
```

```
$ sudo terraform init
```

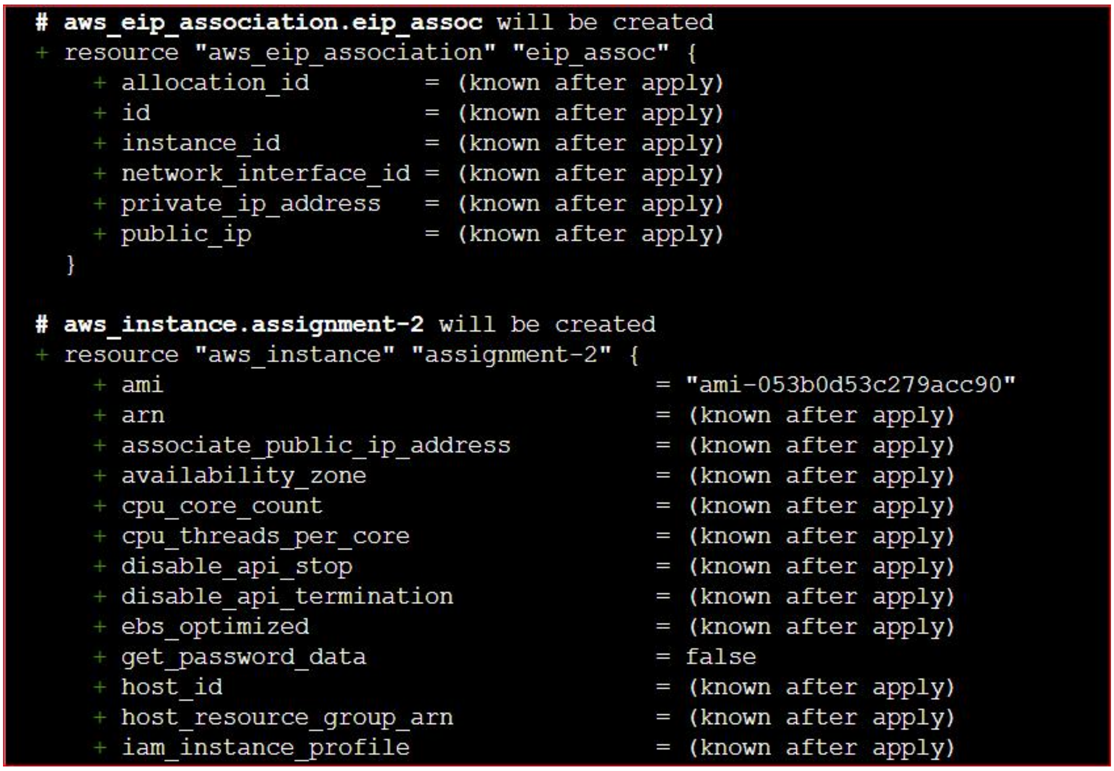


```
ubuntu@terraform-server:~/tcode/assignment2$ sudo terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.14.0

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-server:~/tcode/assignment2$
```

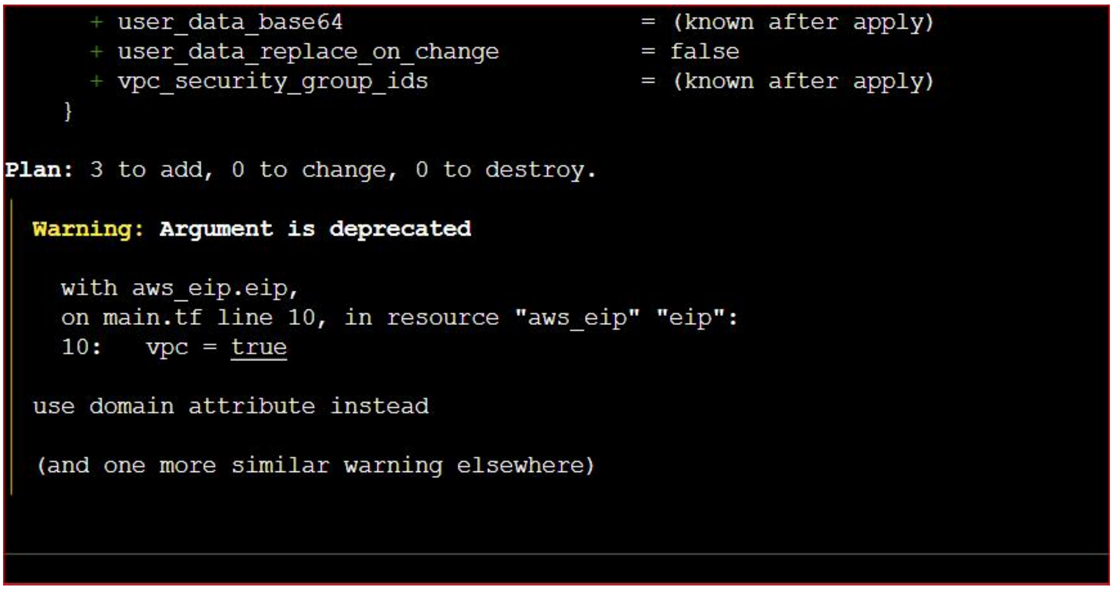
$ sudo terraform plan



```
ubuntu@terraform-server:~/tcode/assignment2$ sudo terraform plan

Terraform used the selected providers to generate the following execution plan. Reso
urce 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)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
      + customer_owned_ip    = (known after apply)
      + domain               = (known after apply)
      + id                   = (known after apply)
      + instance             = (known after apply)
      + network_border_group = (known after apply)
      + network_interface    = (known after apply)
      + private_dns          = (known after apply)
      + private_ip           = (known after apply)
      + public_dns           = (known after apply)
      + public_ip            = (known after apply)
      + public_ipv4_pool     = (known after apply)
      + tags_all             = (known after apply)
```

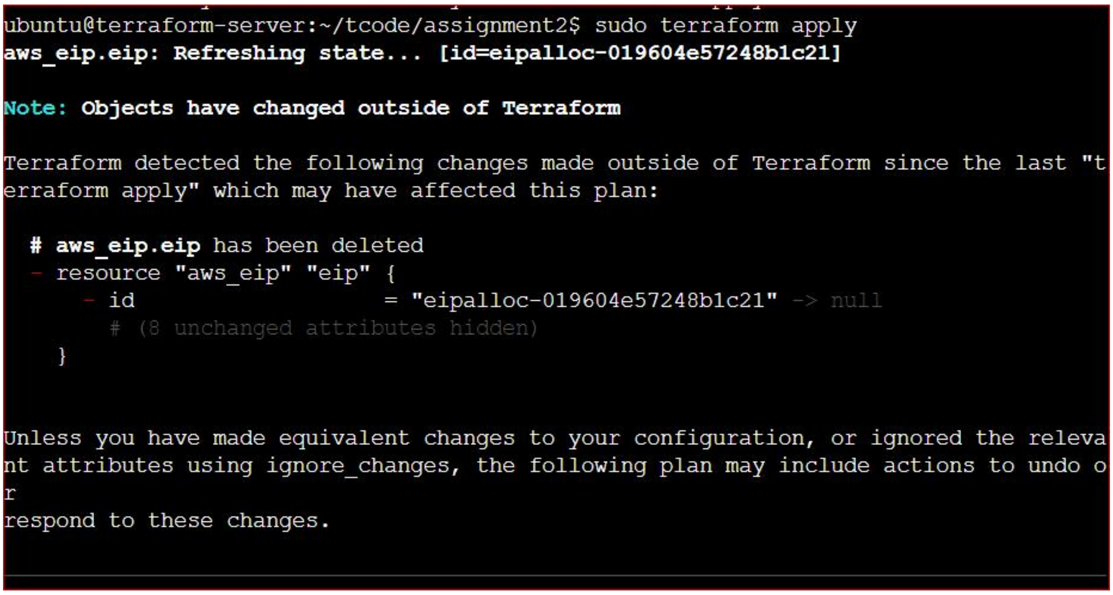


```
  # aws_eip_association.eip_assoc will be created
  + resource "aws_eip_association" "eip_assoc" {
      + allocation_id        = (known after apply)
      + id                   = (known after apply)
      + instance_id          = (known after apply)
      + network_interface_id = (known after apply)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

  # aws_instance.assignment-2 will be created
  + resource "aws_instance" "assignment-2" {
      + ami                                  = "ami-053b0d53c279acc90"
      + 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)
```

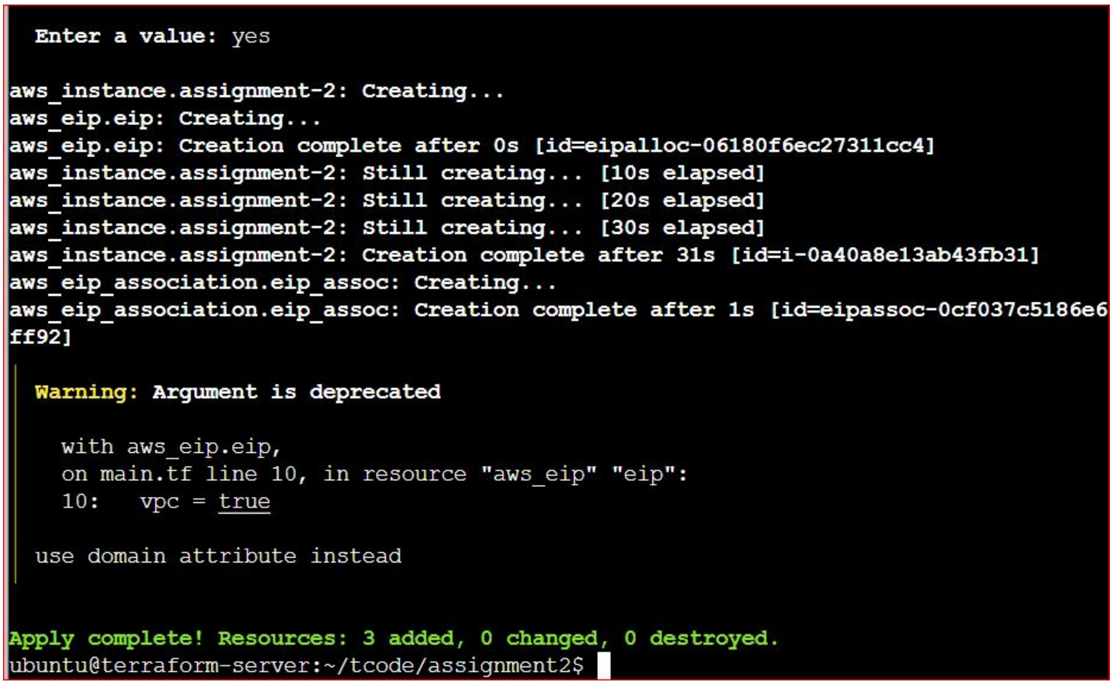


```
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

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

  Warning: Argument is deprecated

    with aws_eip.eip,
    on main.tf line 10, in resource "aws_eip" "eip":
    10:    vpc = true

  use domain attribute instead

  (and one more similar warning elsewhere)
```

$ sudo terraform apply

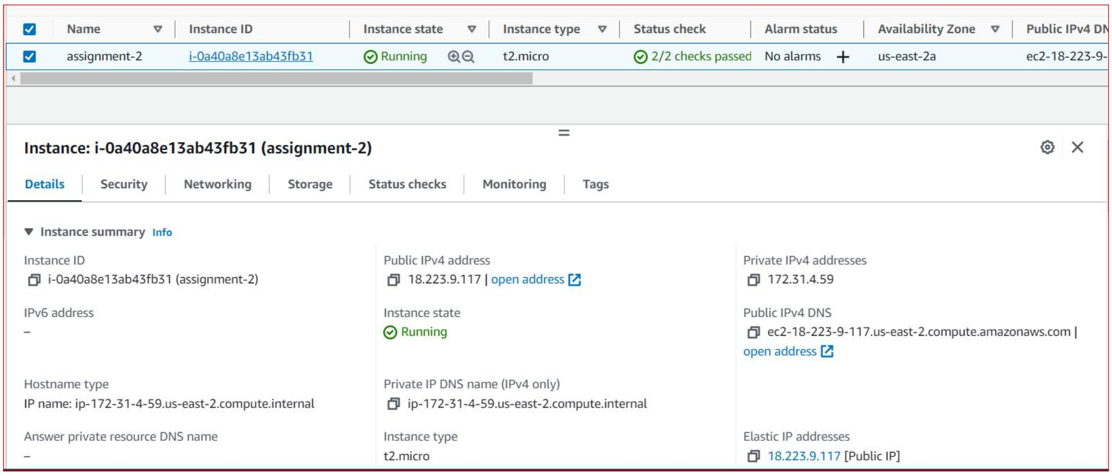


```
ubuntu@terraform-server:~/tcode/assignment2$ sudo terraform apply
aws_eip.eip: Refreshing state... [id=eipalloc-019604e57248b1c21]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "t
erraform apply" which may have affected this plan:

  # aws_eip.eip has been deleted
  - resource "aws_eip" "eip" {
      - id                   = "eipalloc-019604e57248b1c21" -> null
        # (8 unchanged attributes hidden)
    }


Unless you have made equivalent changes to your configuration, or ignored the releva
nt attributes using ignore_changes, the following plan may include actions to undo o
r
respond to these changes.
```



```
  Enter a value: yes

aws_instance.assignment-2: Creating...
aws_eip.eip: Creating...
aws_eip.eip: Creation complete after 0s [id=eipalloc-06180f6ec27311cc4]
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-0a40a8e13ab43fb31]
aws_eip_association.eip_assoc: Creating...
aws_eip_association.eip_assoc: Creation complete after 1s [id=eipassoc-0cf037c5186e6
ff92]

  Warning: Argument is deprecated

    with aws_eip.eip,
    on main.tf line 10, in resource "aws_eip" "eip":
    10:   vpc = true

  use domain attribute instead


Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@terraform-server:~/tcode/assignment2$
```

* Instance created present in AWS console with their Elastic IP:-



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
☑	assignment-2	i-0a40a8e13ab43fb31	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms	us-east-2a	ec2-18-223-9-

Instance: i-0a40a8e13ab43fb31 (assignment-2)

▼ Instance summary Info

Instance ID
i-0a40a8e13ab43fb31 (assignment-2)

Public IPv4 address
18.223.9.117 | open address

Private IPv4 addresses
172.31.4.59

IPv6 address
–

Instance state
✓ Running

Public IPv4 DNS
ec2-18-223-9-117.us-east-2.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-4-59.us-east-2.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-4-59.us-east-2.compute.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
18.223.9.117 [Public IP]