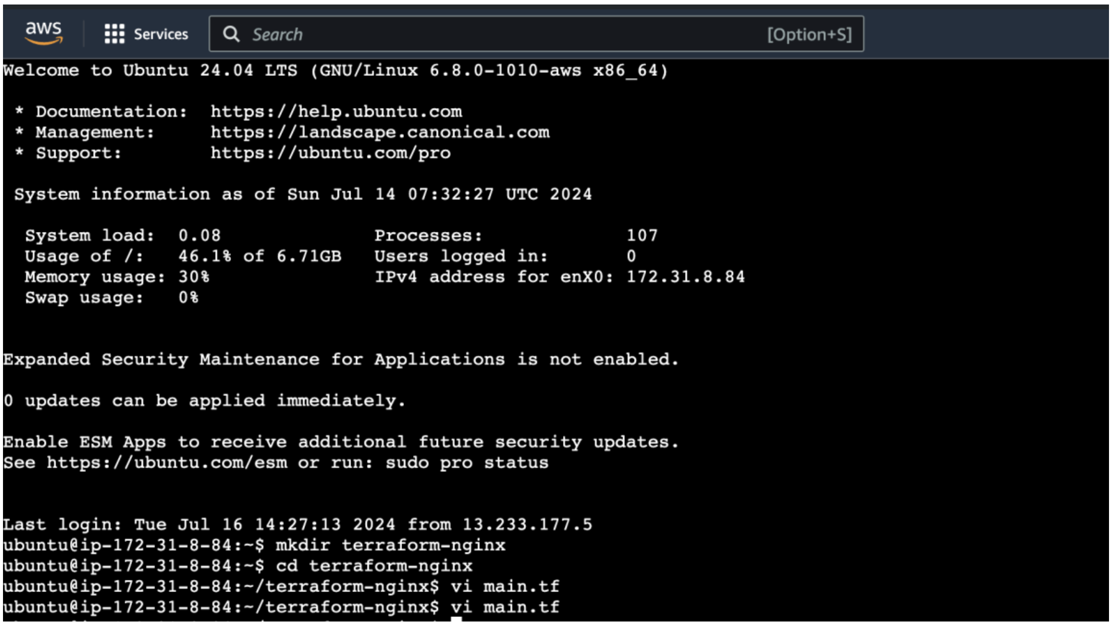
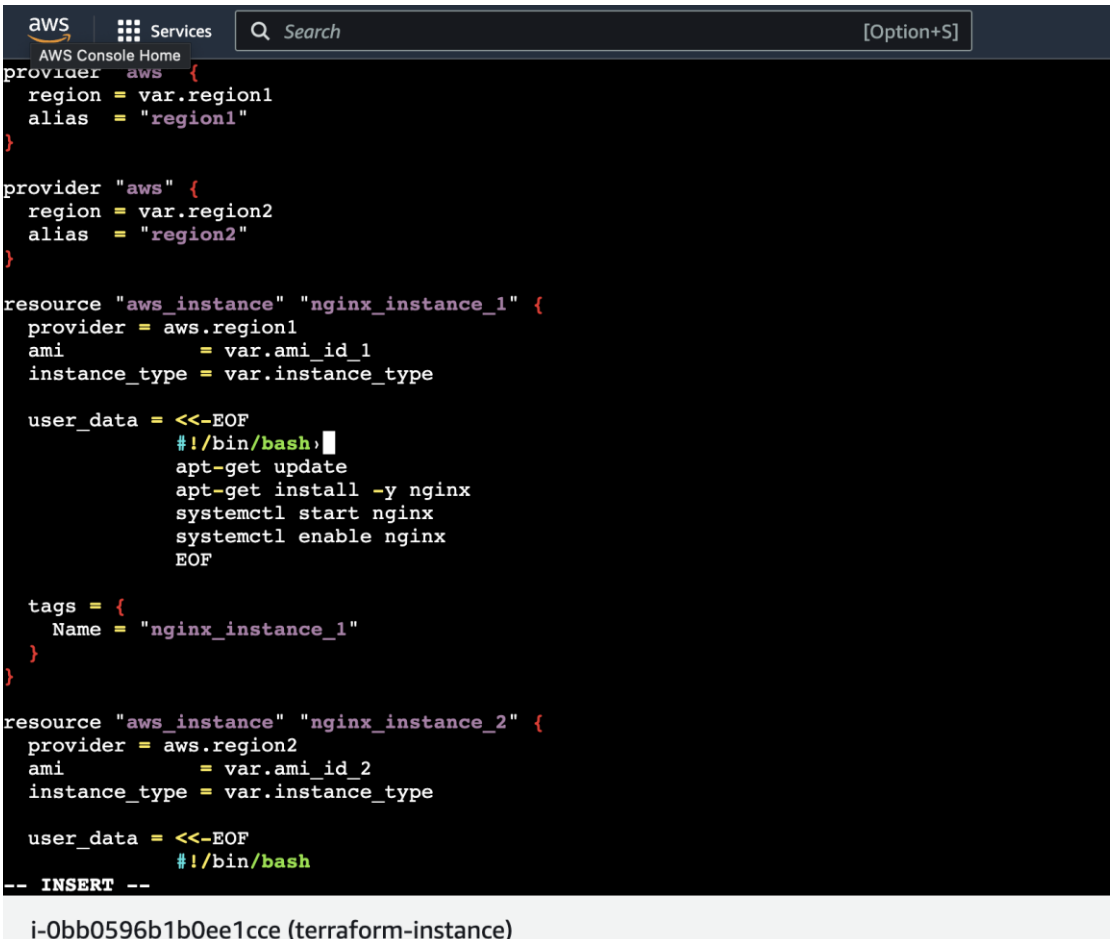
Task – 18

Creating a new directory and writing the main.tf file:

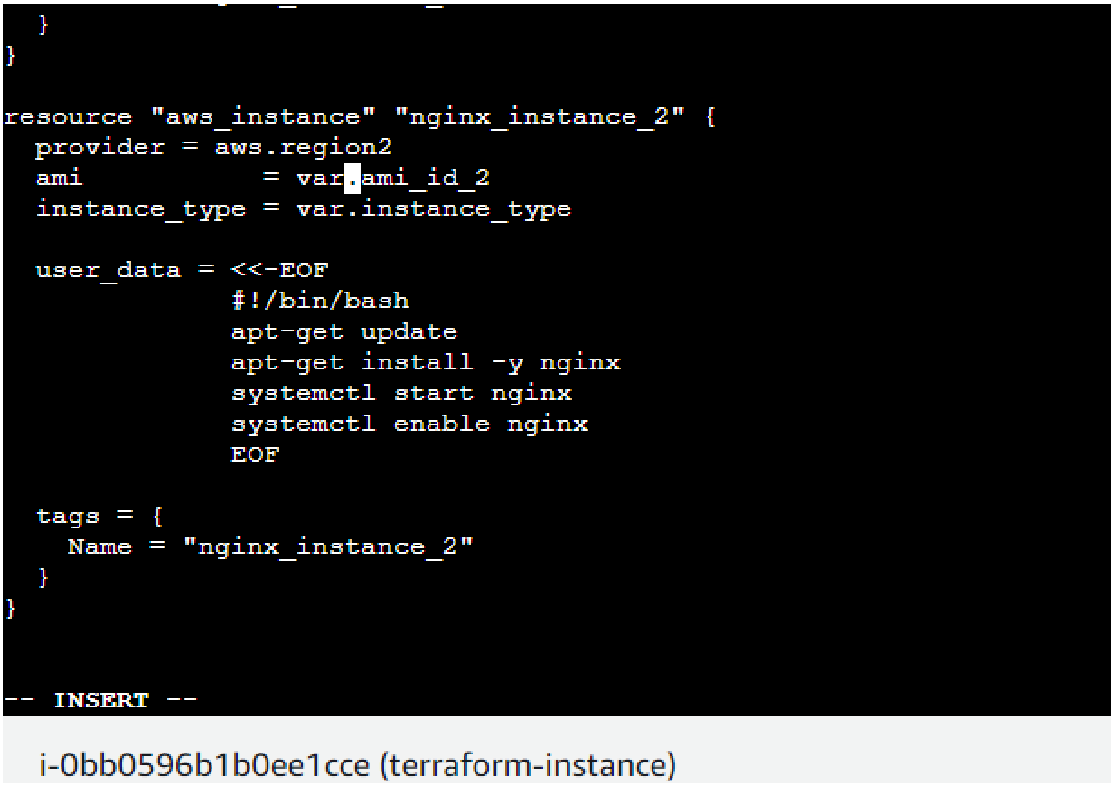


```
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1010-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Jul 14 07:32:27 UTC 2024

  System load:  0.08              Processes:             107
  Usage of /:   46.1% of 6.71GB   Users logged in:       0
  Memory usage: 30%               IPv4 address for enX0: 172.31.8.84
  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


Last login: Tue Jul 16 14:27:13 2024 from 13.233.177.5
ubuntu@ip-172-31-8-84:~$ mkdir terraform-nginx
ubuntu@ip-172-31-8-84:~$ cd terraform-nginx
ubuntu@ip-172-31-8-84:~/terraform-nginx$ vi main.tf
ubuntu@ip-172-31-8-84:~/terraform-nginx$ vi main.tf
```



```
provider "aws" {
  region = var.region1
  alias  = "region1"
}

provider "aws" {
  region = var.region2
  alias  = "region2"
}

resource "aws_instance" "nginx_instance_1" {
  provider = aws.region1
  ami           = var.ami_id_1
  instance_type = var.instance_type

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

  tags = {
    Name = "nginx_instance_1"
  }
}

resource "aws_instance" "nginx_instance_2" {
  provider = aws.region2
  ami           = var.ami_id_2
  instance_type = var.instance_type

  user_data = <<-EOF
              #!/bin/bash
-- INSERT --
```

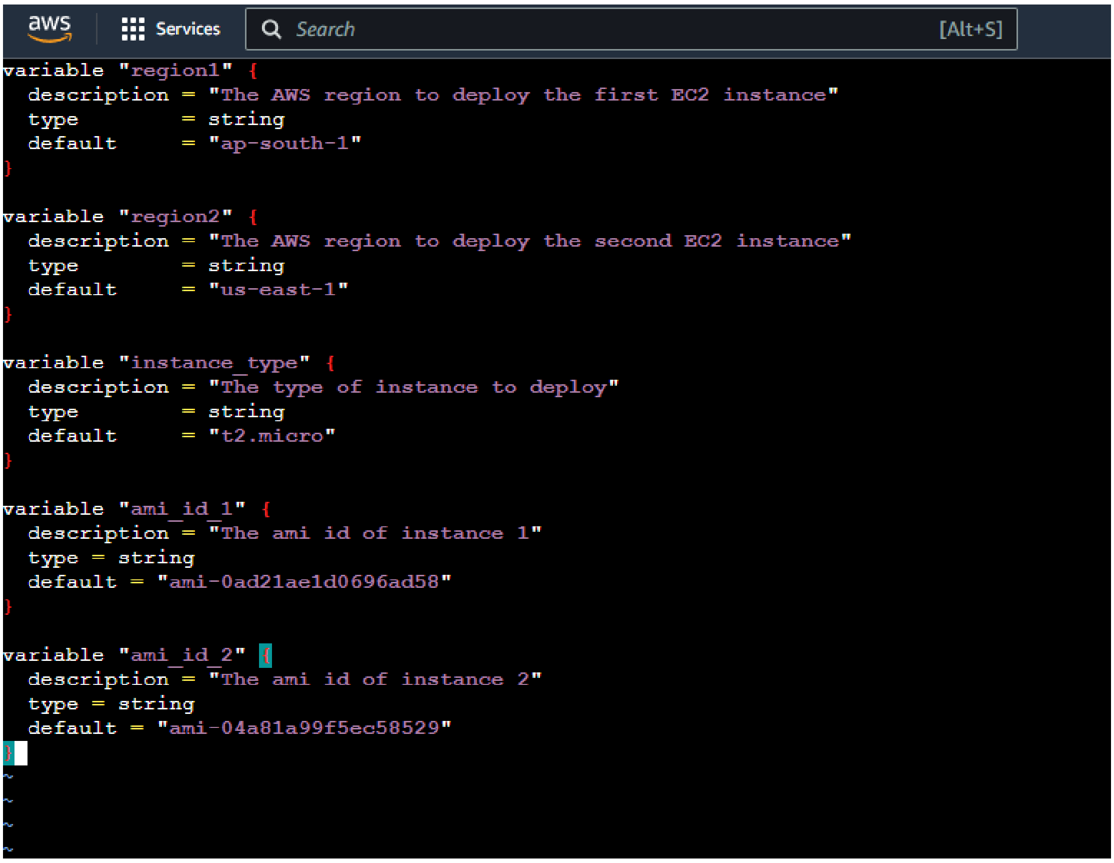
i-0bb0596b1b0ee1cce (terraform-instance)



```
  }
}

resource "aws_instance" "nginx_instance_2" {
  provider = aws.region2
  ami           = var.ami_id_2
  instance_type = var.instance_type

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

  tags = {
    Name = "nginx_instance_2"
  }
}


-- INSERT --
```

i-0bb0596b1b0ee1cce (terraform-instance)

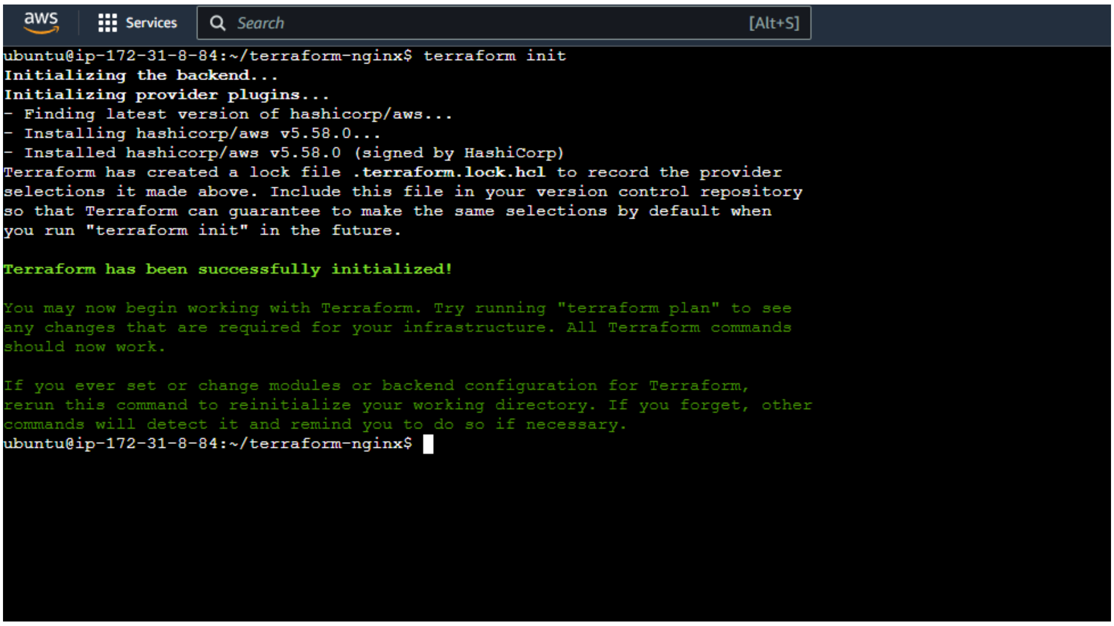
Writing variables.tf file



```
variable "region1" {
  description = "The AWS region to deploy the first EC2 instance"
  type        = string
  default     = "ap-south-1"
}

variable "region2" {
  description = "The AWS region to deploy the second EC2 instance"
  type        = string
  default     = "us-east-1"
}

variable "instance_type" {
  description = "The type of instance to deploy"
  type        = string
  default     = "t2.micro"
}

variable "ami_id_1" {
  description = "The ami id of instance 1"
  type = string
  default = "ami-0ad21ae1d0696ad58"
}

variable "ami_id_2" {
  description = "The ami id of instance 2"
  type = string
  default = "ami-04a81a99f5ec58529"
}
~
~
~
~
```

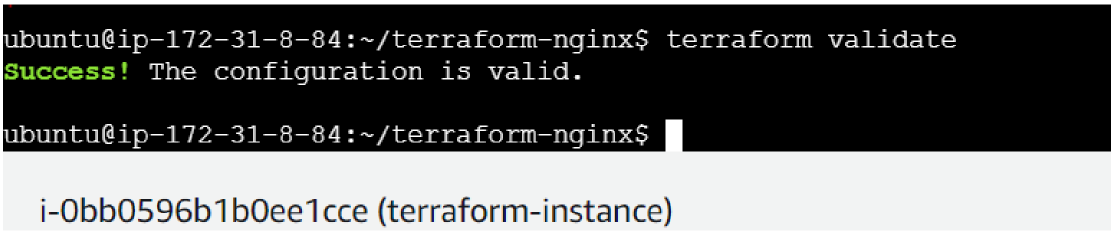
Terraform Initializing and applying

Terraform Init:



```
ubuntu@ip-172-31-8-84:~/terraform-nginx$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.58.0...
- Installed hashicorp/aws v5.58.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-8-84:~/terraform-nginx$
```

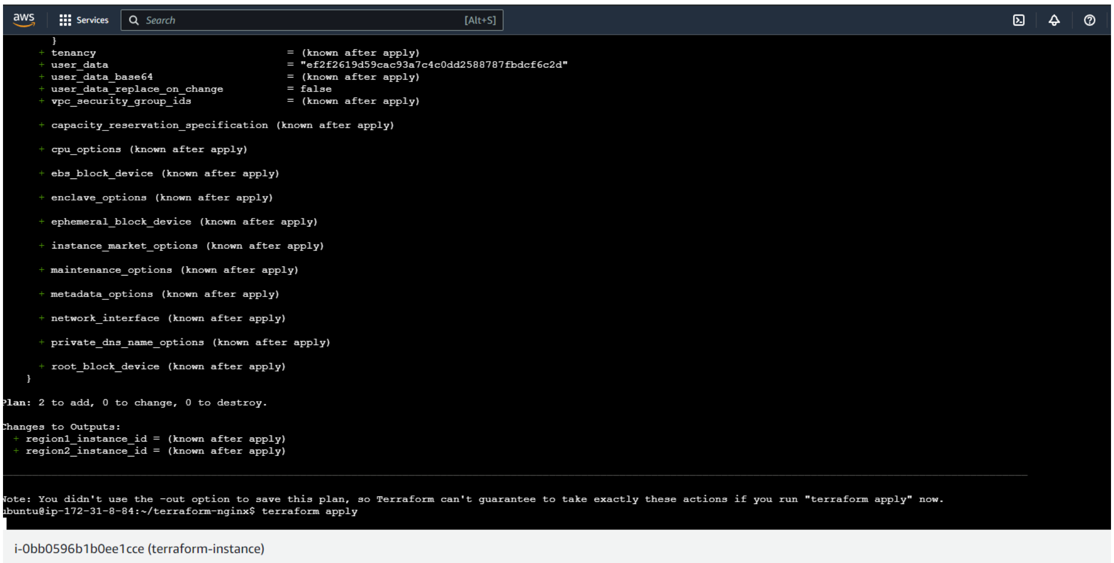
Terraform validate:



```
ubuntu@ip-172-31-8-84:~/terraform-nginx$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-8-84:~/terraform-nginx$
```

i-0bb0596b1b0ee1cce (terraform-instance)

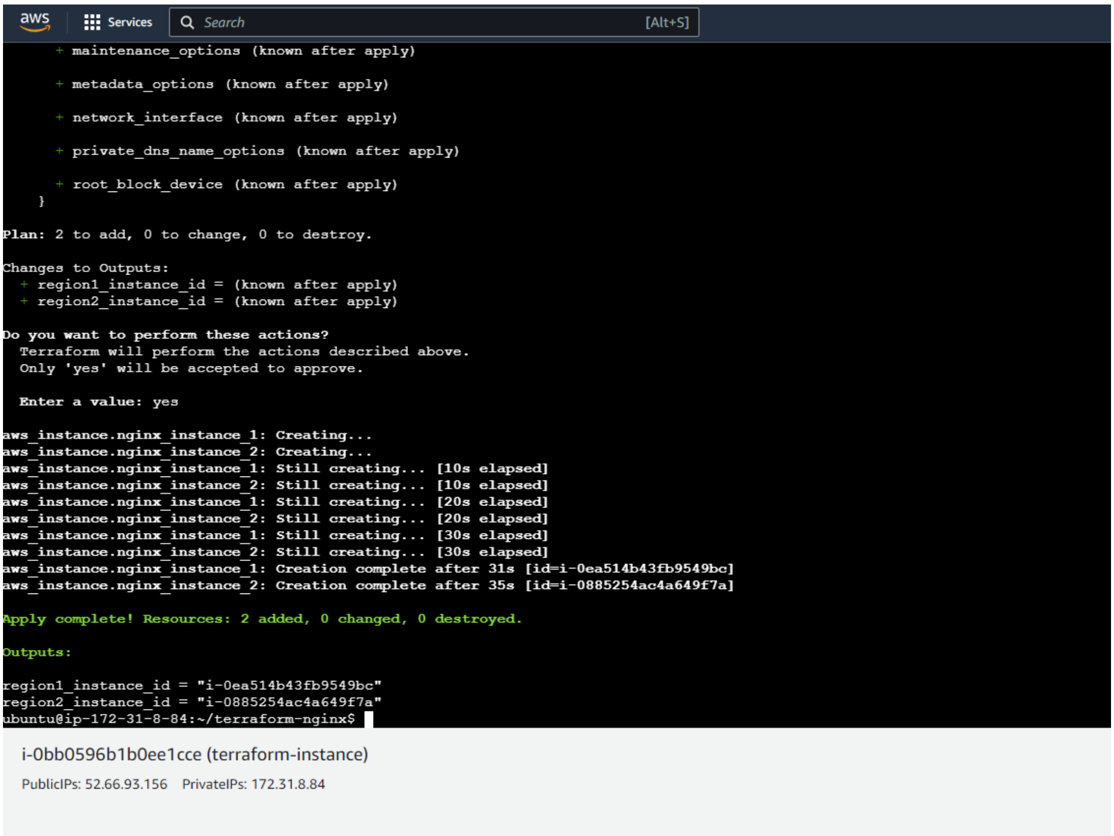
Terraform plan:



```
      }
    + tenancy                              = (known after apply)
    + user_data                            = "ef2f2619d59cac93a7c4c0dd2588787fbdcf6c2d"
    + user_data_base64                     = (known after apply)
    + user_data_replace_on_change          = false
    + vpc_security_group_ids               = (known after apply)

    + capacity_reservation_specification (known after apply)

    + cpu_options (known after apply)

    + ebs_block_device (known after apply)

    + enclave_options (known after apply)

    + ephemeral_block_device (known after apply)

    + instance_market_options (known after apply)

    + maintenance_options (known after apply)

    + metadata_options (known after apply)

    + network_interface (known after apply)

    + private_dns_name_options (known after apply)

    + root_block_device (known after apply)
  }

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

Changes to Outputs:
  + region1_instance_id = (known after apply)
  + region2_instance_id = (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@ip-172-31-8-84:~/terraform-nginx$ terraform apply
```

i-0bb0596b1b0ee1cce (terraform-instance)

Terraform apply:



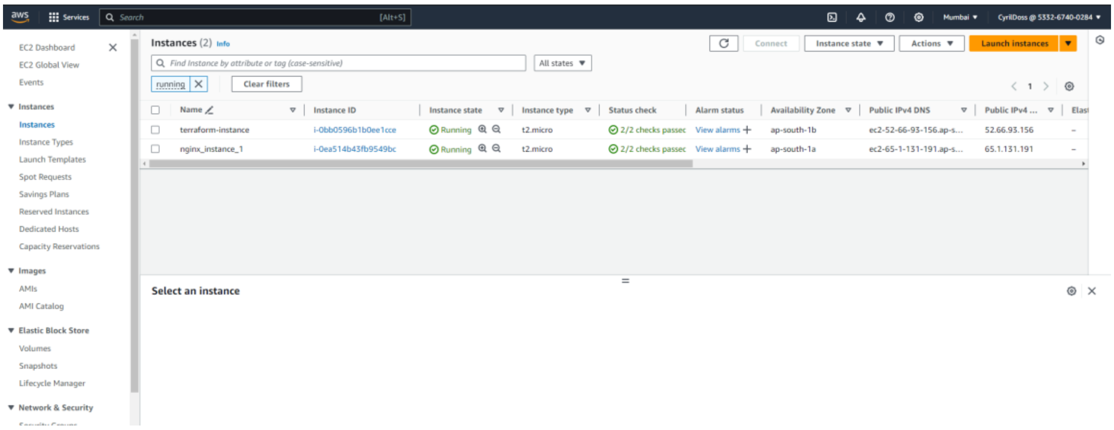
```
    + maintenance_options (known after apply)

    + metadata_options (known after apply)

    + network_interface (known after apply)

    + private_dns_name_options (known after apply)

    + root_block_device (known after apply)
  }

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

Changes to Outputs:
  + region1_instance_id = (known after apply)
  + region2_instance_id = (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.nginx_instance_1: Creating...
aws_instance.nginx_instance_2: Creating...
aws_instance.nginx_instance_1: Still creating... [10s elapsed]
aws_instance.nginx_instance_2: Still creating... [10s elapsed]
aws_instance.nginx_instance_1: Still creating... [20s elapsed]
aws_instance.nginx_instance_2: Still creating... [20s elapsed]
aws_instance.nginx_instance_1: Still creating... [30s elapsed]
aws_instance.nginx_instance_2: Still creating... [30s elapsed]
aws_instance.nginx_instance_1: Creation complete after 31s [id=i-0ea514b43fb9549bc]
aws_instance.nginx_instance_2: Creation complete after 35s [id=i-0885254ac4a649f7a]

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

Outputs:

region1_instance_id = "i-0ea514b43fb9549bc"
region2_instance_id = "i-0885254ac4a649f7a"
ubuntu@ip-172-31-8-84:~/terraform-nginx$
```

i-0bb0596b1b0ee1cce (terraform-instance)

PublicIPs: 52.66.93.156 PrivateIPs: 172.31.8.84

Output

Instance 1:

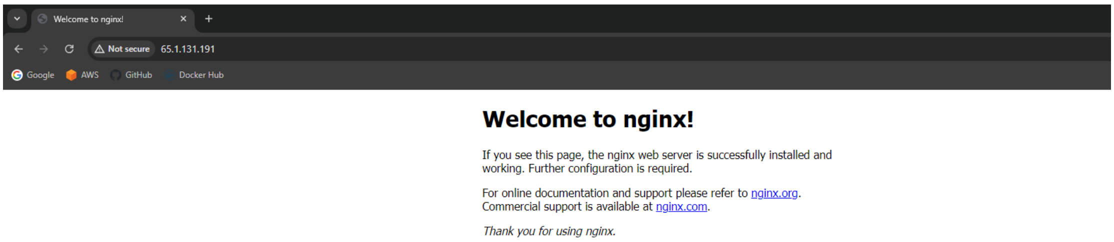


Instances (2)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elast
terraform-instance	i-0bb0596b1b0ee1cce	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-52-66-93-156.ap-s...	52.66.93.156	–
nginx_instance_1	i-0ea514b43fb9549bc	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-65-1-131-191.ap-s...	65.1.131.191	–

Select an instance

Accessing Nginx in instance 1:



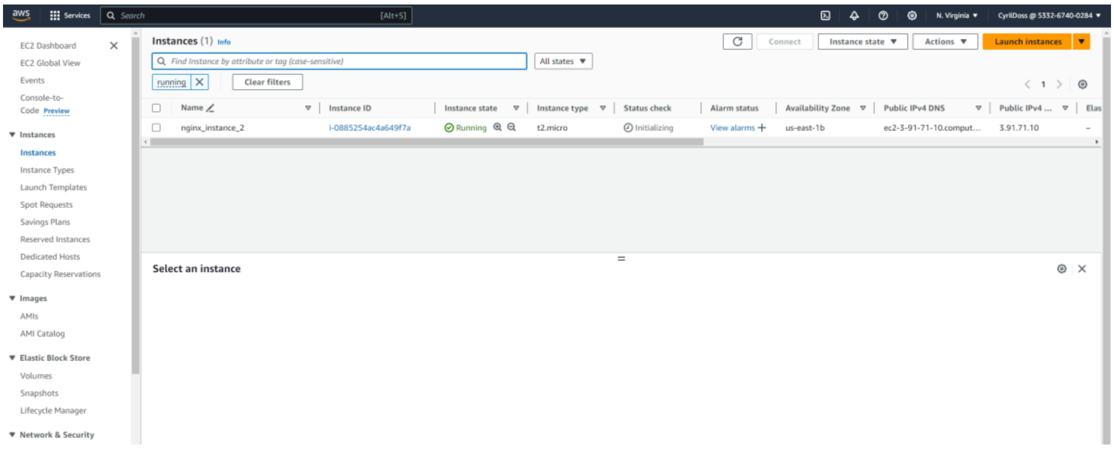
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Instance 2:

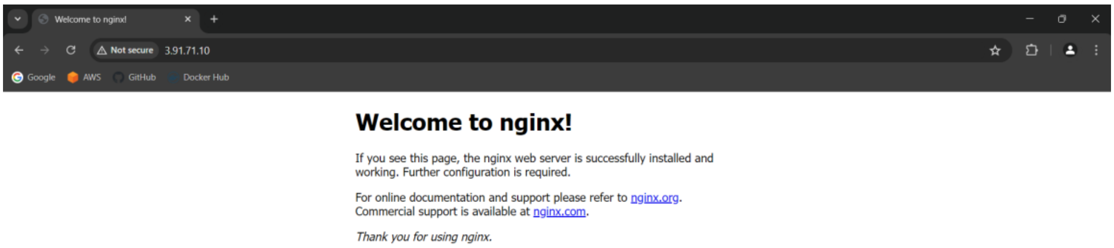


Instances (1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elas
nginx_instance_2	i-0885254ac4a649f7a	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-3-91-71-10.comput...	3.91.71.10	–

Select an instance

Accessing Nginx in instance 2:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.