Terraform Task-2

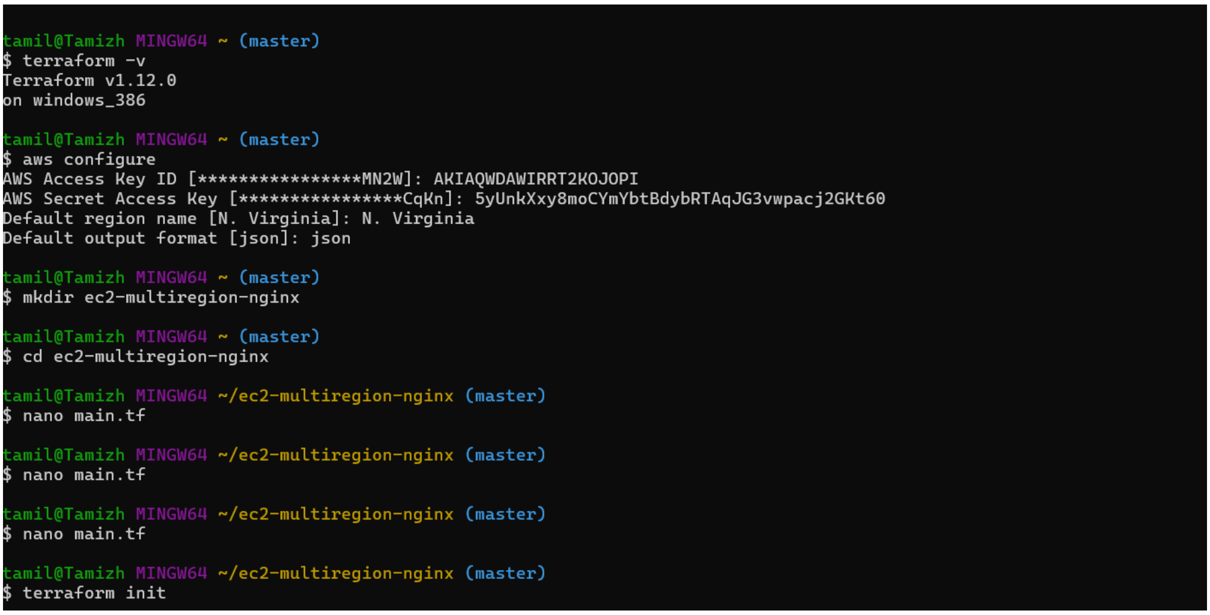
Task Description:

Create 2 EC2 instances on 2 different regions and install nginx using terraform script.

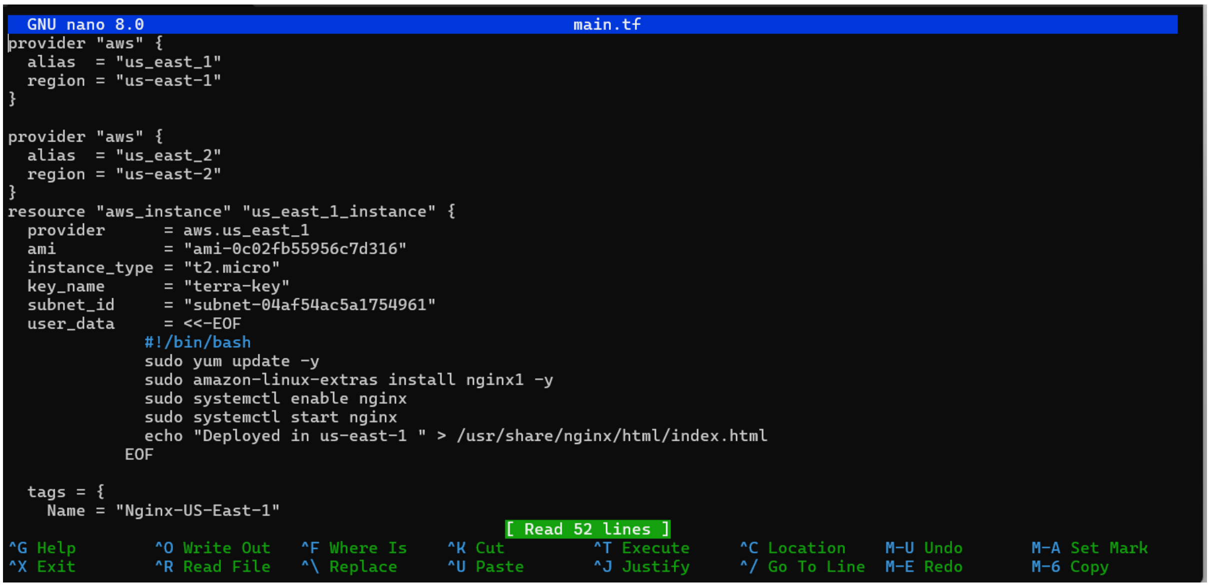
Tech stacks need to be used:

- AWS EC2
- Terraform
- AWS CLI

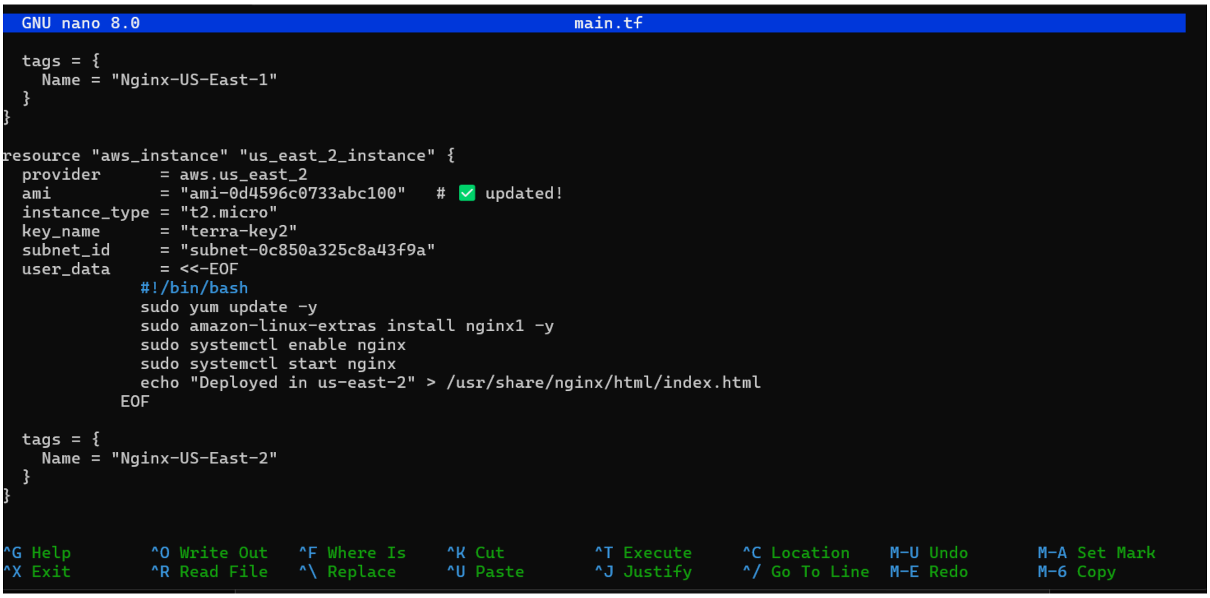
SCREENSHORTS:



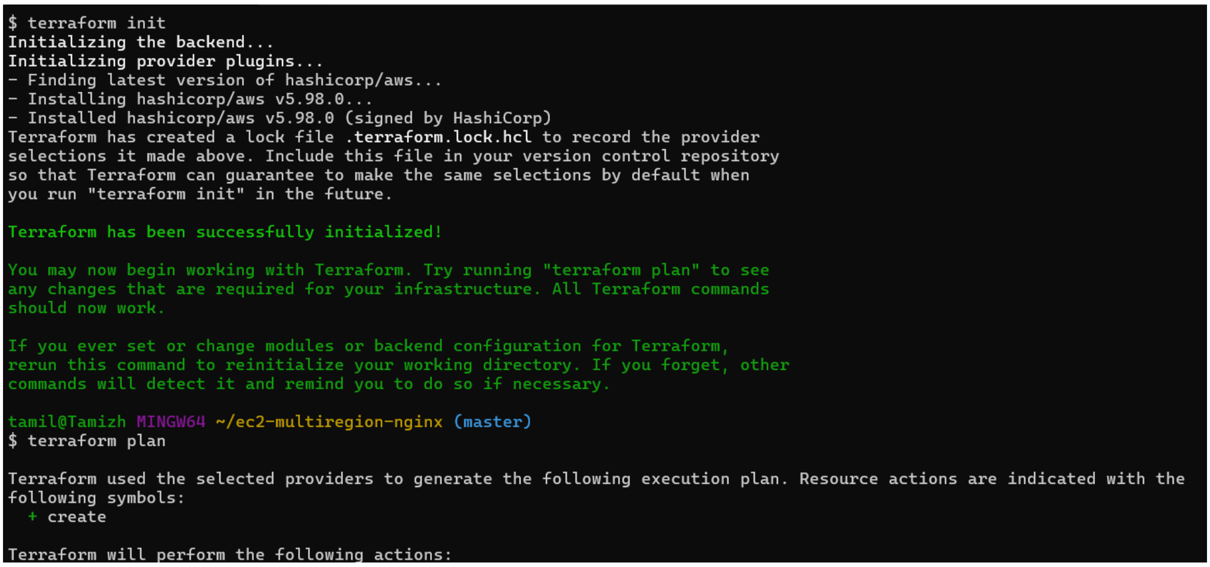
```
tamil@Tamizh MINGW64 ~ (master)
$ terraform -v
Terraform v1.12.0
on windows_386

tamil@Tamizh MINGW64 ~ (master)
$ aws configure
AWS Access Key ID [****************MN2W]: AKIAQWDAWIRRT2KOJOPI
AWS Secret Access Key [****************CqKn]: 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
Default region name [N. Virginia]: N. Virginia
Default output format [json]: json

tamil@Tamizh MINGW64 ~ (master)
$ mkdir ec2-multiregion-nginx

tamil@Tamizh MINGW64 ~ (master)
$ cd ec2-multiregion-nginx

tamil@Tamizh MINGW64 ~/ec2-multiregion-nginx (master)
$ nano main.tf

tamil@Tamizh MINGW64 ~/ec2-multiregion-nginx (master)
$ nano main.tf

tamil@Tamizh MINGW64 ~/ec2-multiregion-nginx (master)
$ nano main.tf

tamil@Tamizh MINGW64 ~/ec2-multiregion-nginx (master)
$ terraform init
```



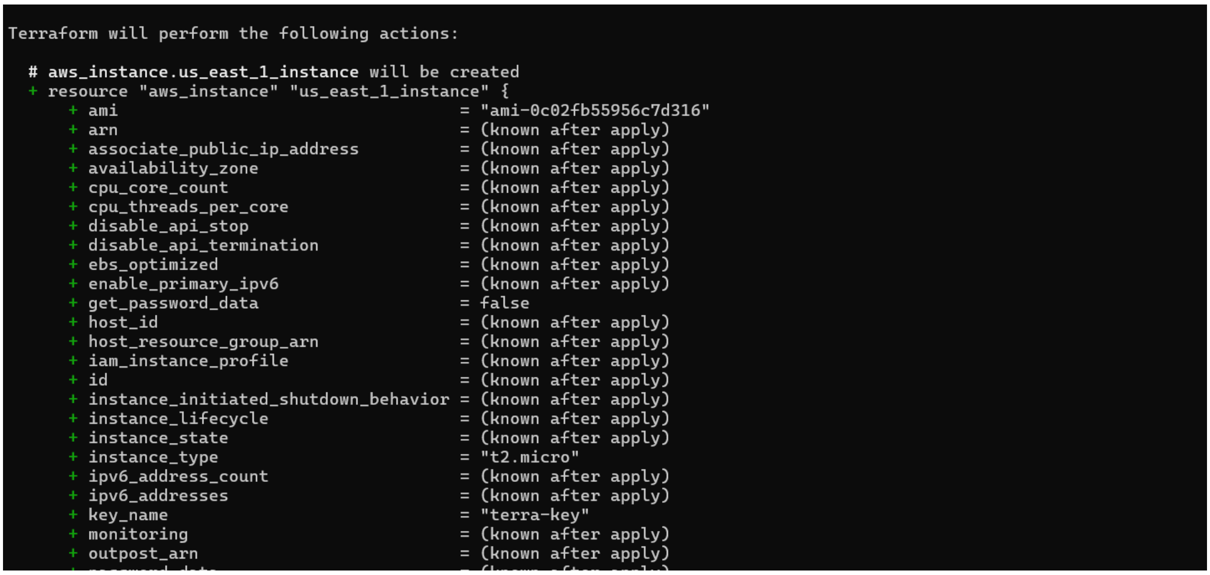
```
  GNU nano 8.0                                    main.tf
provider "aws" {
  alias  = "us_east_1"
  region = "us-east-1"
}

provider "aws" {
  alias  = "us_east_2"
  region = "us-east-2"
}
resource "aws_instance" "us_east_1_instance" {
  provider      = aws.us_east_1
  ami           = "ami-0c02fb55956c7d316"
  instance_type = "t2.micro"
  key_name      = "terra-key"
  subnet_id     = "subnet-04af54ac5a1754961"
  user_data     = <<-EOF
              #!/bin/bash
              sudo yum update -y
              sudo amazon-linux-extras install nginx1 -y
              sudo systemctl enable nginx
              sudo systemctl start nginx
              echo "Deployed in us-east-1 " > /usr/share/nginx/html/index.html
            EOF

  tags = {
    Name = "Nginx-US-East-1"
                                                     [ Read 52 lines ]
^G Help       ^O Write Out   ^F Where Is   ^K Cut        ^T Execute    ^C Location    M-U Undo      M-A Set Mark
^X Exit       ^R Read File   ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line  M-E Redo      M-6 Copy
```



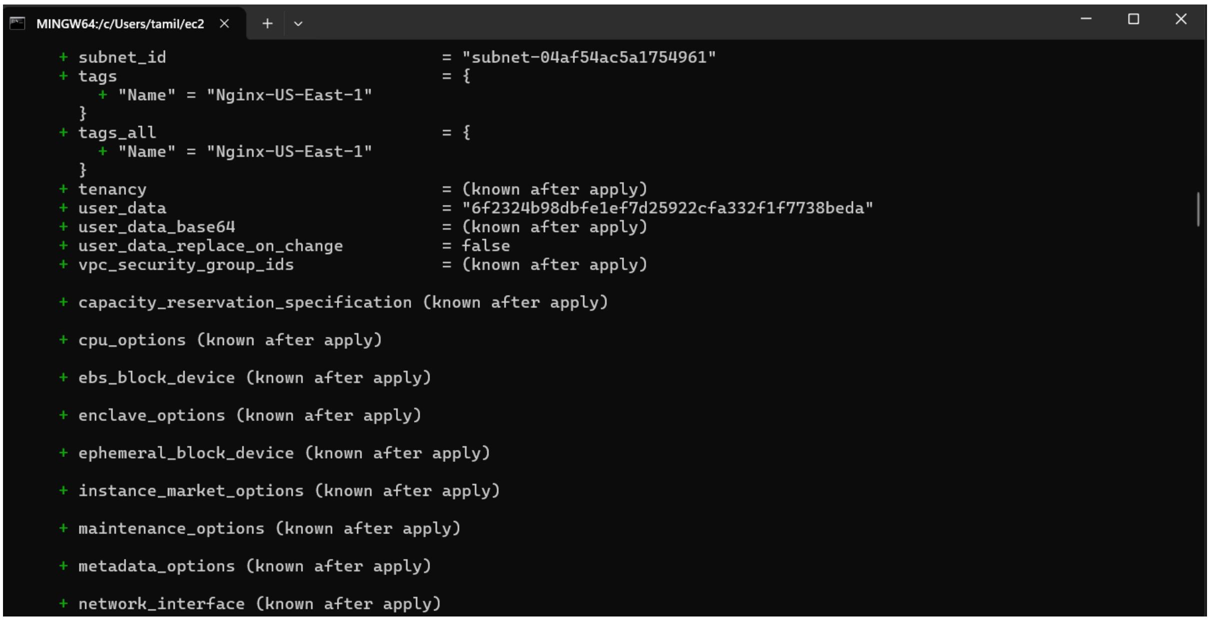
```
  GNU nano 8.0                                    main.tf

  tags = {
    Name = "Nginx-US-East-1"
  }
}

resource "aws_instance" "us_east_2_instance" {
  provider      = aws.us_east_2
  ami           = "ami-0d4596c0733abc100"   # ✅ updated!
  instance_type = "t2.micro"
  key_name      = "terra-key2"
  subnet_id     = "subnet-0c850a325c8a43f9a"
  user_data     = <<-EOF
              #!/bin/bash
              sudo yum update -y
              sudo amazon-linux-extras install nginx1 -y
              sudo systemctl enable nginx
              sudo systemctl start nginx
              echo "Deployed in us-east-2" > /usr/share/nginx/html/index.html
            EOF

  tags = {
    Name = "Nginx-US-East-2"
  }
}

^G Help       ^O Write Out   ^F Where Is   ^K Cut        ^T Execute    ^C Location    M-U Undo      M-A Set Mark
^X Exit       ^R Read File   ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line  M-E Redo      M-6 Copy
```



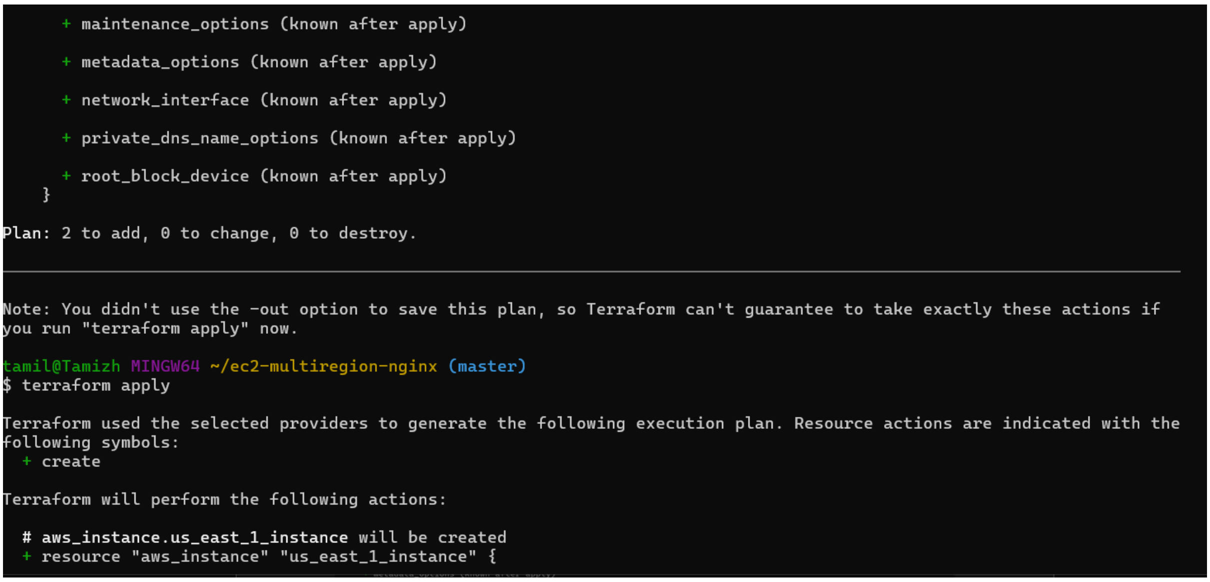
```
$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.98.0...
- Installed hashicorp/aws v5.98.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections 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.

tamil@Tamizh MINGW64 ~/ec2-multiregion-nginx (master)
$ terraform plan

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

Terraform will perform the following actions:
```



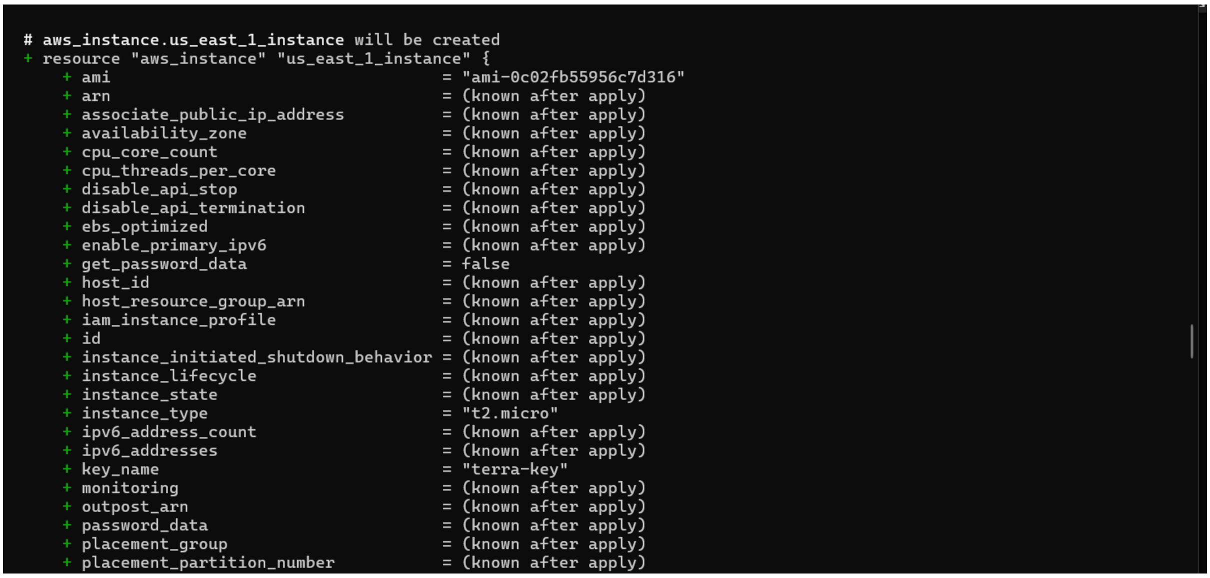
```
Terraform will perform the following actions:

  # aws_instance.us_east_1_instance will be created
  + resource "aws_instance" "us_east_1_instance" {
      + ami                                  = "ami-0c02fb55956c7d316"
      + 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)
      + enable_primary_ipv6                  = (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                             = "terra-key"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
```



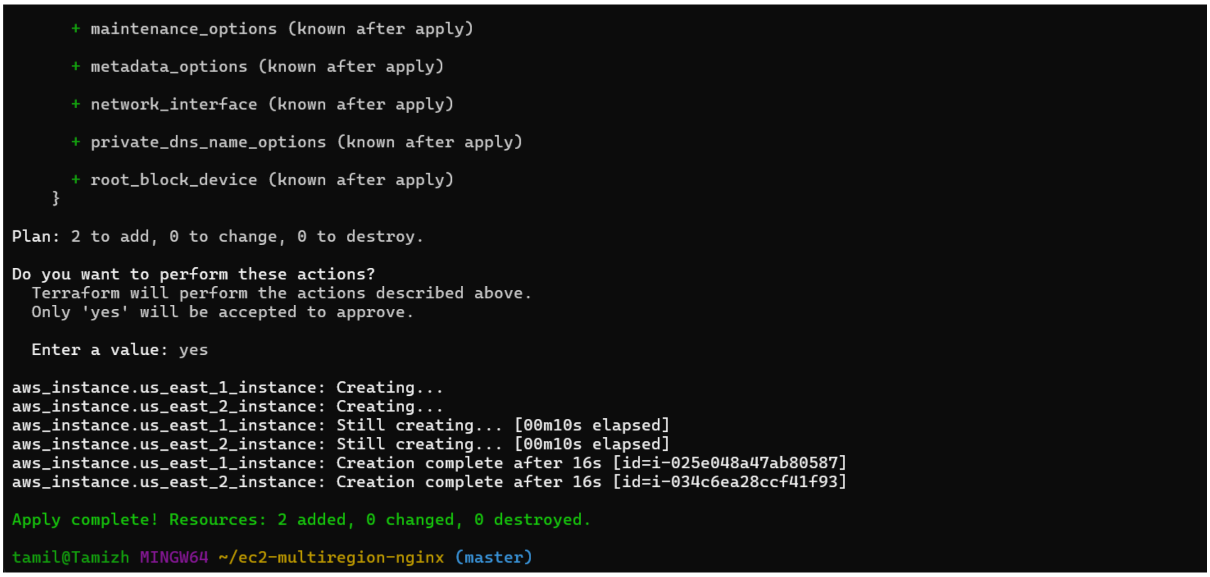
```
      + subnet_id                            = "subnet-04af54ac5a1754961"
      + tags                                 = {
          + "Name" = "Nginx-US-East-1"
        }
      + tags_all                             = {
          + "Name" = "Nginx-US-East-1"
        }
      + tenancy                              = (known after apply)
      + user_data                            = "6f2324b98dbfe1ef7d25922cfa332f1f7738beda"
      + 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)
```



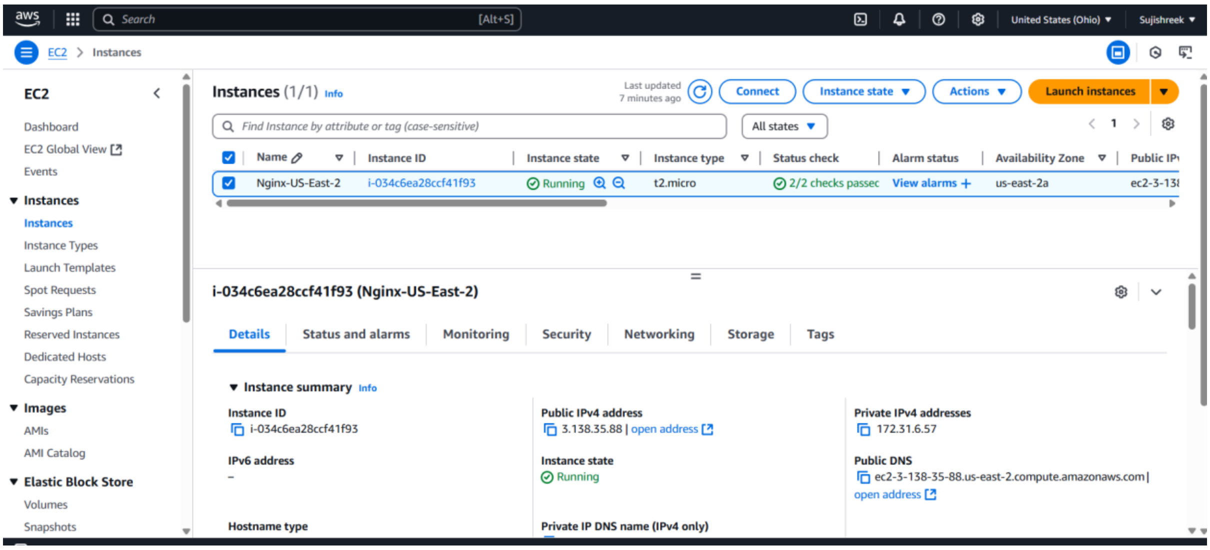
```
      + maintenance_options (known after apply)

      + 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.

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

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.

tamil@Tamizh MINGW64 ~/ec2-multiregion-nginx (master)
$ 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.us_east_1_instance will be created
  + resource "aws_instance" "us_east_1_instance" {
```



```
  # aws_instance.us_east_1_instance will be created
  + resource "aws_instance" "us_east_1_instance" {
      + ami                                  = "ami-0c02fb55956c7d316"
      + 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)
      + enable_primary_ipv6                  = (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                             = "terra-key"
      + 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)
```



```
      + 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.

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.us_east_1_instance: Creating...
aws_instance.us_east_2_instance: Creating...
aws_instance.us_east_1_instance: Still creating... [00m10s elapsed]
aws_instance.us_east_2_instance: Still creating... [00m10s elapsed]
aws_instance.us_east_1_instance: Creation complete after 16s [id=i-025e048a47ab80587]
aws_instance.us_east_2_instance: Creation complete after 16s [id=i-034c6ea28ccf41f93]

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

tamil@Tamizh MINGW64 ~/ec2-multiregion-nginx (master)
```



Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☑	Nginx-US-East-2	i-034c6ea28ccf41f93	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-2a	ec2-3-13

i-034c6ea28ccf41f93 (Nginx-US-East-2)

▼ Instance summary Info

Instance ID
i-034c6ea28ccf41f93

IPv6 address
–

Hostname type

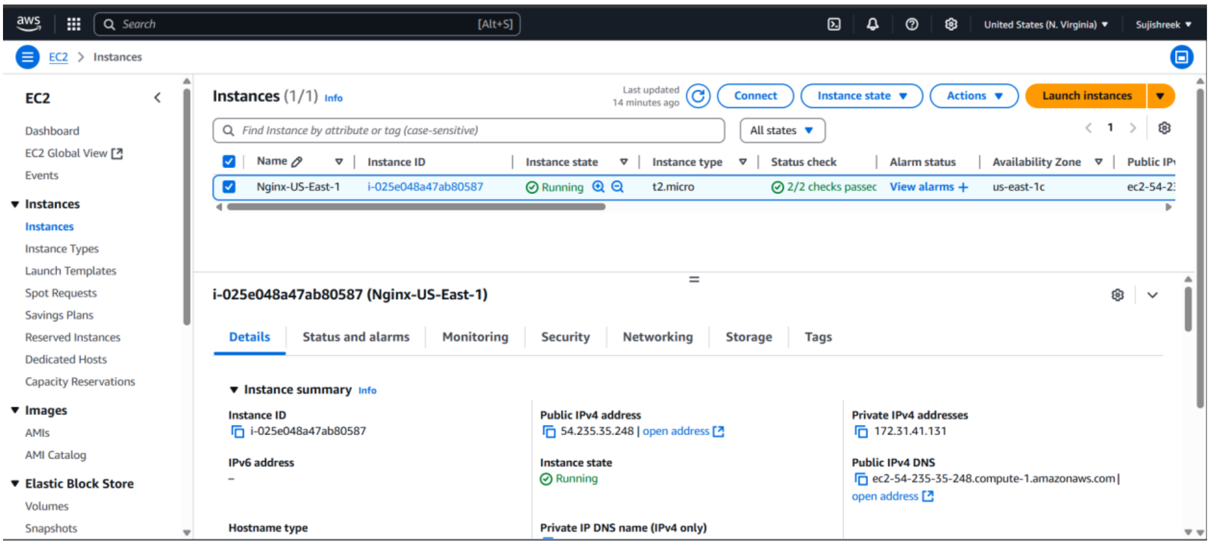
Public IPv4 address
3.138.35.88 | open address

Instance state
✓ Running

Private IP DNS name (IPv4 only)

Private IPv4 addresses
172.31.6.57

Public DNS
ec2-3-138-35-88.us-east-2.compute.amazonaws.com | open address



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☑	Nginx-US-East-1	i-025e048a47ab80587	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1c	ec2-54-2:

i-025e048a47ab80587 (Nginx-US-East-1)

▾ Instance summary Info

Instance ID
i-025e048a47ab80587

Public IPv4 address
54.235.35.248 | open address

Private IPv4 addresses
172.31.41.131

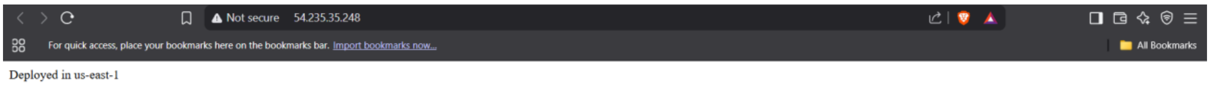
IPv6 address
–

Instance state
✓ Running

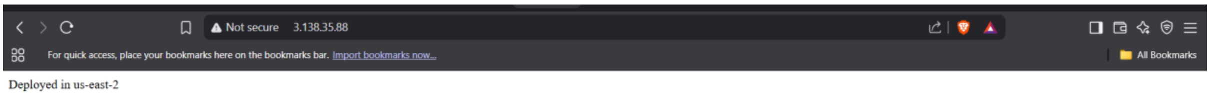
Public IPv4 DNS
ec2-54-235-35-248.compute-1.amazonaws.com | open address

Hostname type

Private IP DNS name (IPv4 only)


Not secure 54.235.35.248
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
All Bookmarks

Deployed in us-east-1


Not secure 3.138.35.88
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
All Bookmarks

Deployed in us-east-2