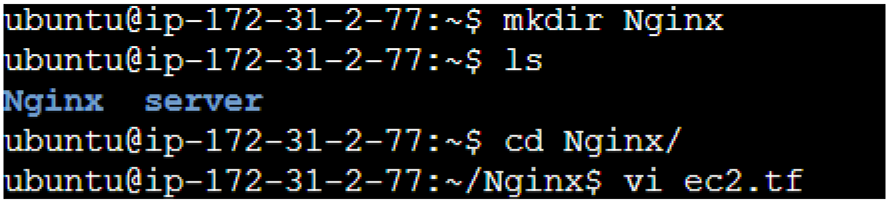
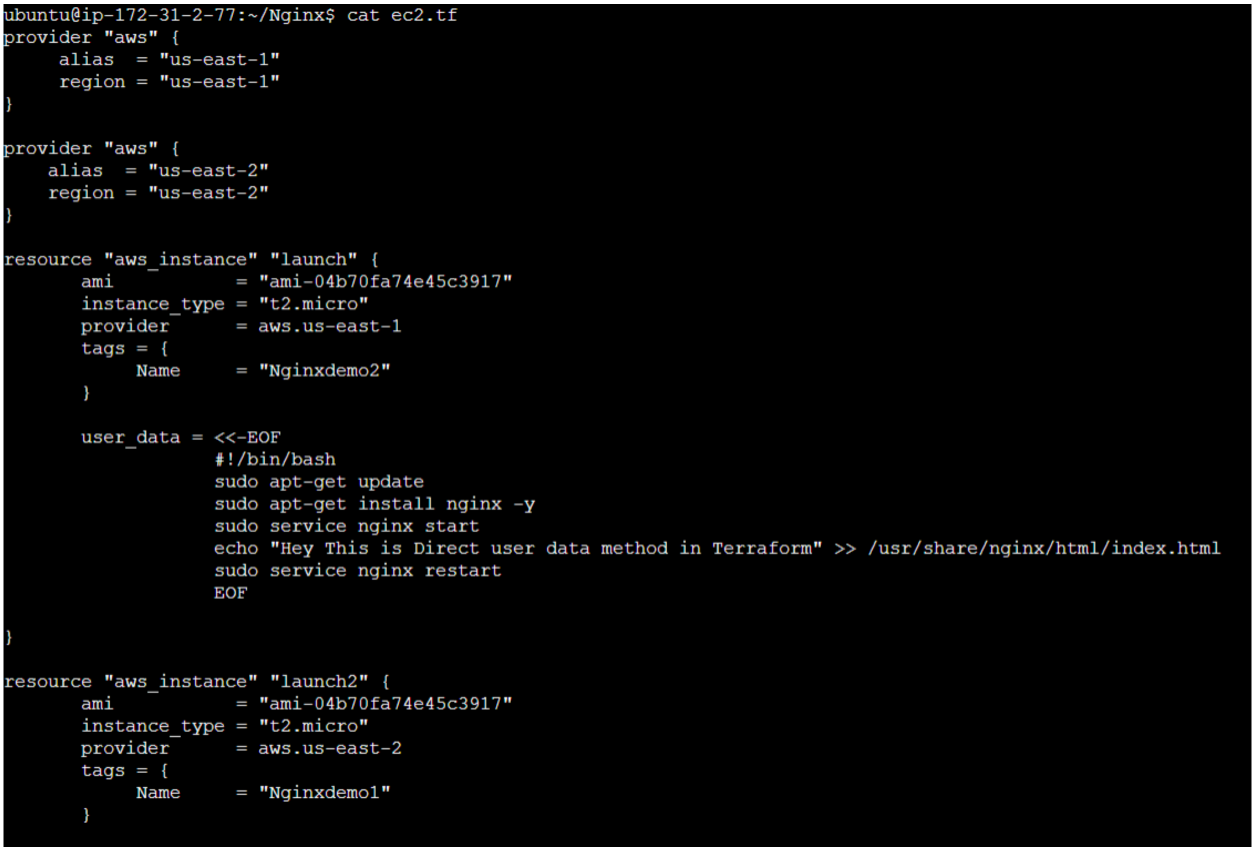
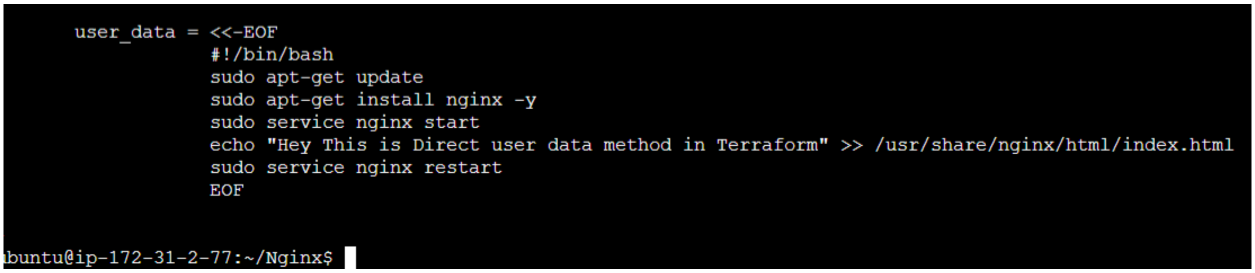
create 2 EC2 instance on 2 different regions and install nginx using terraform script



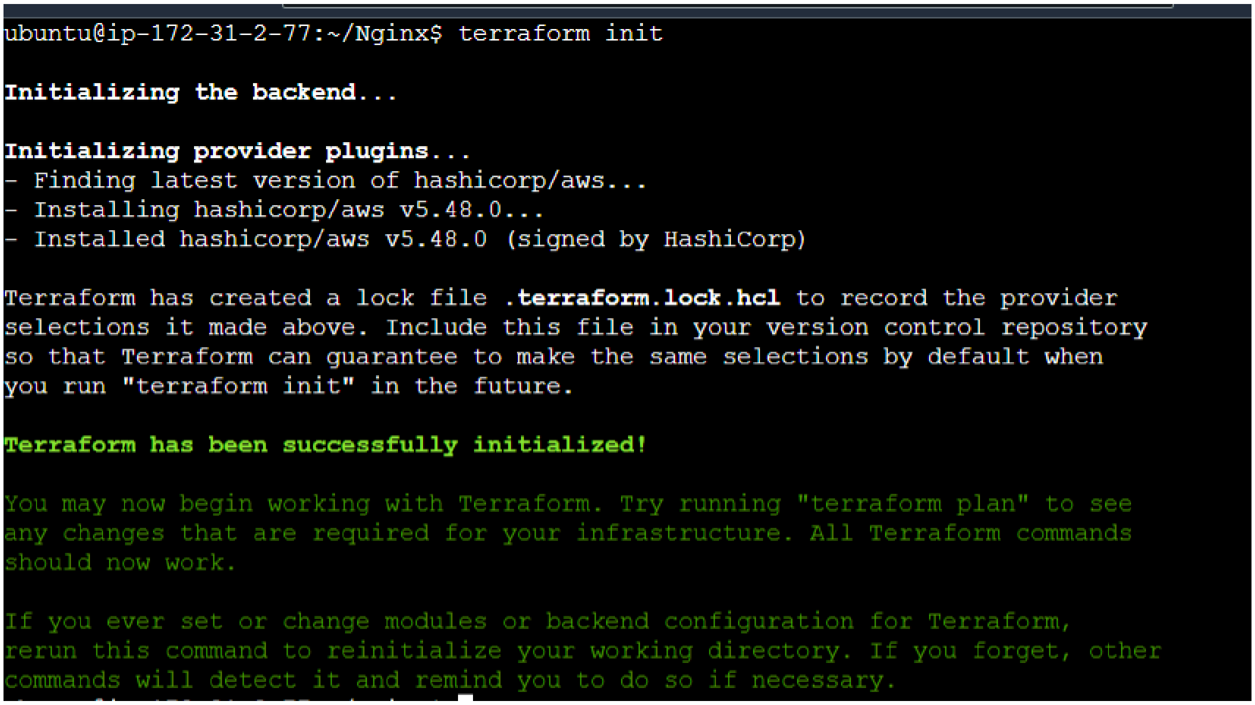
```
ubuntu@ip-172-31-2-77:~$ mkdir Nginx
ubuntu@ip-172-31-2-77:~$ ls
Nginx  server
ubuntu@ip-172-31-2-77:~$ cd Nginx/
ubuntu@ip-172-31-2-77:~/Nginx$ vi ec2.tf
```



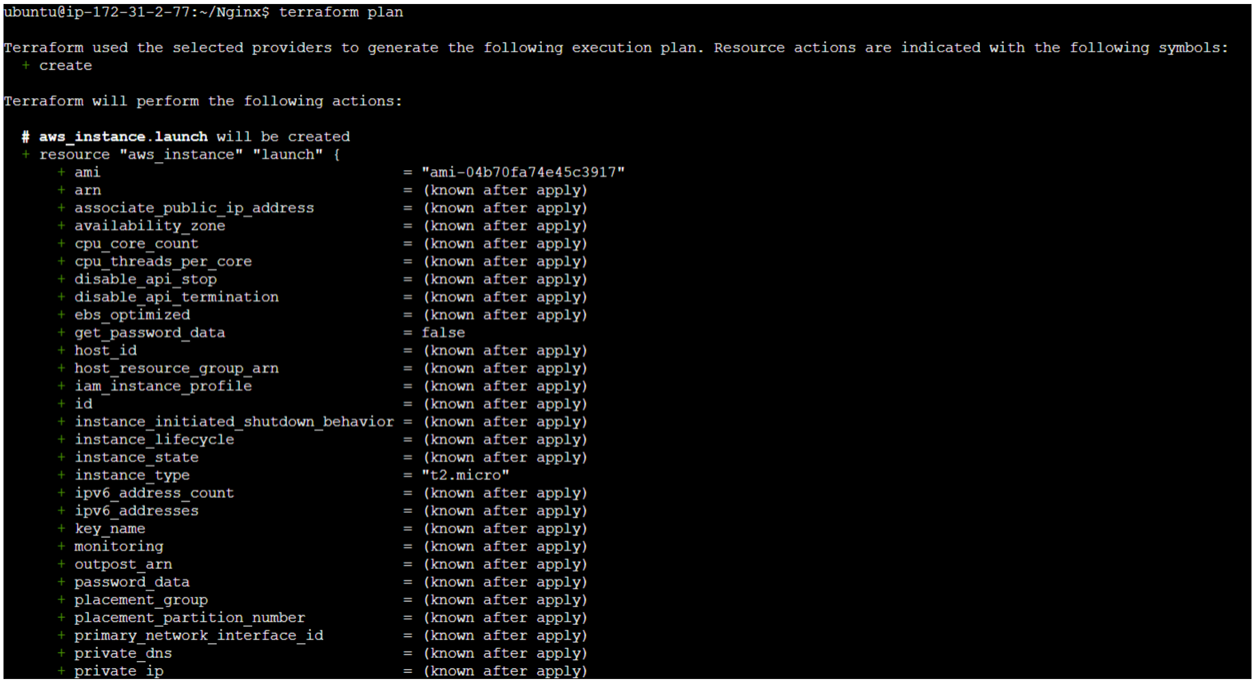
```
ubuntu@ip-172-31-2-77:~/Nginx$ cat ec2.tf
provider "aws" {
     alias  = "us-east-1"
     region = "us-east-1"
}

provider "aws" {
    alias  = "us-east-2"
    region = "us-east-2"
}

resource "aws_instance" "launch" {
       ami           = "ami-04b70fa74e45c3917"
       instance_type = "t2.micro"
       provider      = aws.us-east-1
       tags = {
            Name     = "Nginxdemo2"
       }

       user_data = <<-EOF
                   #!/bin/bash
                   sudo apt-get update
                   sudo apt-get install nginx -y
                   sudo service nginx start
                   echo "Hey This is Direct user data method in Terraform" >> /usr/share/nginx/html/index.html
                   sudo service nginx restart
                   EOF

}

resource "aws_instance" "launch2" {
       ami           = "ami-04b70fa74e45c3917"
       instance_type = "t2.micro"
       provider      = aws.us-east-2
       tags = {
            Name     = "Nginxdemo1"
       }
```



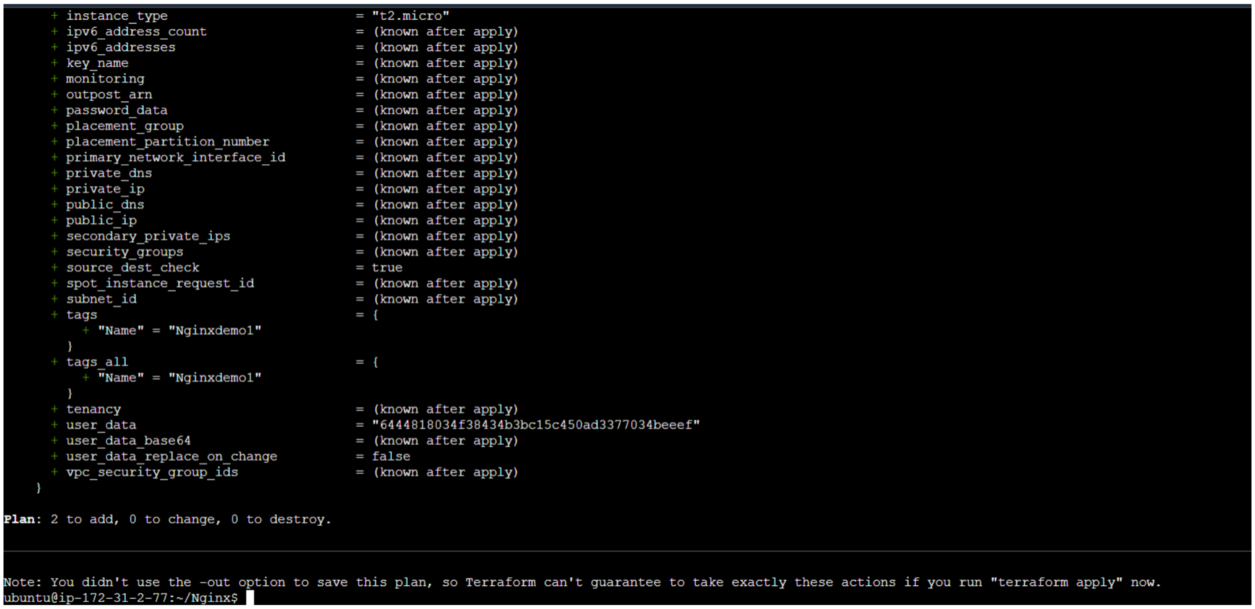
```
      user_data = <<-EOF
                  #!/bin/bash
                  sudo apt-get update
                  sudo apt-get install nginx -y
                  sudo service nginx start
                  echo "Hey This is Direct user data method in Terraform" >> /usr/share/nginx/html/index.html
                  sudo service nginx restart
                  EOF

ubuntu@ip-172-31-2-77:~/Nginx$
```



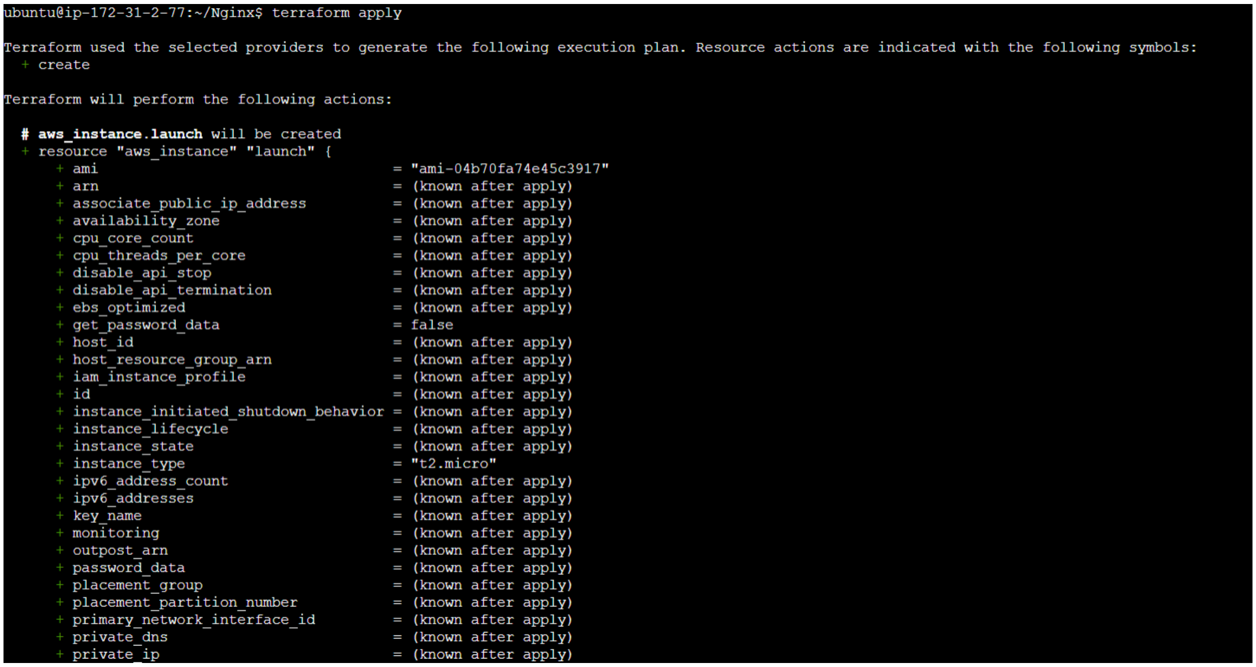
```
ubuntu@ip-172-31-2-77:~/Nginx$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.48.0...
- Installed hashicorp/aws v5.48.0 (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.
```



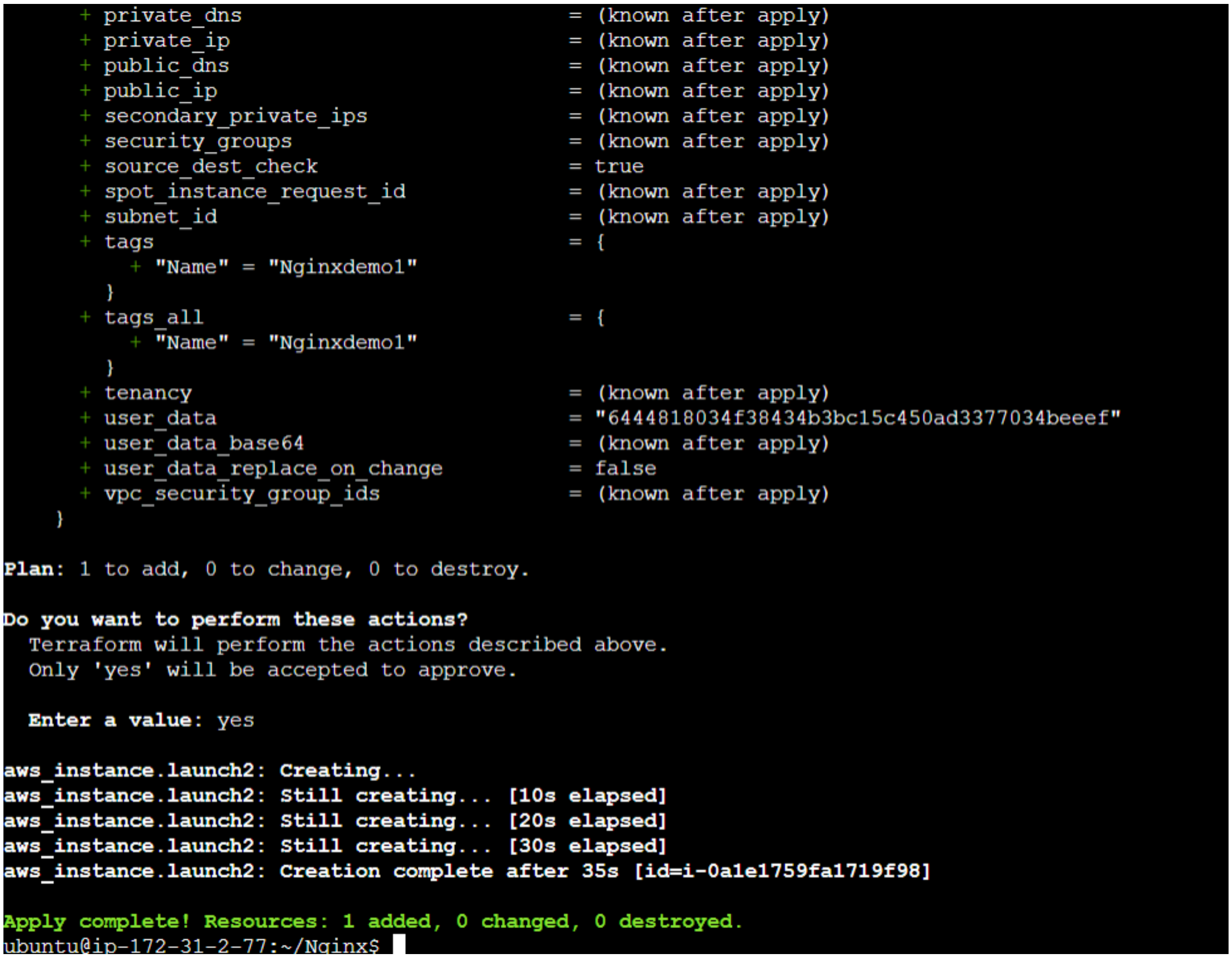
```
ubuntu@ip-172-31-2-77:~/Nginx$ terraform plan

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.launch will be created
  + resource "aws_instance" "launch" {
      + ami                                  = "ami-04b70fa74e45c3917"
      + 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                             = (known after apply)
      + 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)
```



```
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + 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" = "Nginxdemo1"
        }
      + tags_all                             = {
          + "Name" = "Nginxdemo1"
        }
      + tenancy                              = (known after apply)
      + user_data                            = "6444818034f38434b3bc15c450ad3377034beeef"
      + 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@ip-172-31-2-77:~/Nginx$
```

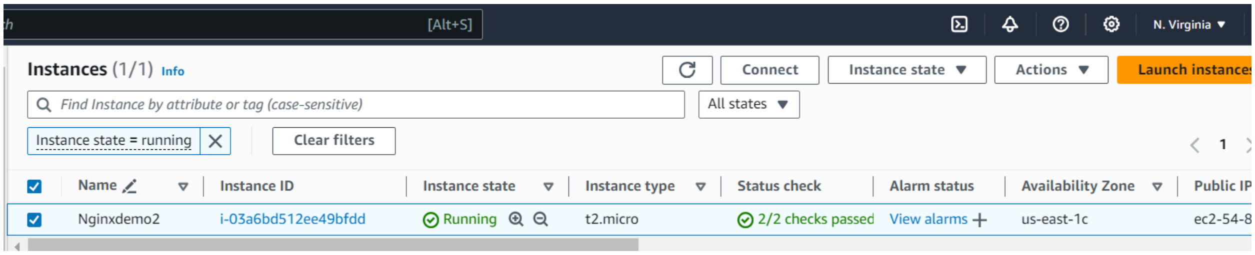


```
ubuntu@ip-172-31-2-77:~/Nginx$ terraform apply

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.launch will be created
  + resource "aws_instance" "launch" {
      + ami                                  = "ami-04b70fa74e45c3917"
      + 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                             = (known after apply)
      + 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)
```



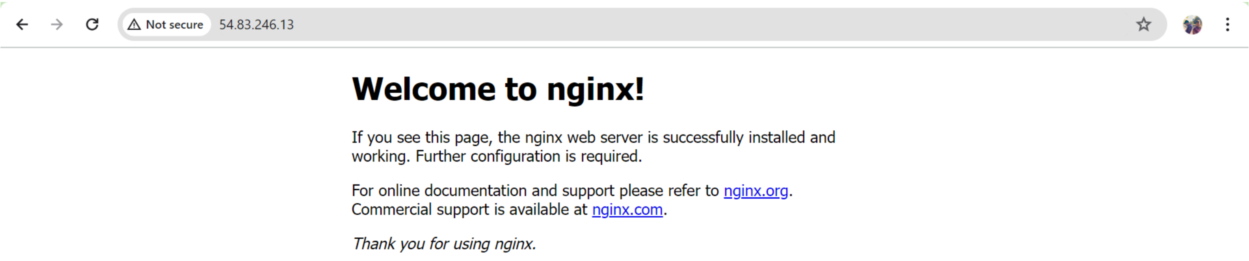
```
      + private_dns                         = (known after apply)
      + private_ip                          = (known after apply)
      + public_dns                          = (known after apply)
      + public_ip                           = (known after apply)
      + secondary_private_ips               = (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" = "Nginxdemo1"
        }
      + tags_all                            = {
          + "Name" = "Nginxdemo1"
        }
      + tenancy                             = (known after apply)
      + user_data                           = "6444818034f38434b3bc15c450ad3377034beeef"
      + 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.

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.launch2: Creating...
aws_instance.launch2: Still creating... [10s elapsed]
aws_instance.launch2: Still creating... [20s elapsed]
aws_instance.launch2: Still creating... [30s elapsed]
aws_instance.launch2: Creation complete after 35s [id=i-0a1e1759fa1719f98]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-2-77:~/Nginx$
```

us-east-1



Instance state = running

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☑	Nginxdemo2	i-03a6bd512ee49bfdd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-54-8



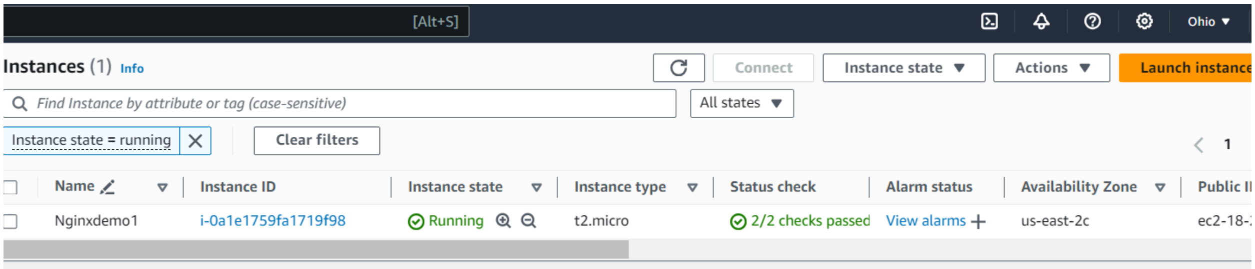
Welcome to nginx!

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.

us-east-2

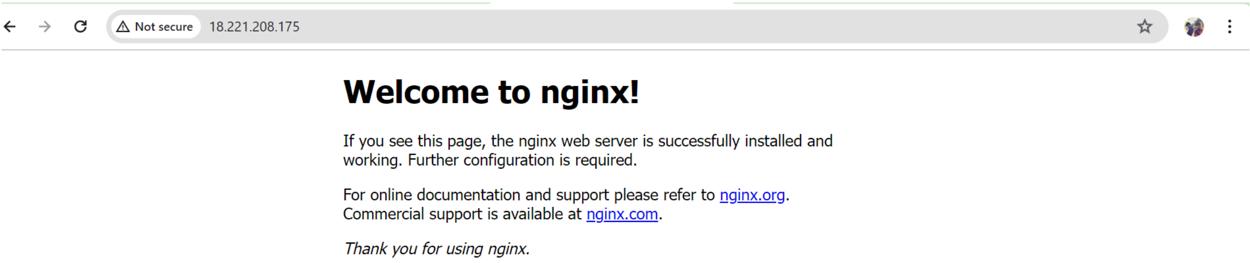


Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public I
□	Nginxdemo1	i-0a1e1759fa1719f98	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	us-east-2c	ec2-18-



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.