

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

```
yugan@Yugan:~/Tform1$ cat main.tf
#####
# Providers
#####

provider "aws" {
  alias  = "south"
  region = "ap-south-2"
}

provider "aws" {
  alias  = "east"
  region = "sa-east-1"
}

#####
# EC2 in ap-South-2
#####

resource "aws_instance" "South_instance" {
  provider      = aws.south
  ami           = "ami-02774d409be696d81"
  instance_type = "t3.micro"
  user_data     = <<-EOF
                  #!/bin/bash
                  sudo apt update -y
                  sudo apt install nginx -y
                  sudo systemctl start nginx
                  sudo systemctl enable nginx
                  EOF

  tags = {
    Name = "South-Nginx-ec2"
  }
}
```

```
#####
# EC2 in sa-east-1
#####

resource "aws_instance" "East_instance" {
  provider      = aws.east
  ami           = "ami-0a14809f48c07e3b7"
  instance_type = "t3.micro"
  user_data     = <<-EOF
    #!/bin/bash
    sudo apt update -y
    sudo apt install nginx -y
    sudo systemctl start nginx
    sudo systemctl enable nginx
  EOF

  tags = {
    Name = "East-Nginx-ec2"
  }
}
yugan@Yugan:~/Tform1$
```

```
yugan@Yugan:~/Tform1$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.100.0

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.
```

```
yugan@Yugan:~/Tform1$ terraform apply
aws_instance.South_instance: Refreshing state... [id=i-0853caa4ead92d7a8]
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.East_instance will be created
+ resource "aws_instance" "East_instance" {
  + ami           = "ami-0a14809f48c07e3b7"
  + 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)
```

Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone	Public IP	Public IPv4
East-Nginx...	i-07183fbf12f...	Running	t3.micro	3/3 checks p	View alarms +	sa-east-1a	ec2-18-22...	18.228.221...

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.

Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone	Public IP	Public IPv4
1x-ec2	i-0a6b52d0...	Running	t3.micro	3/3 checks p	View alarms +	ap-south-2a	ec2-16-11...	16.112.121.251

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.