

Created main.tf file for 2 regions South-2 and West-2 with user data file:

```
tamil@Aravind:~/Terraform$ cat main.tf
#####
# Providers
#####

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

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

#####
# 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 us-west-2
#####

resource "aws_instance" "west_instance" {
  provider = aws.west
  ami      = "ami-0786adace1541ca80"
```

```
#####
# EC2 in us-west-2
#####

resource "aws_instance" "west_instance" {
  provider      = aws.west
  ami           = "ami-0786adace1541ca80"
  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 = "west-Nginx-ec2"
  }
}

tamil@Aravind:~/Terraform$
```

Initializing Terraform:

```
tamil@Aravind:~/Terraform$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.31.0

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.
tamil@Aravind:~/Terraform$
```

```
tamil@Aravind:~/Terraform$ 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.South_instance will be created
+ resource "aws_instance" "South_instance" {
  + ami              = "ami-027774d409be696d81"
  + arn              = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (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)
  + force_destroy     = false
  + 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       = "t3.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_group_id   = (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)
  + region               = "ap-south-2"
  + 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)
```

```

+ spot_instance_request_id      = (known after apply)
+ subnet_id                    = (known after apply)
+ tags                          = {
+   + "Name" = "South-Nginx-ec2"
+ }
+ tags_all                      = {
+   + "Name" = "South-Nginx-ec2"
+ }
+ tenancy                      = (known after apply)
+ user_data                    = <<-EOT
    #!/bin/bash
    sudo apt update -y
    sudo apt install nginx -y
    sudo systemctl start nginx
    sudo systemctl enable nginx
EOT
+ 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)

+ primary_network_interface (known after apply)

+ private_dns_name_options (known after apply)

+ root_block_device (known after apply)
}

# aws_instance.west_instance will be created
+ resource "aws_instance" "west_instance" {

```

```
# aws_instance.west_instance will be created
+ resource "aws_instance" "west_instance" {
+   ami                        = "ami-0786adace1541ca80"
+   arn                       = (known after apply)
+   associate_public_ip_address = (known after apply)
+   availability_zone         = (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)
+   force_destroy              = false
+   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              = "t3.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_group_id          = (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)
+   region                      = "us-west-2"
+   secondary_private_ips        = (known after apply)
+   security_groups              = (known after apply)
+   source_dest_check           = true
}
```

```

}
+ tags_all                      = {
+   "Name" = "west-Nginx-ec2"
+ }
+ tenancy                      = (known after apply)
+ user_data                    = <<-EOT
    #!/bin/bash
    sudo apt update -y
    sudo apt install nginx -y
    sudo systemctl start nginx
    sudo systemctl enable nginx
EOT
+ 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)

+ primary_network_interface (known after apply)

+ private_dns_name_options (known after apply)

+ root_block_device (known after apply)
}
lan: 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.
amil@Aravind:~/Terraform\$

Ec2 launched in south-2 region:

[Alt+S] [Icons] Asia Pacific (Hyderabad) Senthamil2691 (3192-2014-3636) Senthamil2691

Instances (1) [Info](#) Last updated less than a minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) Running

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	South-Nginx-ec2	i-0767954c3a62e2398	Running	t3.micro	3/3 checks passed	View alarms +	ap-south-2a	ec2-98-130-121-32

[Alt+S] [Icons] Asia Pacific (Hyderabad) Senthamil2691 (3192-2014-3636) Senthamil2691

i-0767954c3a62e2398

Instance summary for i-0767954c3a62e2398 (South-Nginx-ec2) [Info](#) [Connect](#) [Instance state](#) [Actions](#)

Updated less than a minute ago

Instance ID i-0767954c3a62e2398	Public IPv4 address 98.130.121.32 open address	Private IPv4 addresses 172.31.7.29
IPv6 address -	Instance state Running	Public DNS ec2-98-130-121-32.ap-south-2.compute.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	

Validated application is getting hosted in web:

← ↻ ⚠ Not secure 98.130.121.32 [Icons]

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.

Ec2 launched in west-2 region:

[Alt+S] [Icons] Ask Amazon Q United States (Oregon) Senthamil2691 (3192-2014-3636) Senthamil2691

Instances (1) [Info](#) Last updated less than a minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) Running


<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	west-Nginx-ec2	i-0c8da7a3e42ec277e	Running	t3.micro	3/3 checks passed	View alarms +	us-west-2b	ec2-34-21-

i-8da7a3e42ec277e

Instance summary for i-0c8da7a3e42ec277e (west-Nginx-ec2) Info

Updated less than a minute ago


Instance ID

 i-0c8da7a3e42ec277e


IPv6 address

—


Public IPv4 address

 34.212.6.169 | [open address](#)


Instance state


 Running


Private IPv4 addresses


 172.31.26.5

Public DNS




 ec2-34-212-6-169.us-west-2.compute.amazonaws.com | [open address](#)

 Connect

 Instance state

 Actions

Validated application is getting hosted in web:

   Not secure 34.212.6.169

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.