```
provider "aws" {
    alias  = "region1"
    region = "us-east-1"
}

provider "aws" {
    alias  = "region2"
    region = "us-west-2"
}

resource "aws_instance" "instance_east_1" {
    provider = aws.region1
    ami         = "ami-0427090fd1714168b"
    instance_type = "t2.micro"

    tags = {
        Name = "instance-us-east-1"
    }

    user_data = <<-EOF
"main.tf" 45L, 911B
```

```
    tags = {
        Name = "instance-us-east-1"
    }

    user_data = <<-EOF
            #!/bin/bash
            sudo yum update -y
            sudo yum install -y nginx
            sudo systemctl start nginx
            sudo systemctl enable nginx
            EOF
}

resource "aws_instance" "instance_us_west_2" {
    provider = aws.region2
    ami         = "ami-074be47313f84fa38"
    instance_type = "t2.micro"

    tags = {
        Name = "instance-us-west-2"
```

```
        EOF
}

resource "aws_instance" "instance_us_west_2" {
    provider = aws.region2
    ami         = "ami-074be47313f84fa38"
    instance_type = "t2.micro"

    tags = {
        Name = "instance-us-west-2"
    }

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

```
[ec2-user@ip-172-31-39-48 terraform-EC2-nginx]$ vi main.tf
[ec2-user@ip-172-31-39-48 terraform-EC2-nginx]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.60.0

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.
[ec2-user@ip-172-31-39-48 terraform-EC2-nginx]$ terraform validate
Success! The configuration is valid.

[ec2-user@ip-172-31-39-48 terraform-EC2-nginx]$
```

```
        + private_dns_name_options (known after apply)

        + root_block_device (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.instance_east_1: Creating...
aws_instance.instance_east_1: Still creating... [10s elapsed]
aws_instance.instance_east_1: Still creating... [20s elapsed]
aws_instance.instance_east_1: Still creating... [30s elapsed]
aws_instance.instance_east_1: Creation complete after 35s [id=i-03537c254a6227fef]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-39-48 terraform-EC2-nginx]$
```

Instances (1) Info   ↻   Connect   Instance state ▾   Actions ▾   **Launch instances** ▾   ⊘

EC2 Dashboard   ✕

EC2 Global View

Events   Q Find Instance by attribute or tag (case-sensitive)   All states ▾   ‹ 1 › ⚙

Console-to-Code Preview

☐ Name ✎   ▽   Instance ID   Instance state ▾   Instance type ▾   Status check

▼ Instances

☐ instance-us-east-1   i-03537c254a6227fef   ⊘ Running 🔍 🔍   t2.micro   ⏱ Initializing

---

Instances (2) Info   ↻   Connect   Instance state ▾   Actions ▾   **Launch instances** ▾   ⊘

EC2 Dashboard   ✕

EC2 Global View

Events   Q Find Instance by attribute or tag (case-sensitive)   All states ▾   ‹ 1 › ⚙

▼ Instances

  Instances   ☐ Name ✎   ▽   Instance ID   Instance state ▾   Instance type ▾   Status check   Alarm sta

  Instance Types   ☐ instance-us-w...   i-0af51df5ca1ddb6d3   ⊖ Terminated 🔍 🔍   t2.micro   –   View alarm

                   ☐ instance-us-w...   i-0442bb14f874529a1   ⊘ Running 🔍 🔍   t2.micro   ⊘ 2/2 checks passec View alarm

---

```
[ec2-user@ip-172-31-38-122 ~]$ ls
[ec2-user@ip-172-31-38-122 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-07-28 10:55:28 UTC; 22min ago
   Main PID: 3580 (nginx)
      Tasks: 2 (limit: 1114)
     Memory: 2.2M
        CPU: 57ms
     CGroup: /system.slice/nginx.service
             ├─3580 "nginx: master process /usr/sbin/nginx"
             └─3599 "nginx: worker process"

Jul 28 10:55:28 ip-172-31-38-122.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Jul 28 10:55:28 ip-172-31-38-122.ec2.internal nginx[3472]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jul 28 10:55:28 ip-172-31-38-122.ec2.internal nginx[3472]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jul 28 10:55:28 ip-172-31-38-122.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-38-122 ~]$ ls
```

```
[ec2-user@ip-172-31-22-197 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Sun 2024-07-28 11:40:36 UTC; 29s ago
   Main PID: 26482 (nginx)
      Tasks: 2 (limit: 1114)
     Memory: 2.2M
        CPU: 55ms
     CGroup: /system.slice/nginx.service
             ├─26482 "nginx: master process /usr/sbin/nginx"
             └─26484 "nginx: worker process"

Jul 28 11:40:36 ip-172-31-22-197.us-west-2.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy>
Jul 28 11:40:36 ip-172-31-22-197.us-west-2.compute.internal nginx[26452]: nginx: the configuration file /etc/nginx/nginx.conf syn>
Jul 28 11:40:36 ip-172-31-22-197.us-west-2.compute.internal nginx[26452]: nginx: configuration file /etc/nginx/nginx.conf test is>
Jul 28 11:40:36 ip-172-31-22-197.us-west-2.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy>
lines 1-15/15 (END)
```