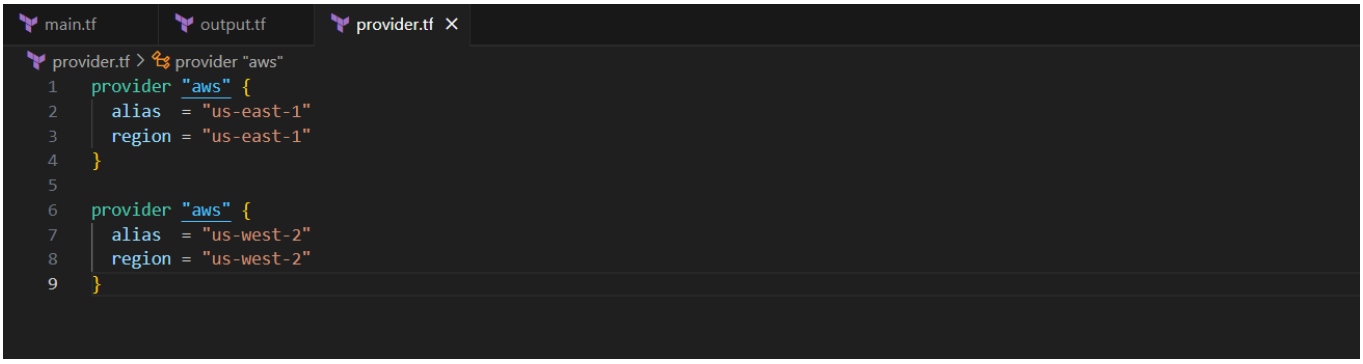
Terraform Task -2

Launch Linux EC2 instances in two regions using a single Terraform file.



creating the provider file with 2 region

Create the instance with the security group

```
}

resource "aws_instance" "us_east_instance" {
  provider          = aws.us-east-1
  ami               = "ami-0427090fd1714168b" # Replace with a valid AMI ID in us-east-1
  instance_type     = "t2.micro"
  key_name          = "temp"
  security_groups   = [aws_security_group.allow_ssh_http.name]

  tags = {
    Name = "US-East-Instance"
  }

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

resource "aws_instance" "us_west_instance" {
  provider          = aws.us-west-2
  ami               = "ami-074be47313f84fa38" # Replace with a valid AMI ID in us-west-2
  instance_type     = "t2.micro"
  key_name          = "temp1"
  security_groups   = [aws_security_group.allow_ssh_http_west.name]

  tags = {
    Name = "US-West-Instance"
  }

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

Ln 93, Col 1 (2 selected) Spaces: 4 UTF-8 CRLF {} Terraform Go Live

Create the instance with the security group

main.tf | output.tf | provider.tf

output.tf > output "us_west_instance_public_ip"

```
output "us_east_instance_public_ip" {
  value = aws_instance.us_east_instance.public_ip
}

output "us_west_instance_public_ip" {
  value = aws_instance.us_west_instance.public_ip
}
```

output the IP

```
PS D:\Terraform 1> 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.
```

CREATING THE TERRAFORM FILE.

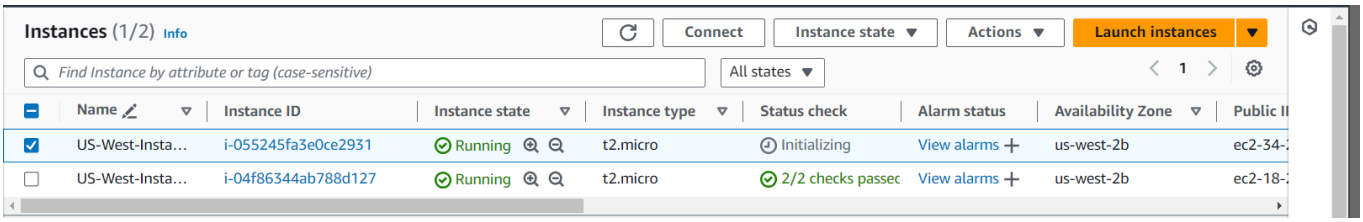
```
commands will detect it and remind you to do so if necessary.
PS D:\Terraform 1> terraform plan
aws_security_group.allow_ssh_http: Refreshing state... [id=sg-01c56c0f53fae99c6]
aws_security_group.allow_ssh_http_west: Refreshing state... [id=sg-0244e66154bdee9a6]
aws_instance.us_east_instance: Refreshing state... [id=i-0c645d063e46231f2]

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_west_instance will be created
  + resource "aws_instance" "us_west_instance" {
      + ami                                  = "ami-074be47313f84fa38"
      + 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                             = "temp1"
      + 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)
```

Ln 7, Col 2 Spaces: 4 UTF-8 CRLF {} Terraform Go Live

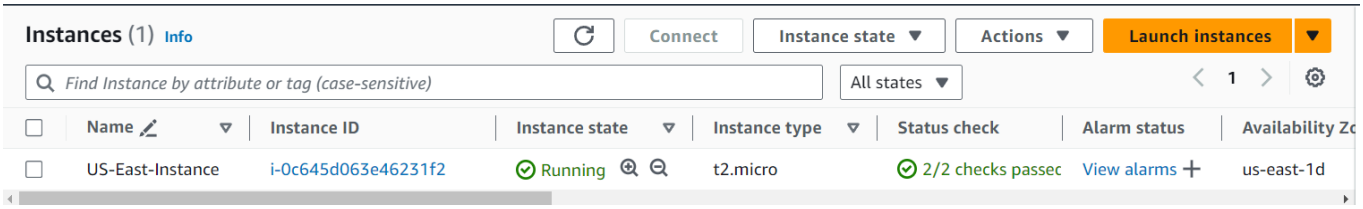
PLANNING PROCESS

```
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.
PS D:\Terraform 1> terraform apply --auto-approve
aws_security_group.allow_ssh_http: Refreshing state... [id=sg-01c56c0f53fae99c6]
aws_security_group.allow_ssh_http_west: Refreshing state... [id=sg-0244e66154bdee9a6]
aws_instance.us_east_instance: Refreshing state... [id=i-0c645d063e46231f2]

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_west_instance will be created
  + resource "aws_instance" "us_west_instance" {
      + ami                                  = "ami-074be47313f84fa38"
      + 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                             = "temp1"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
```

CREATING THE INSTANCE USING TERRAFORM



CREATED INSTANCE



ANOTHER ZONE INSTANCE