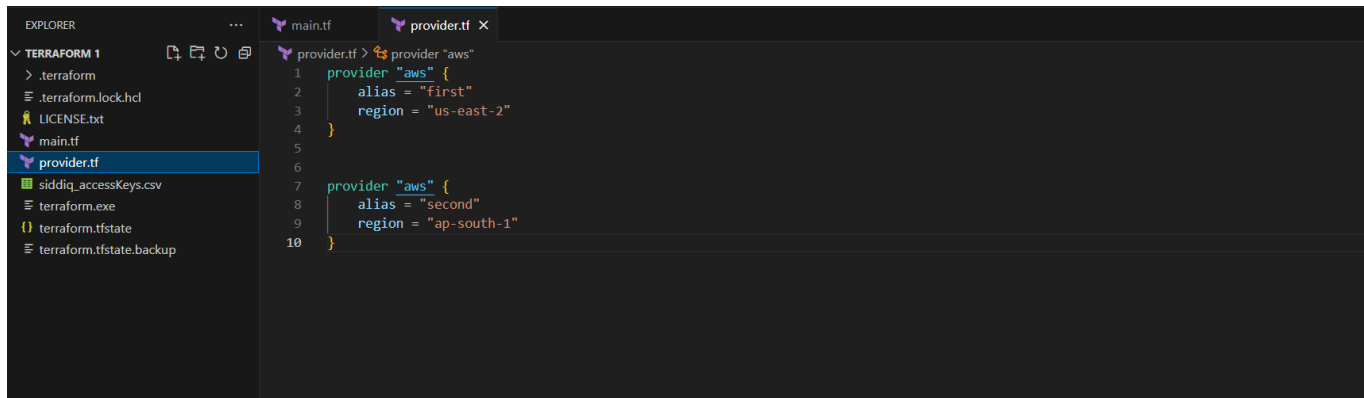


Terraform file

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

creating the terraform file :



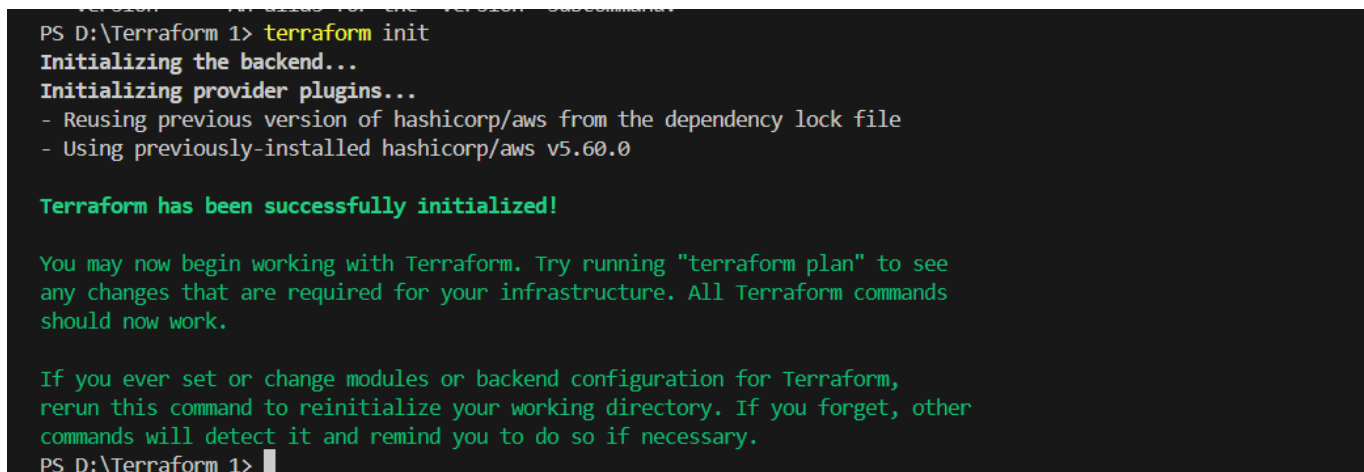
```
1 provider "aws" {
2   alias = "first"
3   region = "us-east-2"
4 }
5
6
7 provider "aws" {
8   alias = "second"
9   region = "ap-south-1"
10 }
```

create the provider file and mention the provider and 2 regions



```
1 resource "aws_instance" "p-ec2" {
2   count = 6
3   provider = aws.first
4   ami = "ami-00db8dadb36c9815e"
5   instance_type = "t2.micro"
6 }
7
8
9 resource "aws_instance" "s-ec2" {
10   provider = aws.second
11   ami = "ami-068e0f1a600cd311c"
12   instance_type = "t2.micro"
13 }
14
```

create the main file and create the resource content for two region instance



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

running the terraform init command to make the file to terraform file

```

PS D:\Terraform 1> terraform plan
aws_instance.s-ec2: Refreshing state... [id=i-03f5ec8be500b8ae6]
aws_instance.p-ec2[0]: Refreshing state... [id=i-08ccad357892ec2ef]
aws_instance.p-ec2[4]: Refreshing state... [id=i-08a9fa9c7ded1c18a]
aws_instance.p-ec2[3]: Refreshing state... [id=i-0fa25e88ccc74a19e]
aws_instance.p-ec2[1]: Refreshing state... [id=i-0c9b2bca0635b0cb9]
aws_instance.p-ec2[5]: Refreshing state... [id=i-086995bc03118c292]
aws_instance.p-ec2[2]: Refreshing state... [id=i-0bca7a0e178fe5ad4]

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.p-ec2[0] will be created
+ resource "aws_instance" "p-ec2" {
  + ami                     = "ami-00db8dad36c9815e"
  + 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)

```

run the terraform plan to create the blue print of the 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_instance.s-ec2: Refreshing state... [id=i-03f5ec8be500b8ae6]
aws_instance.p-ec2[3]: Refreshing state... [id=i-0fa25e88ccc74a19e]
aws_instance.p-ec2[1]: Refreshing state... [id=i-0c9b2bca0635b0cb9]
aws_instance.p-ec2[0]: Refreshing state... [id=i-08ccad357892ec2ef]
aws_instance.p-ec2[5]: Refreshing state... [id=i-086995bc03118c292]
aws_instance.p-ec2[2]: Refreshing state... [id=i-0bca7a0e178fe5ad4]
aws_instance.p-ec2[4]: Refreshing state... [id=i-08a9fa9c7ded1c18a]

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.p-ec2[0] will be created
+ resource "aws_instance" "p-ec2" {
  + ami                     = "ami-00db8dad36c9815e"
  + 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)

```

terraform apply command to create the 2 instance in 2 available zone

output

Instances (1) Info

Refresh

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

< 1 > ⚙

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>		i-06348ac1f4d2494ff	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a

in ap-south-1a region

Instances (1) Info

Refresh

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

Instance state = running

Clear filters

< 1 > ⚙

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>		i-0e884a67b80378f6c	Running	t2.micro	2/2 checks passed	View alarms	us-east-2a	ec2-18-115

another zone is us-east-2a