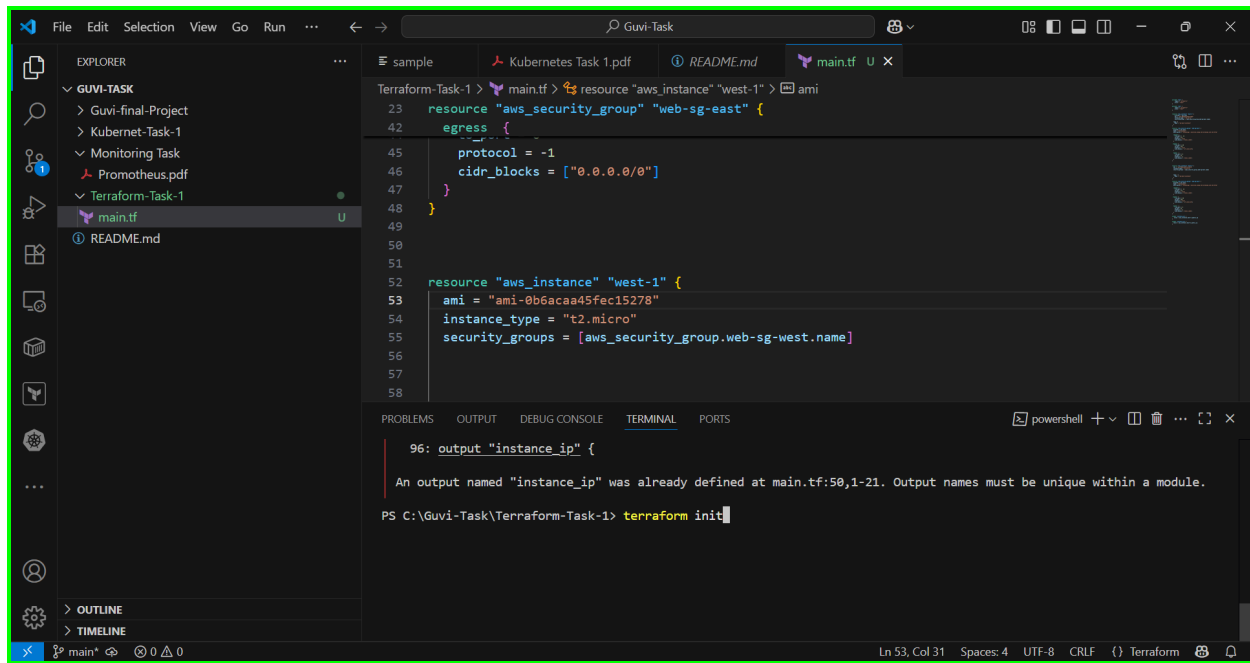


Launch Linux EC2 instances in two regions using a single Terraform file.

The script file is attached in the same git repo, so as per the requirement the script is created and

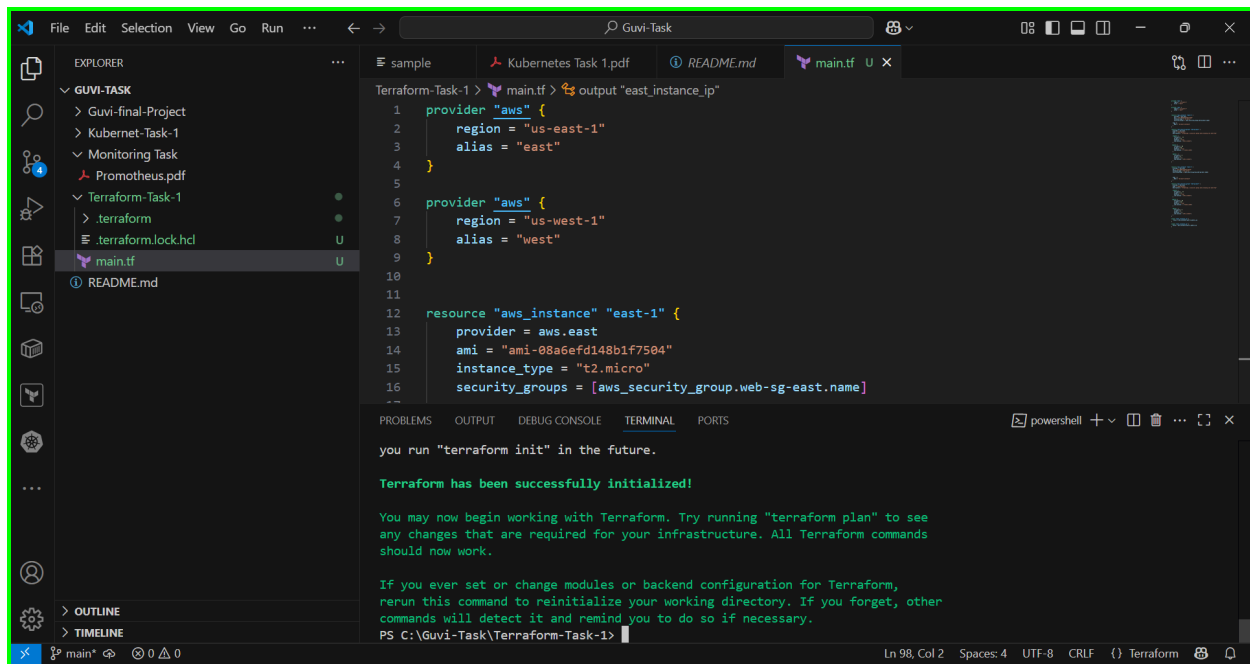
Applying terraform init



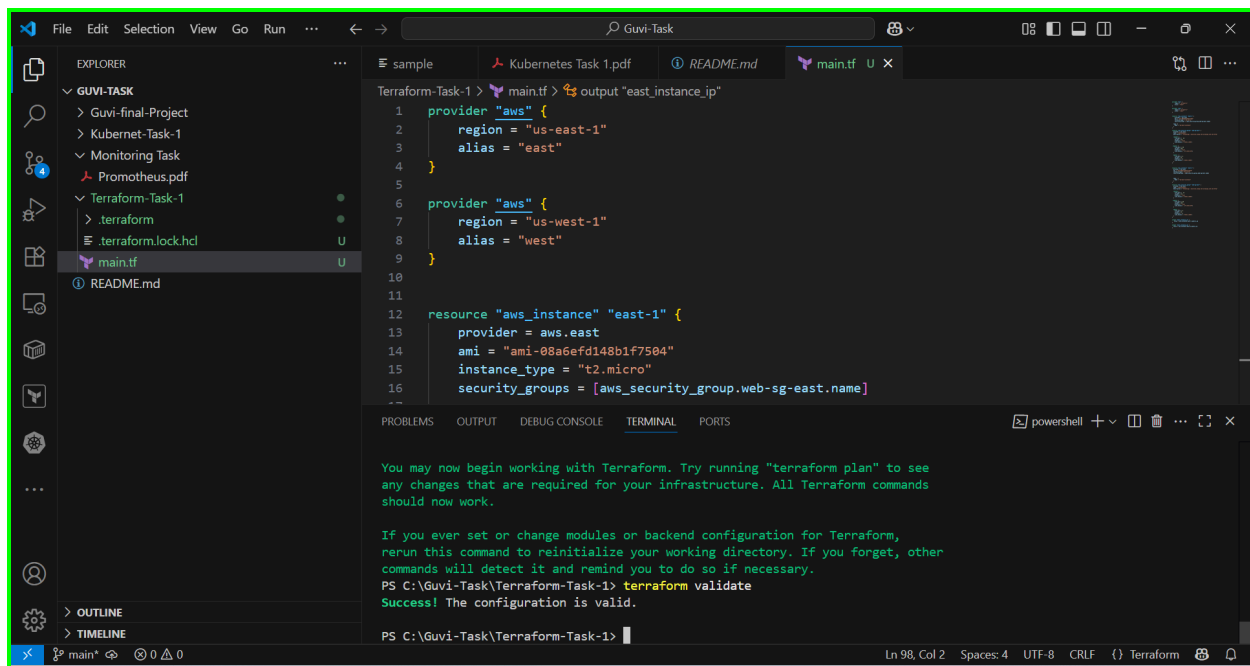
The screenshot shows the Visual Studio Code interface with a Terraform configuration file named `main.tf` open. The file contains Terraform code for creating an AWS security group and an EC2 instance. The terminal window at the bottom shows an error message from Terraform:

```
96: output "instance_ip" {  
  An output named "instance_ip" was already defined at main.tf:50,1-21. Output names must be unique within a module.  
PS C:\Guvi-Task\Terraform-Task-1> terraform init
```

The error message indicates that the output name `instance_ip` is already defined in the file at line 50, column 1. The terminal window also shows the command `terraform init` being executed.



The terraform Init is successful, now give terraform validate for the code



Now give terraform plan to know is any change is done

The screenshot shows a Visual Studio Code editor with a file explorer on the left displaying a project structure for 'GUVI-TASK'. The main editor window shows a Terraform configuration file named 'main.tf' with the following content:

```
1 provider "aws" {
2   region = "us-east-1"
3   alias  = "east"
4 }
5
6 provider "aws" {
7   region = "us-west-1"
8   alias  = "west"
9 }
10
11
12 resource "aws_instance" "east-1" {
13   provider = aws.east
14   ami      = "ami-08a6efd148b1f7504"
15   instance_type = "t2.micro"
16   security_groups = [aws_security_group.web-sg-east.name]
17 }
```

The terminal window at the bottom shows the output of the 'terraform plan' command:

```
Success! The configuration is valid.

PS C:\Guvi-Task\Terraform-Task-1> 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.east-1 will be created
+ resource "aws_instance" "east-1" {
```

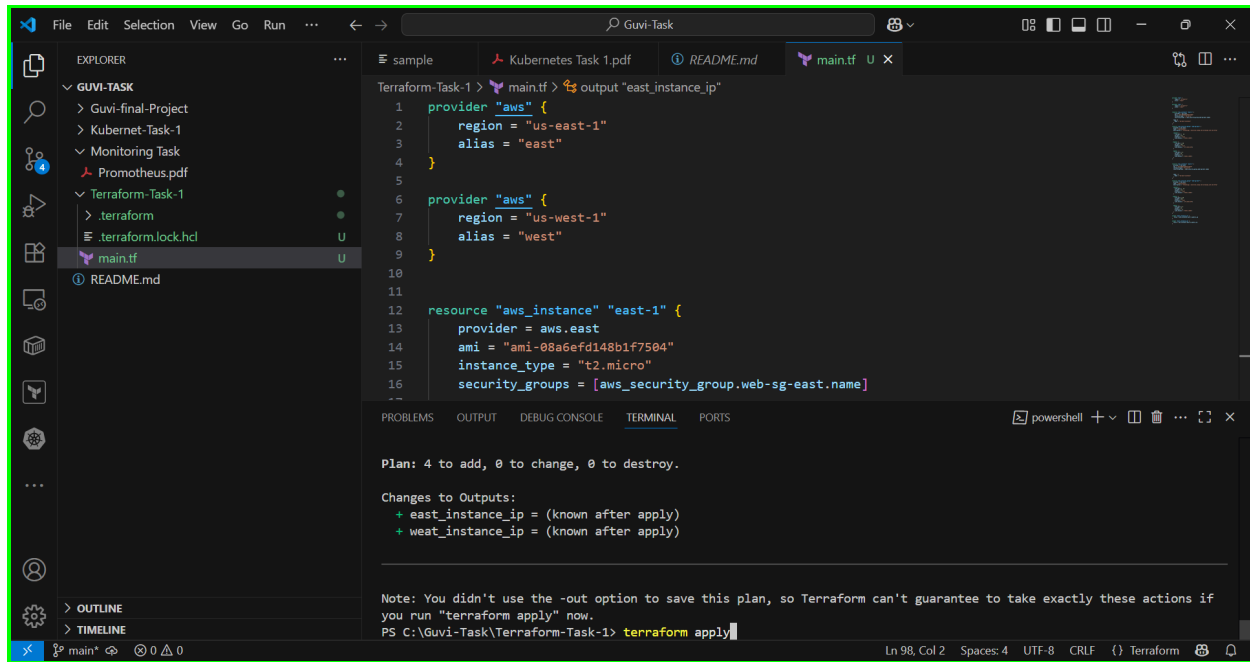
This screenshot shows the same Visual Studio Code editor with the 'main.tf' file. The terminal window displays the output of the 'terraform plan' command, showing the execution plan for the 'aws_instance.east-1' resource.

```
Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ east_instance_ip = (known after apply)
+ west_instance_ip = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee it takes exactly these actions if you run "terraform apply" now.
PS C:\Guvi-Task\Terraform-Task-1>
```

Finally apply terraform apply to execute the script



```
1 provider "aws" {
2   region = "us-east-1"
3   alias  = "east"
4 }
5
6 provider "aws" {
7   region = "us-west-1"
8   alias  = "west"
9 }
10
11
12 resource "aws_instance" "east-1" {
13   provider = aws.east
14   ami      = "ami-08a6efd148b1f7504"
15   instance_type = "t2.micro"
16   security_groups = [aws_security_group.web-sg-east.name]
17 }
```

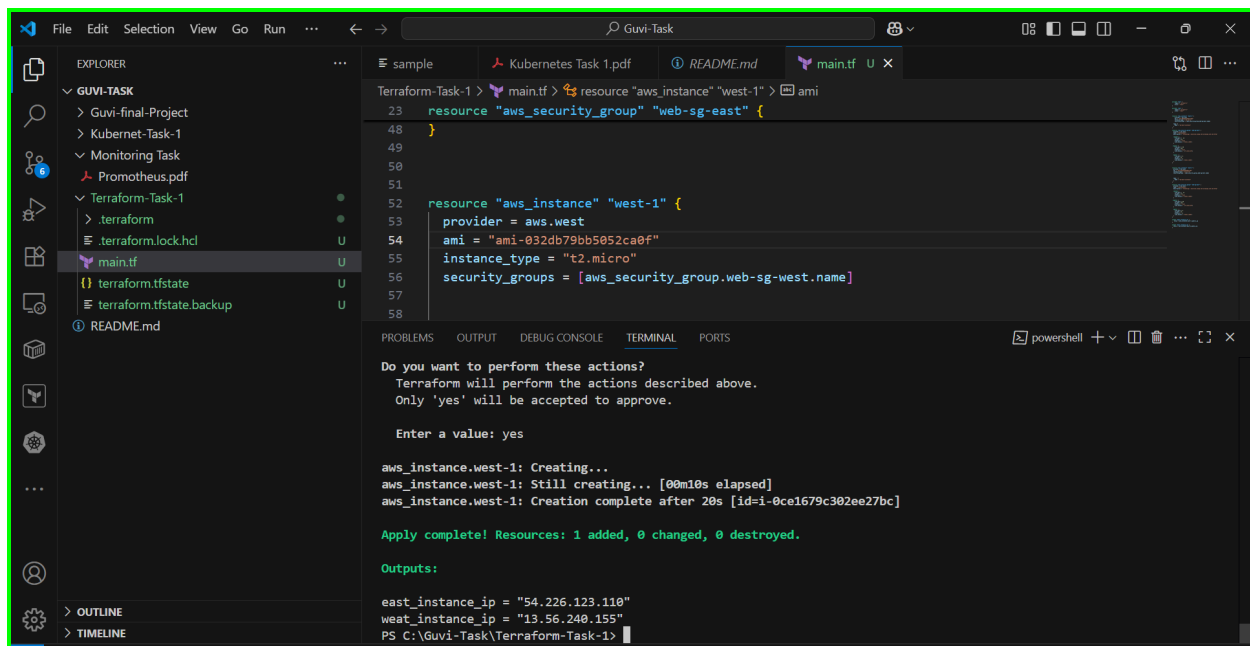
Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:

- + east_instance_ip = (known after apply)
- + west_instance_ip = (known after apply)

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 C:\Guvi-Task\Terraform-Task-1> terraform apply



```
23 resource "aws_security_group" "web-sg-east" {
24   name        = "web-sg-east"
25   description = "Security group for web server"
26   vpc_id      = aws_vpc.main.id
27   ingress {
28     from_port = 80
29     to_port   = 80
30     protocol  = "tcp"
31     cidr_blocks = ["0.0.0.0/0"]
32   }
33 }
34
35 resource "aws_instance" "west-1" {
36   provider = aws.west
37   ami      = "ami-032db79bb5052ca0f"
38   instance_type = "t2.micro"
39   security_groups = [aws_security_group.web-sg-west.name]
40 }
```

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.west-1: Creating...

aws_instance.west-1: Still creating... [00m10s elapsed]

aws_instance.west-1: Creation complete after 20s [id=i-0ce1679c302ee27bc]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

east_instance_ip = "54.226.123.110"

west_instance_ip = "13.56.240.155"

PS C:\Guvi-Task\Terraform-Task-1>

On the apply is done it get create the instance in two different region.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

EC2 > Instances

Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

All states

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	my-east-insta...	i-0aaa5dc3ffd97dd62	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-54-226-123-110

Select an instance

us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#Instances:

EC2 > Instances

Instances (1) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

All states

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	my-west-insta...	i-0ce1679c302ee27bc	Running	t2.micro	2/2 checks passed	View alarms +

Select an instance