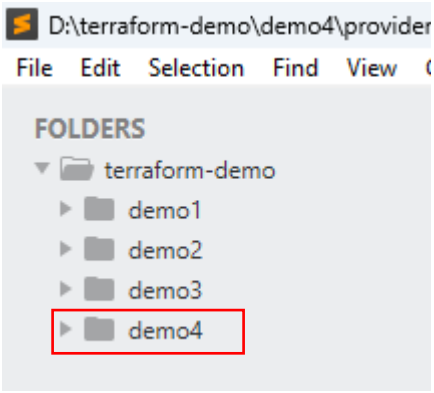
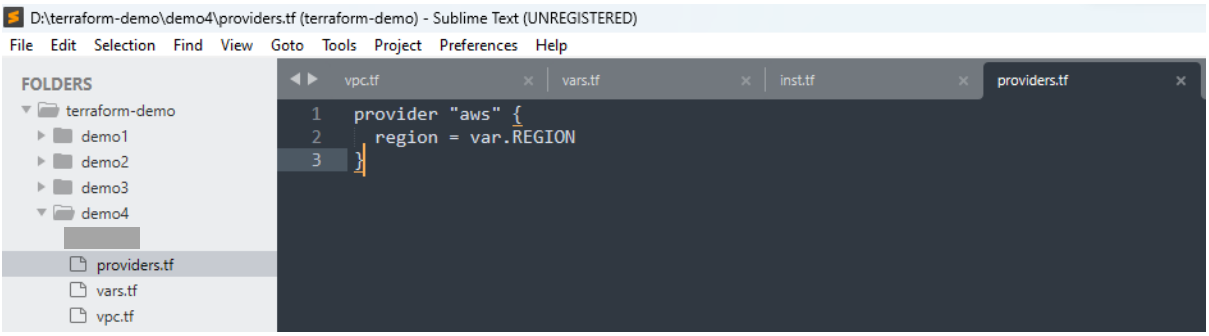
# 😊 Multi-Resources

1. So, in this lab you are going to launch multiple resource through terraform.
2. First you are going to create a new sub folder out of your main folder.




3. After that you are going to create multiple files which will contains the variables and resources.
4. Start with creating providers file then you are going to create variables file after that VPC file because we are going to launch the VPC.

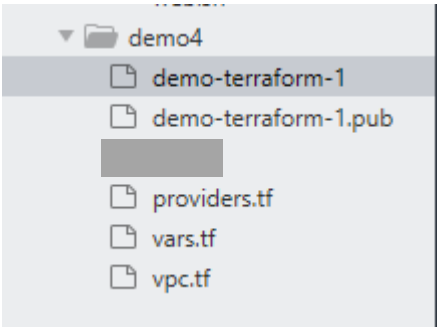



5. Now you are going to write the code in these files. In some files the code is long to get in the snapshot.


6. Again, you can get this code from GitHub.
7. Now you have to copy the public and private key to this folder which you had created in the previous lab.

   **cp demo-terraform-1 ../demo4/**
   **cp demo-terraform-.pub ../demo4/**

```
        LAPTOP-G2CAKBK8 MINGW64 /d/terraform-demo/demo3
$ cp demo-terraform-1 ../demo4/

        LAPTOP-G2CAKBK8 MINGW64 /d/terraform-demo/demo3
$ ls
demo-terraform-1      inst.tf       terraform.tfstate          vars.tf
demo-terraform-1.pub  providers.tf  terraform.tfstate.backup   web.sh*

        LAPTOP-G2CAKBK8 MINGW64 /d/terraform-demo/demo3
$ cp demo-terraform-1.pub ../demo4/
```




8. In the VPC file in the code we have created a VPC, subnets, route table and internet gateway.
9. Once you have copied your key and created your providers file, variable file and your VPC file.
10. Then open your gitbash and execute it, this will create your VPC.
11. Now in gitbash go into your current folder and do the listing of files.

```
$ ls
demo-terraform-1  demo-terraform-1.pub  providers.tf  vars.tf  vpc.tf
```

12. Now run the initialization command and initialize your code.

    **terraform init**

```
$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.38.0...
- Installed hashicorp/aws v5.38.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
```

13. Then run the validate and format command.

    **terraform validate**
    **terraform fmt**

```
$ terraform validate
Success! The configuration is valid.
```

14. Now run the plan command to see what is going to add. Here you can see that the VPC will be created and 12 things are going to add in it.
15. After that run the apply command to execute your resources.

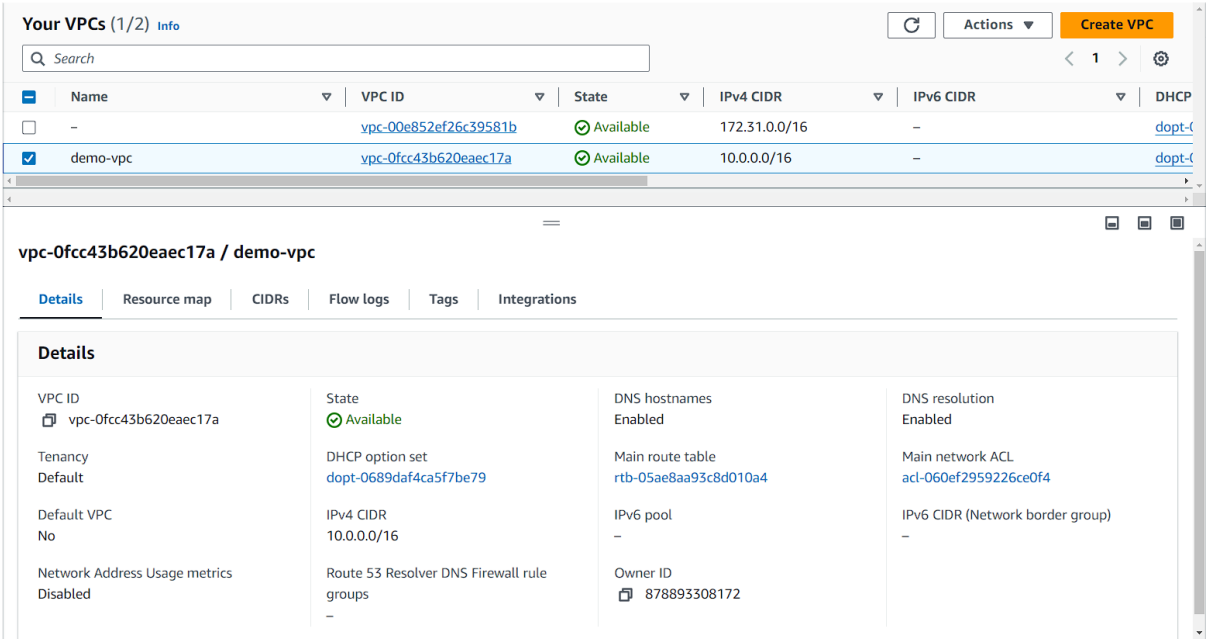
    **terraform plan**
    **terraform apply**

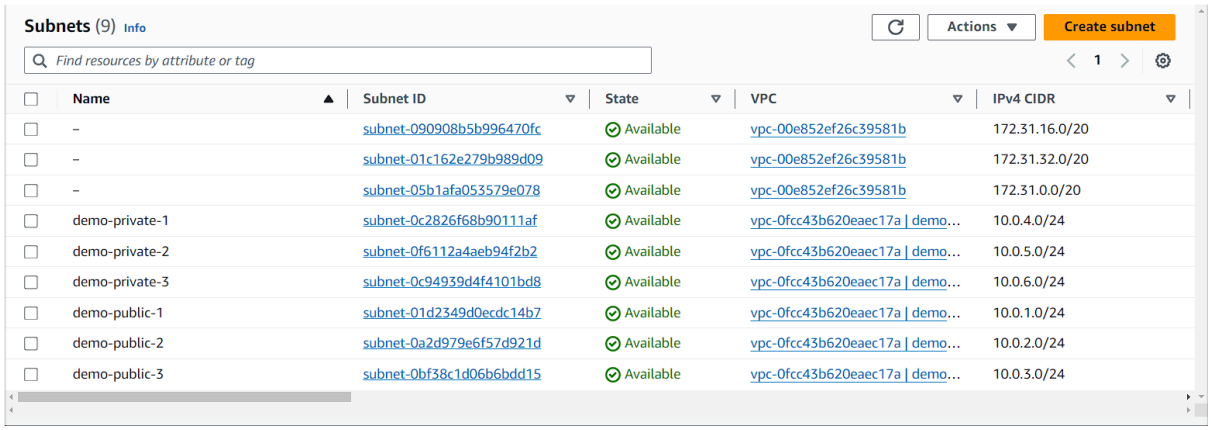
```
  # aws_vpc.demo will be created
  + resource "aws_vpc" "demo" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = true
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags                                 = {
          + "Name" = "demo-vpc"
        }
      + tags_all                             = {
          + "Name" = "demo-vpc"
        }
    }

Plan: 12 to add, 0 to change, 0 to destroy.
```

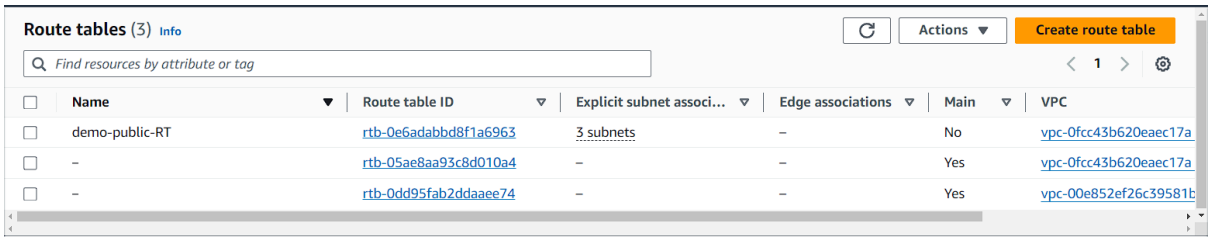
16. Here you can see that everything has been created. Now you can navigate to your console in the VPC you will see all the things that you have created so far.

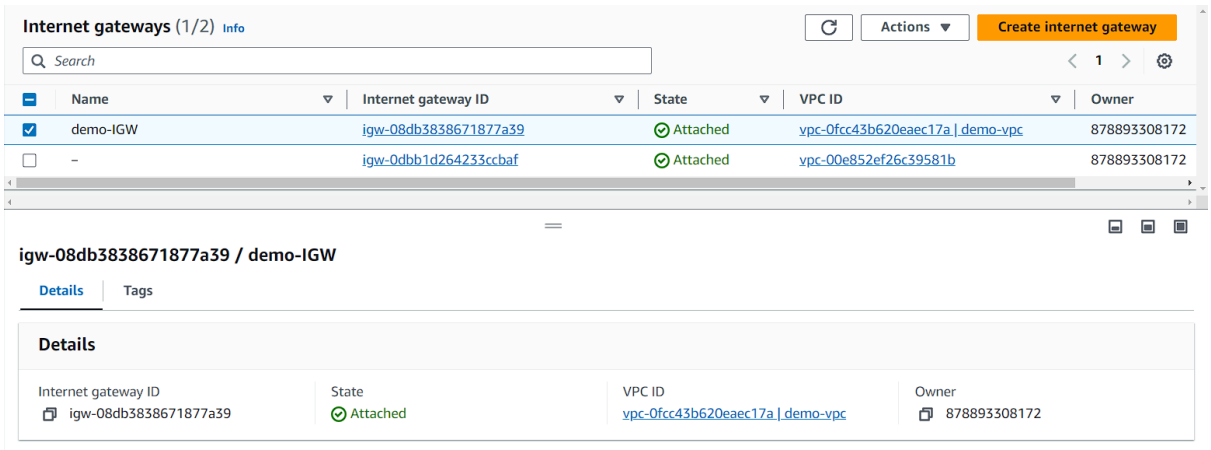
```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.demo: Creating...
aws_vpc.demo: Still creating... [10s elapsed]
aws_vpc.demo: Creation complete after 11s [id=vpc-0fcc43b620eaec17a]
aws_internet_gateway.demo-IGW: Creating...
aws_subnet.demo-public-3: Creating...
aws_subnet.demo-private-2: Creating...
aws_subnet.demo-public-2: Creating...
aws_subnet.demo-public-1: Creating...
aws_subnet.demo-private-3: Creating...
aws_subnet.demo-private-1: Creating...
aws_internet_gateway.demo-IGW: Creation complete after 1s [id=igw-08db3838671877a39]
aws_route_table.demo-public-RT: Creating...
aws_route_table.demo-public-RT: Creation complete after 1s [id=rtb-0e6adabbd8f1a6963]
aws_subnet.demo-public-3: Still creating... [10s elapsed]
aws_subnet.demo-public-1: Still creating... [10s elapsed]
aws_subnet.demo-private-2: Still creating... [10s elapsed]
aws_subnet.demo-public-2: Still creating... [10s elapsed]
aws_subnet.demo-private-1: Still creating... [10s elapsed]
aws_subnet.demo-private-3: Still creating... [10s elapsed]
aws_subnet.demo-private-3: Creation complete after 11s [id=subnet-0c94939d4f4101bd8]
aws_subnet.demo-public-2: Creation complete after 11s [id=subnet-0a2d979e6f57d921d]
aws_route_table_association.demo-public-2-a: Creating...
aws_subnet.demo-private-2: Creation complete after 11s [id=subnet-0f6112a4aeb94f2b2]
aws_subnet.demo-public-3: Creation complete after 11s [id=subnet-0bf38c1d06b6bdd15]
aws_route_table_association.demo-public-3-a: Creating...
aws_subnet.demo-public-1: Creation complete after 11s [id=subnet-01d2349d0ecdc14b7]
aws_route_table_association.demo-public-1-a: Creating...
aws_subnet.demo-private-1: Creation complete after 11s [id=subnet-0c2826f68b90111af]
aws_route_table_association.demo-public-2-a: Creation complete after 1s [id=rtbassoc-0ecc0c4aea34aab1e]
aws_route_table_association.demo-public-3-a: Creation complete after 1s [id=rtbassoc-09dda50f7b3fe115f]
aws_route_table_association.demo-public-1-a: Creation complete after 1s [id=rtbassoc-0da500b51540846a2]

Apply complete! Resources: 12 added, 0 changed, 0 destroyed.
```

17. Below you can see the resources that are created.

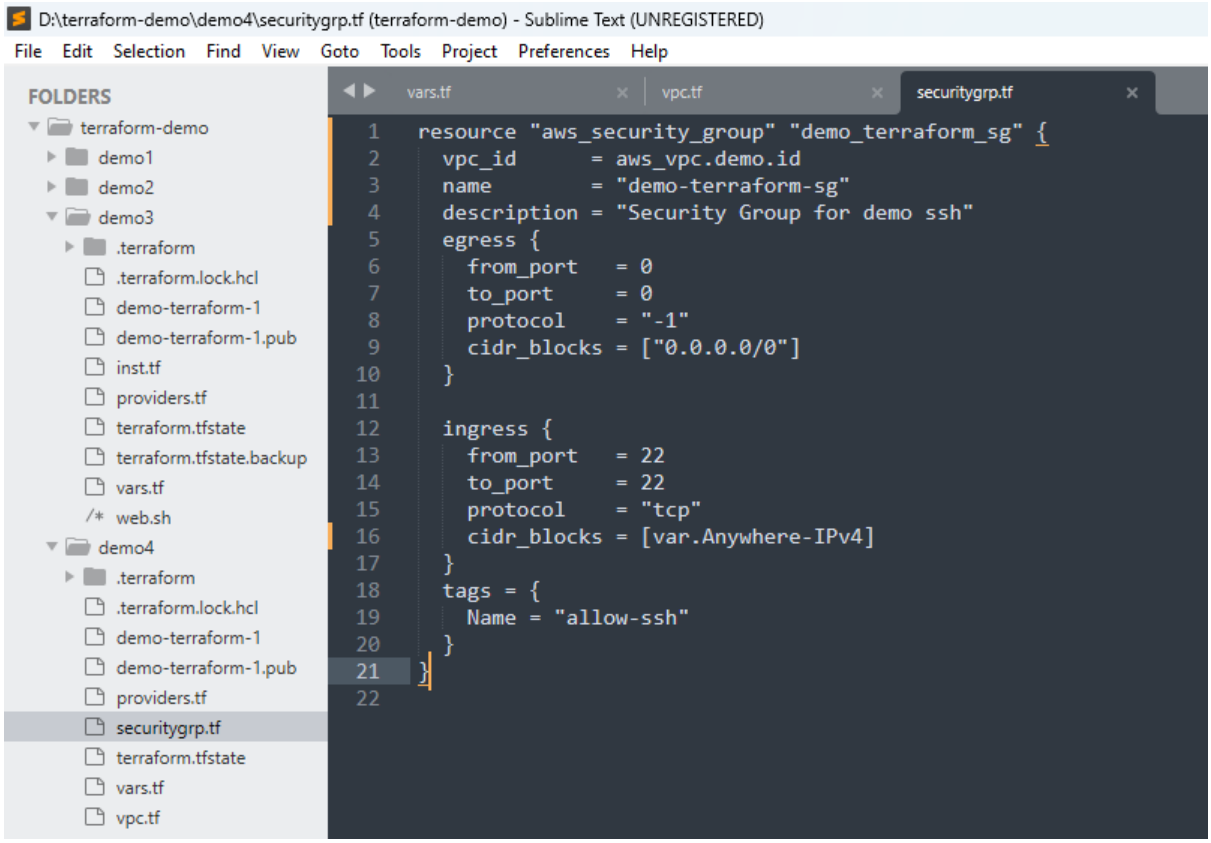




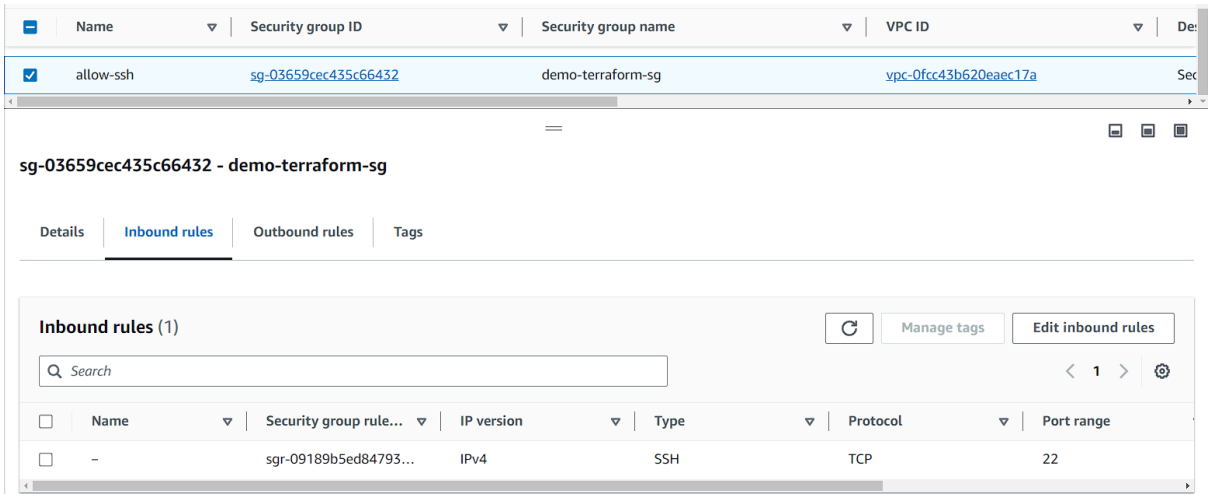




18. Now we are going to create the security group.

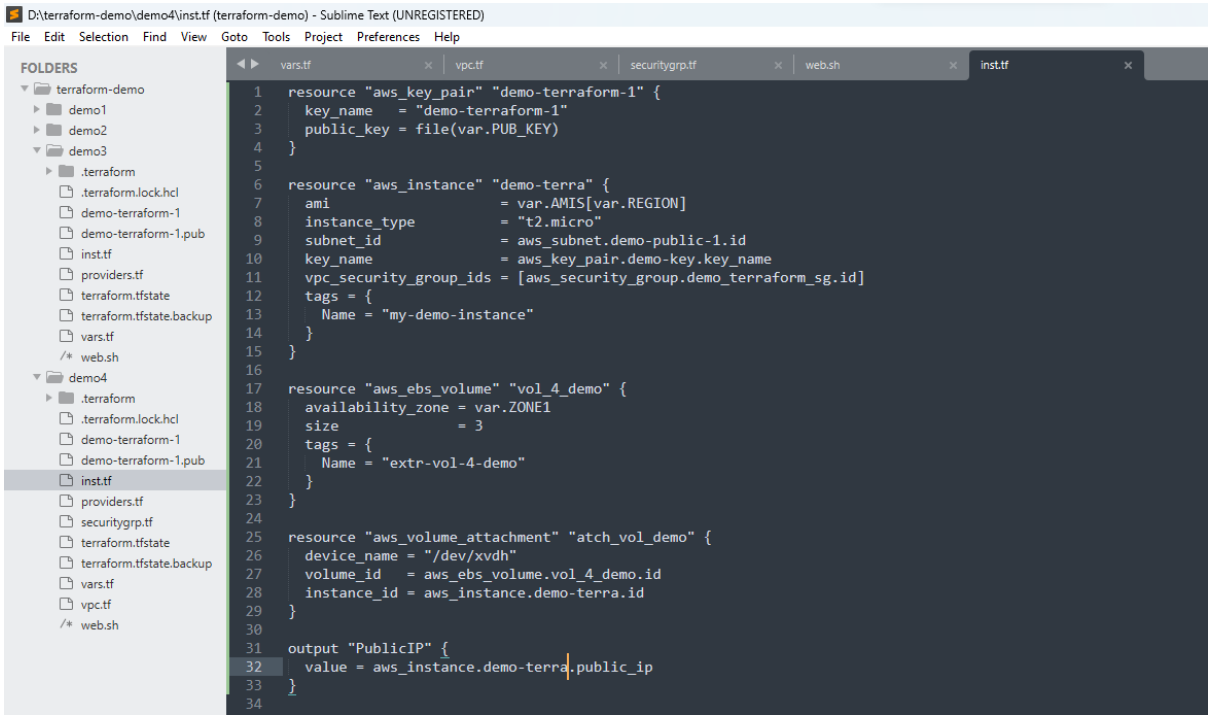



19. Again, go back to gitbash and first run plan command because this will check for the new resource afterwards run the apply command to execute our resource.
20. Below you can see that it has added one resource and then it created it and applied it.
21. You can also check your security group on the console as well.

```
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_security_group.demo_terraform_sg: Creating...
aws_security_group.demo_terraform_sg: Creation complete after 3s [id=sg-03659cec435c66432]

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




22. Now you can also copy your web.sh file from previous lab to this lab.
23. After that create your instance file.



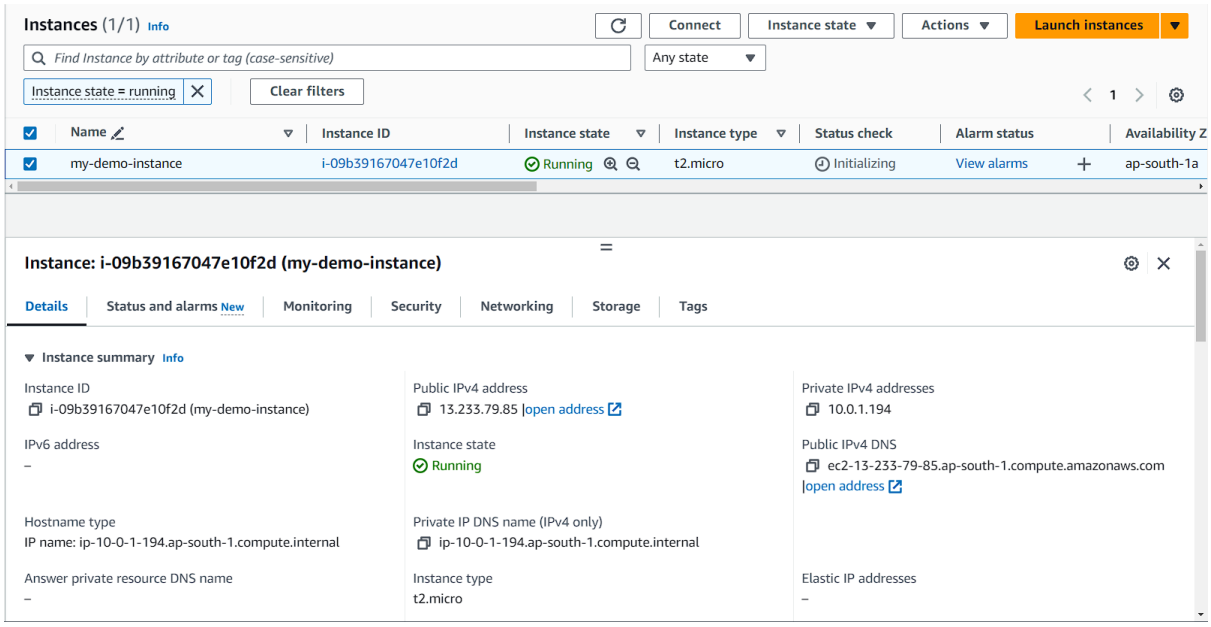

24. Once you are done setting up your instance. Then run the validate command to validate your code.
25. After that run the plan and apply command to execute your resources.

```
$ terraform validate
Success! The configuration is valid.
```

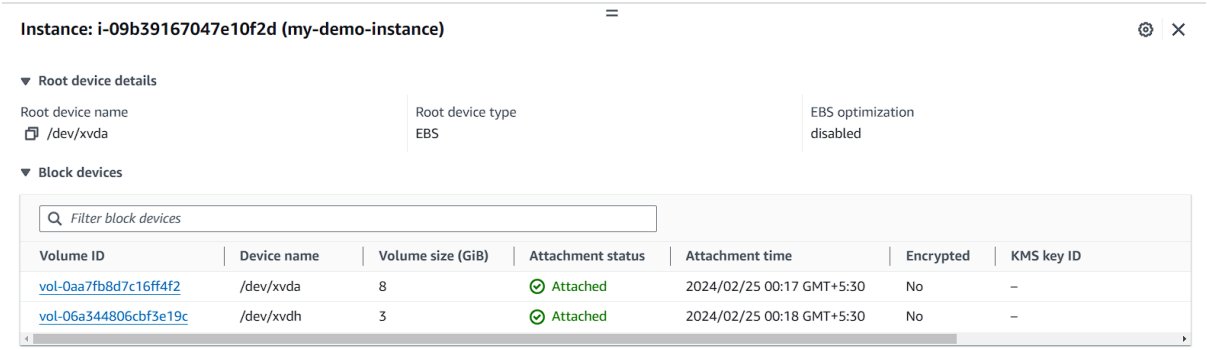
```
  # aws_key_pair.demo-terraform-1 will be created
  + resource "aws_key_pair" "demo-terraform-1" {
      + arn             = (known after apply)
      + fingerprint     = (known after apply)
      + id              = (known after apply)
      + key_name        = "demo-terraform-1"
      + key_name_prefix = (known after apply)
      + key_pair_id     = (known after apply)
      + key_type        = (known after apply)
      + public_key      = "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO3+sM29I9+KJ8fPldt1X3rxQlbfwUMvvRiyCNe60kEa PULKIT@LAPTOP-G2CAKBK8"
      + tags_all        = (known after apply)
    }

  # aws_volume_attachment.atch_vol_demo will be created
  + resource "aws_volume_attachment" "atch_vol_demo" {
      + device_name = "/dev/xvdh"
      + id          = (known after apply)
      + instance_id = (known after apply)
      + volume_id   = (known after apply)
    }

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

Changes to Outputs:
  + PublicIP = (known after apply)
```

```
aws_key_pair.demo-terraform-1: Creating...
aws_ebs_volume.vol_4_demo: Creating...
aws_key_pair.demo-terraform-1: Creation complete after 1s [id=demo-terraform-1]
aws_instance.demo-terra: Creating...
aws_ebs_volume.vol_4_demo: Still creating... [10s elapsed]
aws_ebs_volume.vol_4_demo: Creation complete after 11s [id=vol-06a344806cbf3e19c]
aws_instance.demo-terra: Still creating... [10s elapsed]
aws_instance.demo-terra: Still creating... [20s elapsed]
aws_instance.demo-terra: Still creating... [30s elapsed]
aws_instance.demo-terra: Creation complete after 31s [id=i-09b39167047e10f2d]
aws_volume_attachment.atch_vol_demo: Creating...
aws_volume_attachment.atch_vol_demo: Still creating... [10s elapsed]
aws_volume_attachment.atch_vol_demo: Still creating... [20s elapsed]
aws_volume_attachment.atch_vol_demo: Creation complete after 21s [id=vai-149348497]

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.

Outputs:

PublicIP = "13.233.79.85"
```

26. Below you can see the instance.



27. Here you can see the storage too, the extra 3 gb that we add and attached to it.




28. Once you are done just run the destroy command and destroy all of your resources.