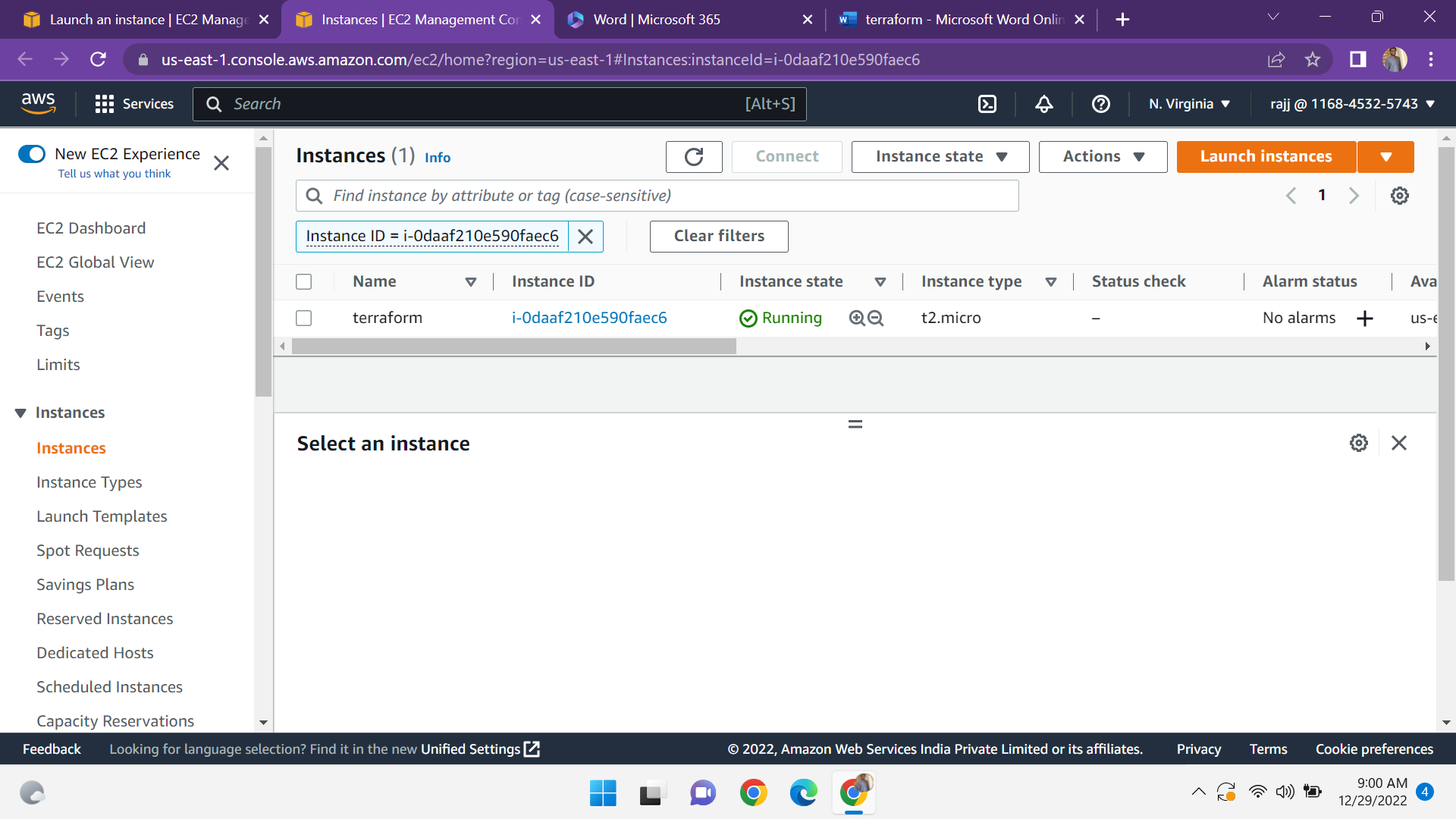
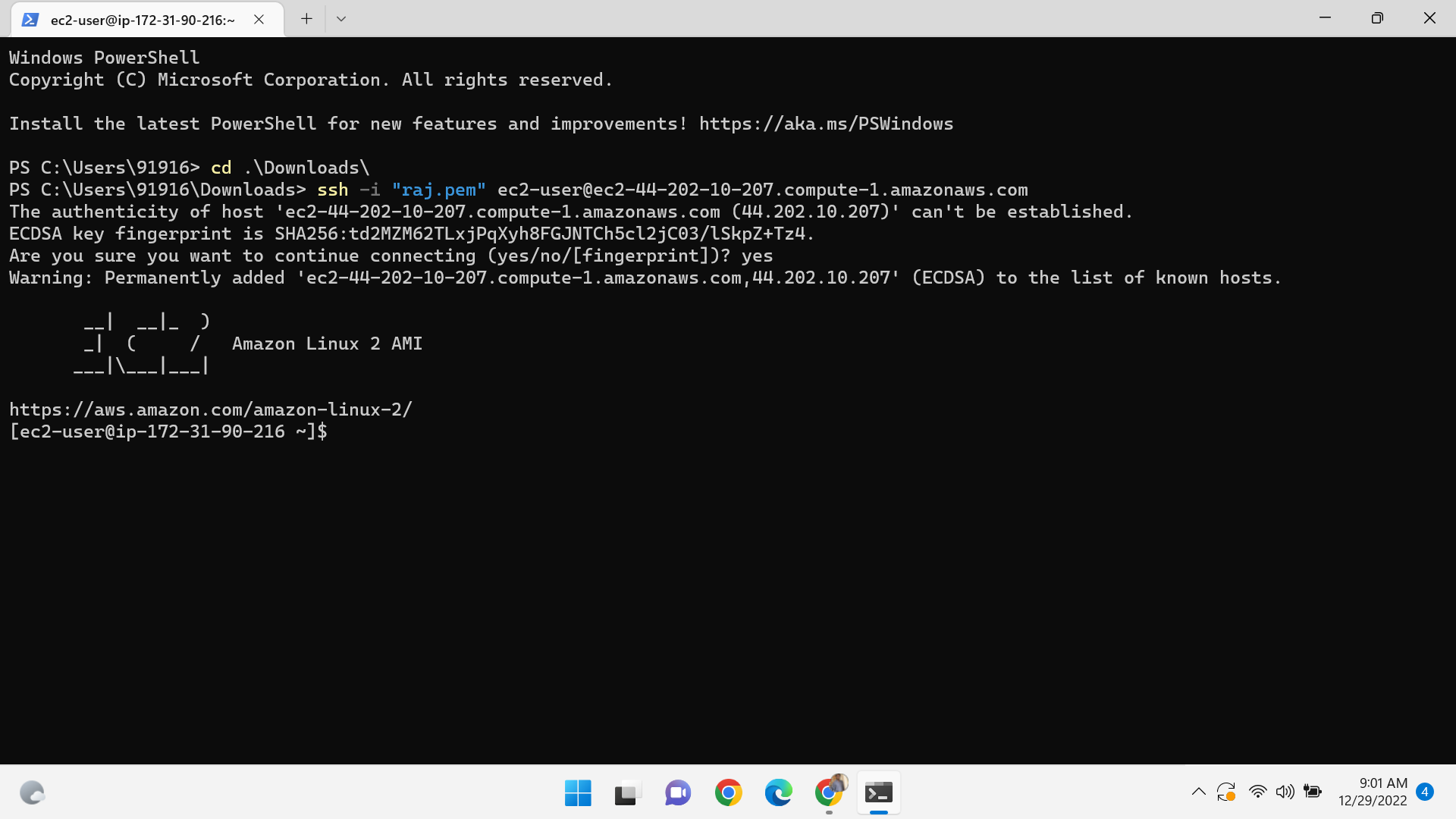
TERRAFORM

Terraform- Terraform is an open-source infrastructure as a code (IAC) tool that allows to create, manage & deploy the production-ready environment. Terraform codifies cloud APIs into declarative configuration files. Terraform can manage both existing service providers and custom in-house solutions.

Create a ec2 with amazon linux server and security group as ssh and keypair.



And the instance is connected to terminal.

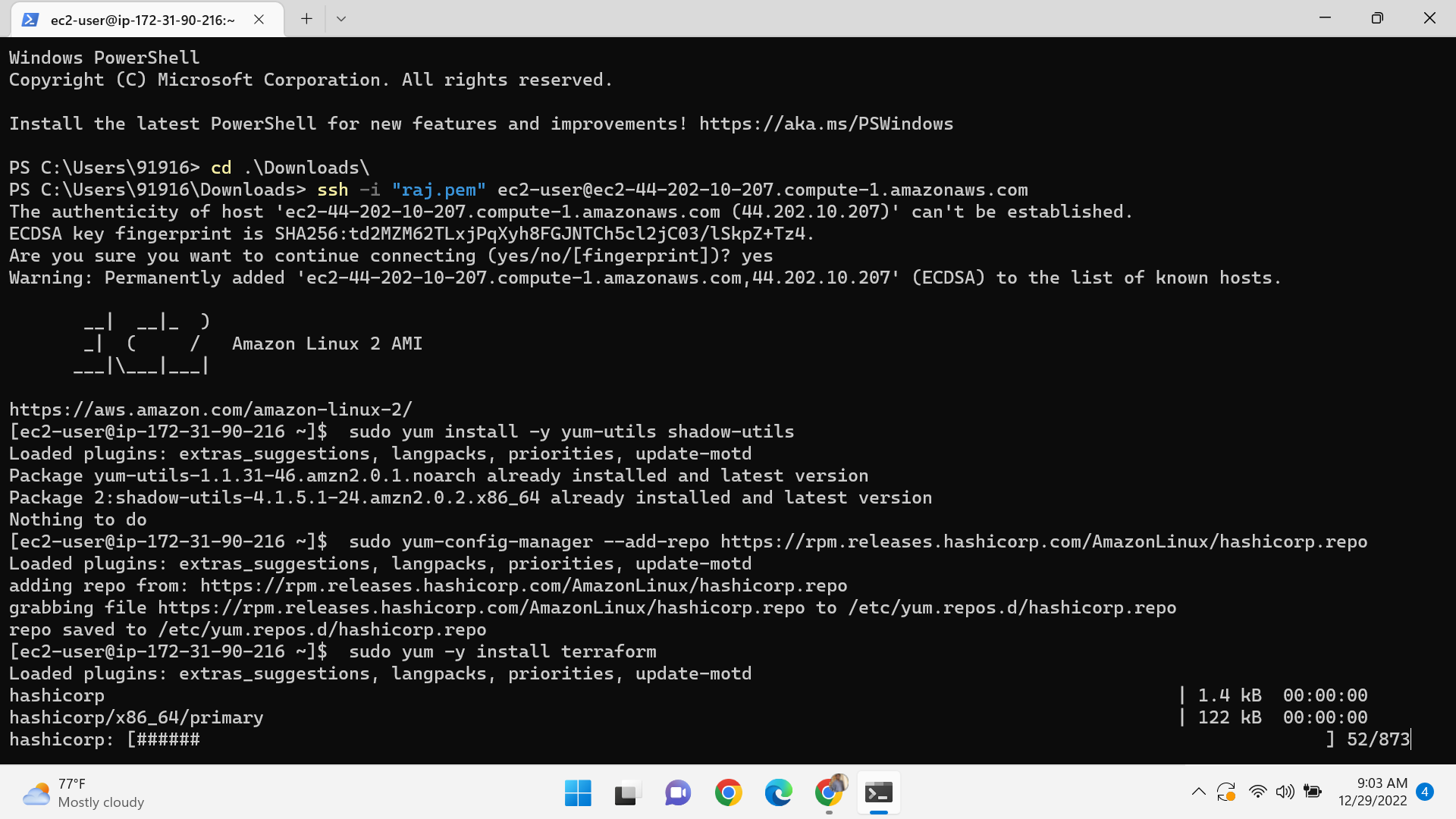


Update the server-and install the terraform and install the packages

sudo yum install -y yum-utils shadow-utils

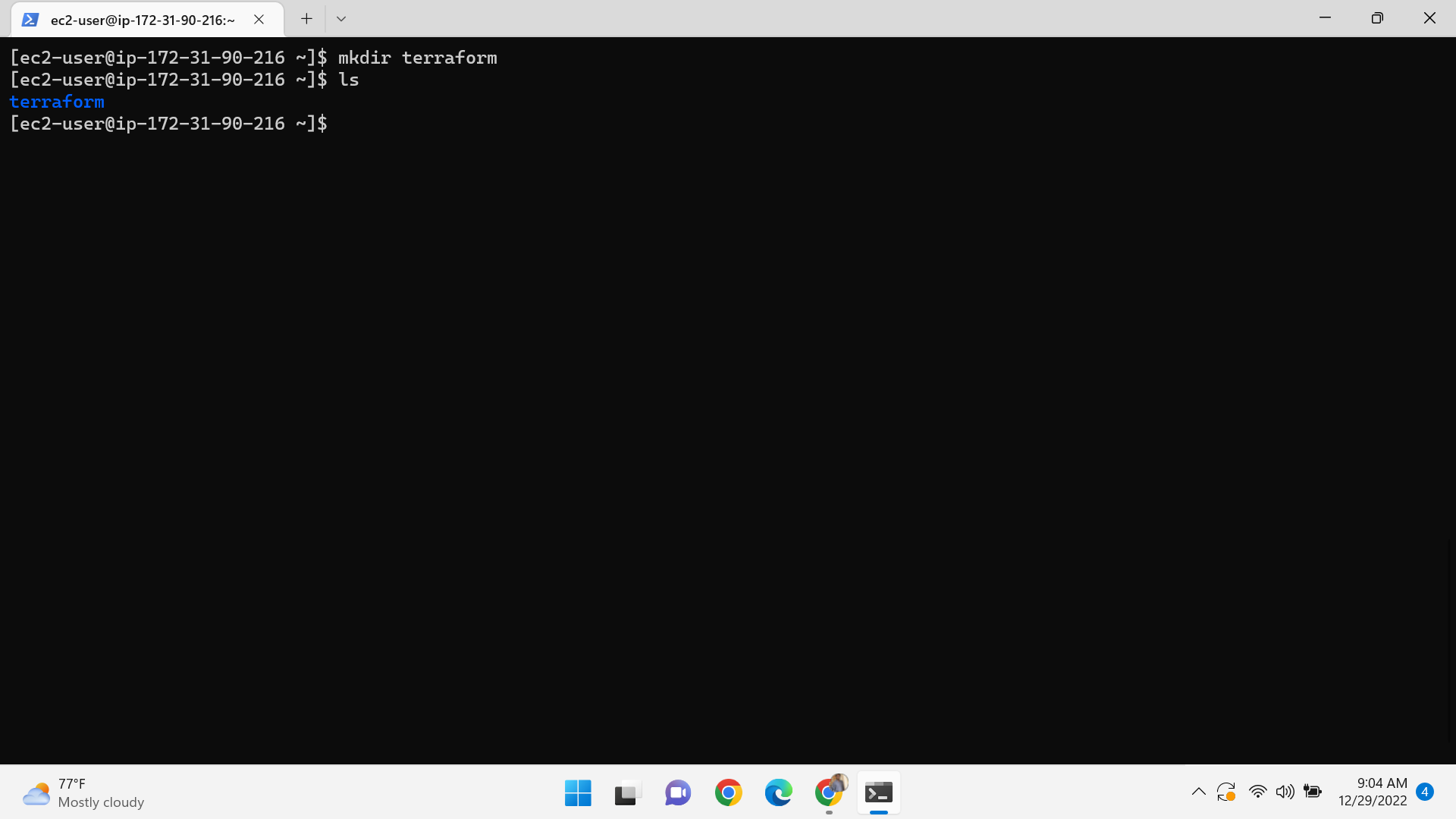
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

Sudo yum –y install terraform

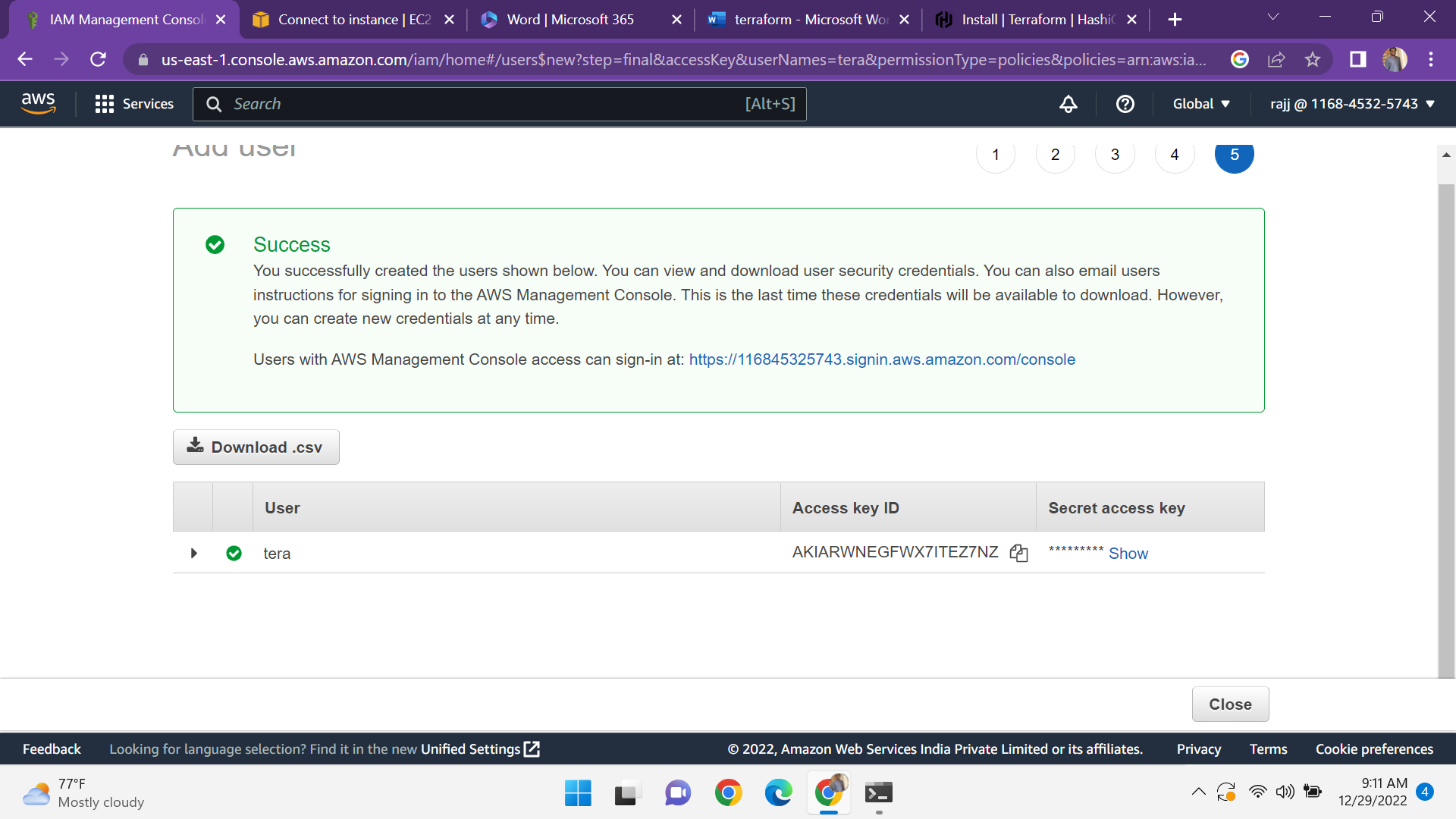


And make a directory by using the command

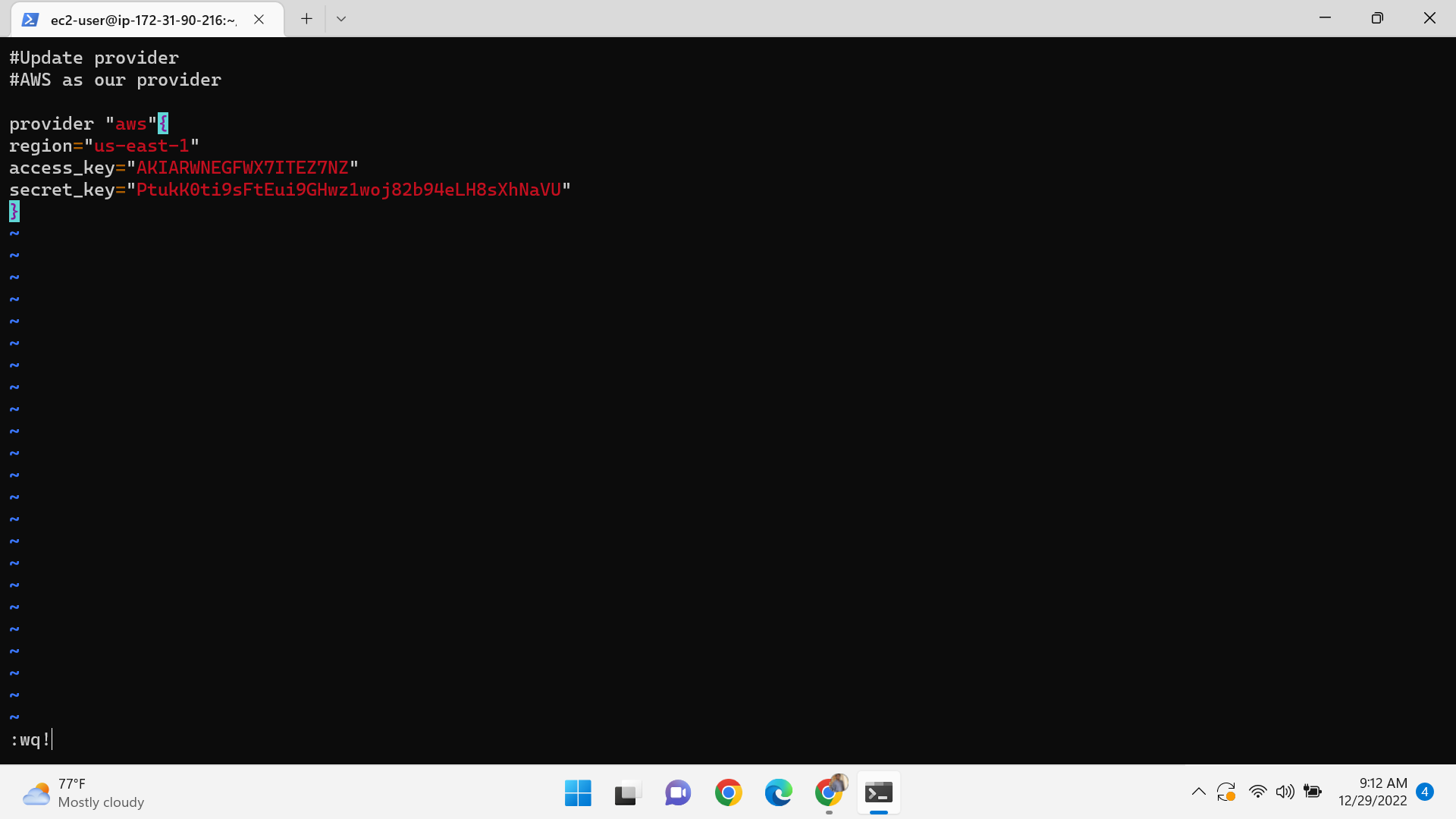
Mkdir terraform



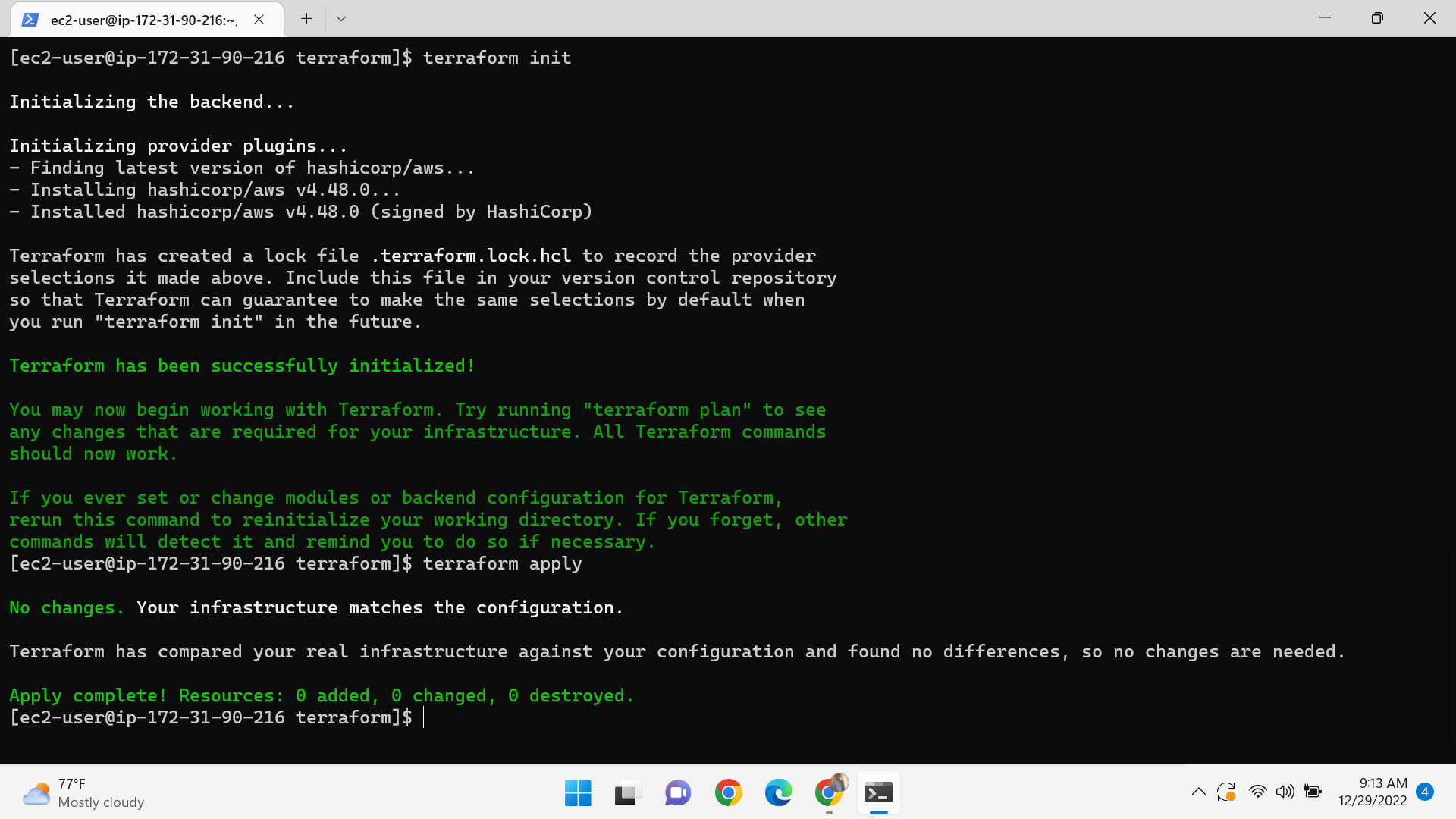
And create a iam user account and give the programmatic access and give the administration access



And in terminal create a provider.tf file and give the aws as a provider and give the access key and secret key created in iam

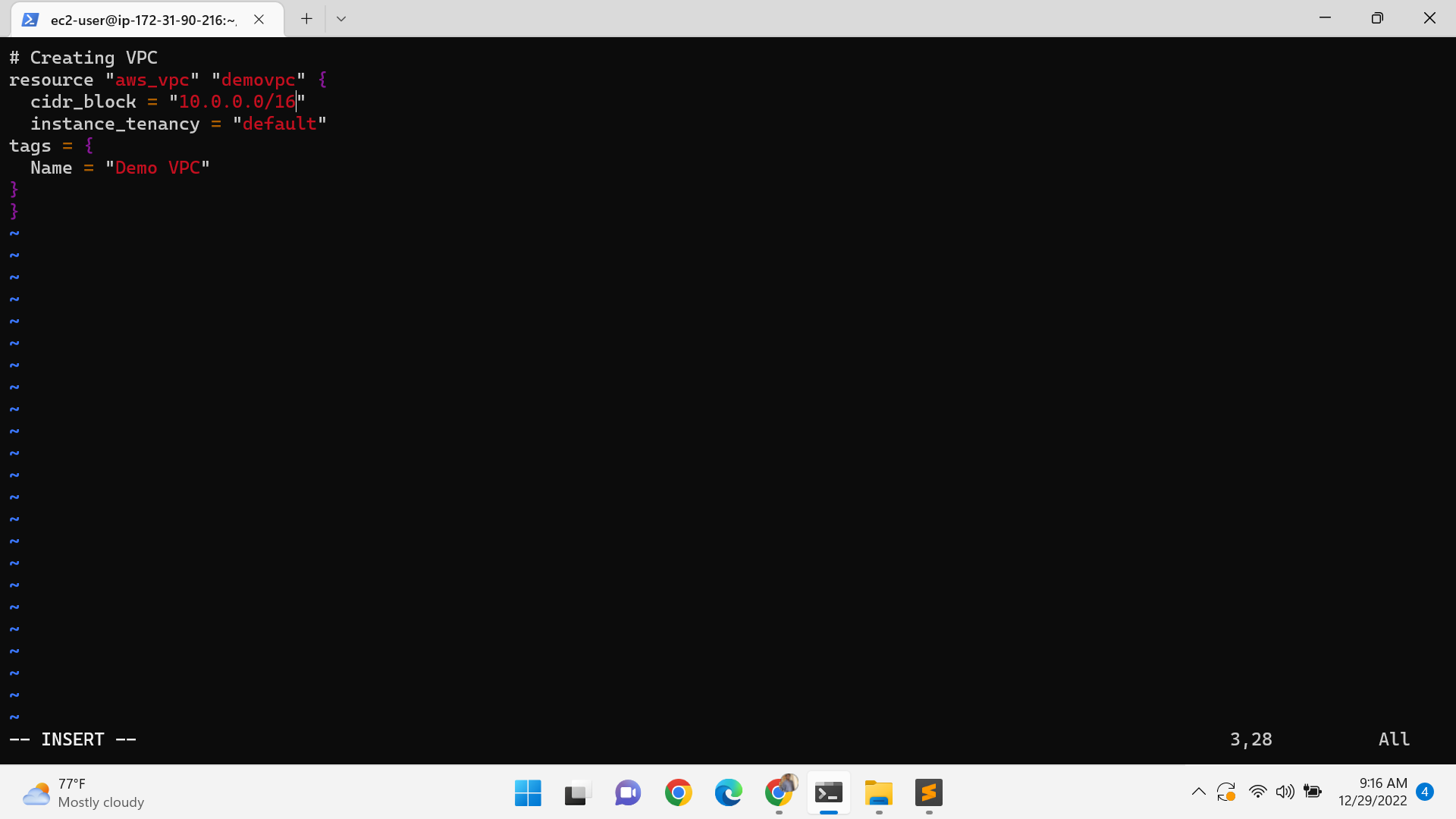


Then initialiaze the terraform and terraform apply



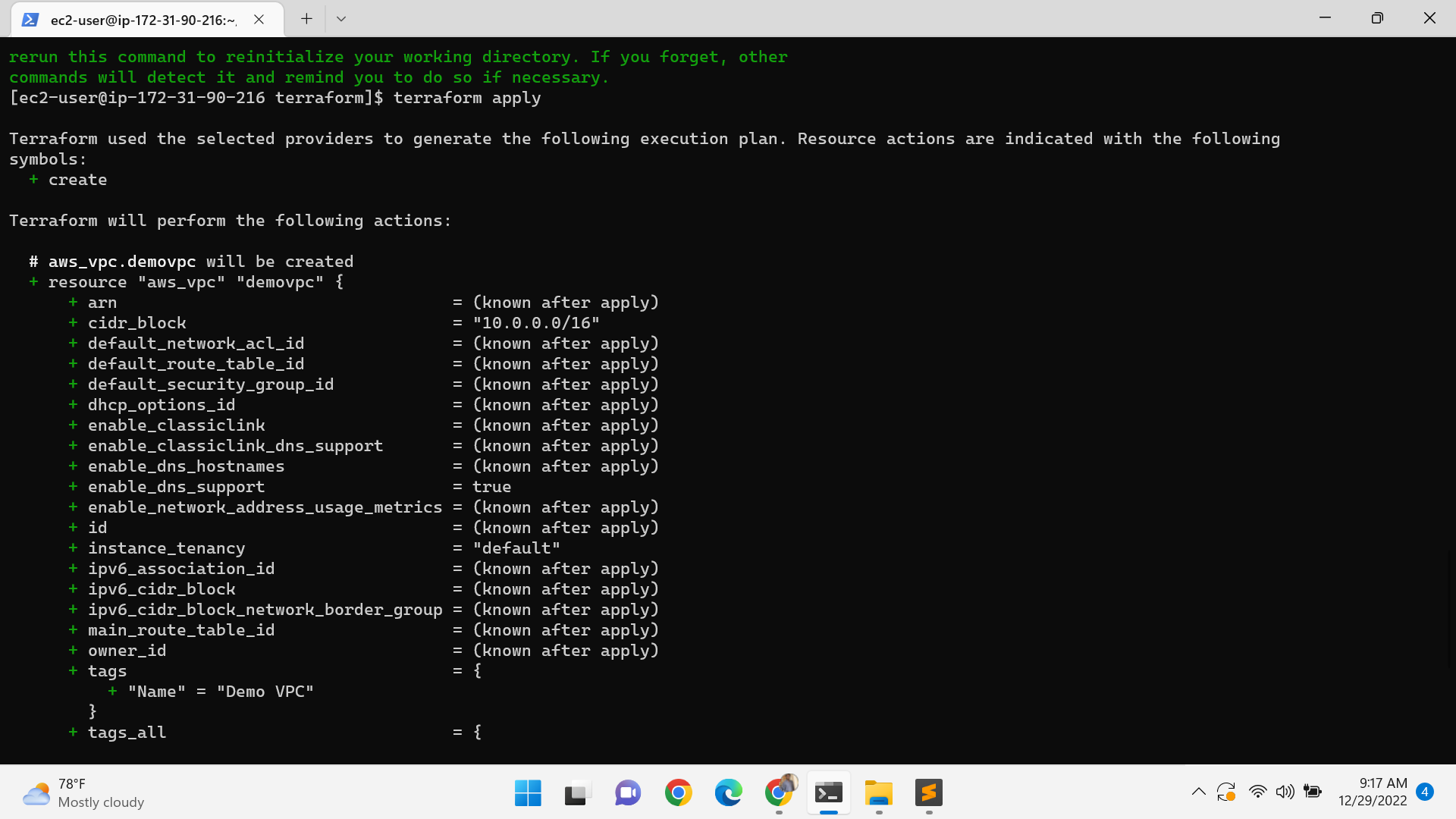
Create vpc.tf file by using the command

Vi vpc.tf

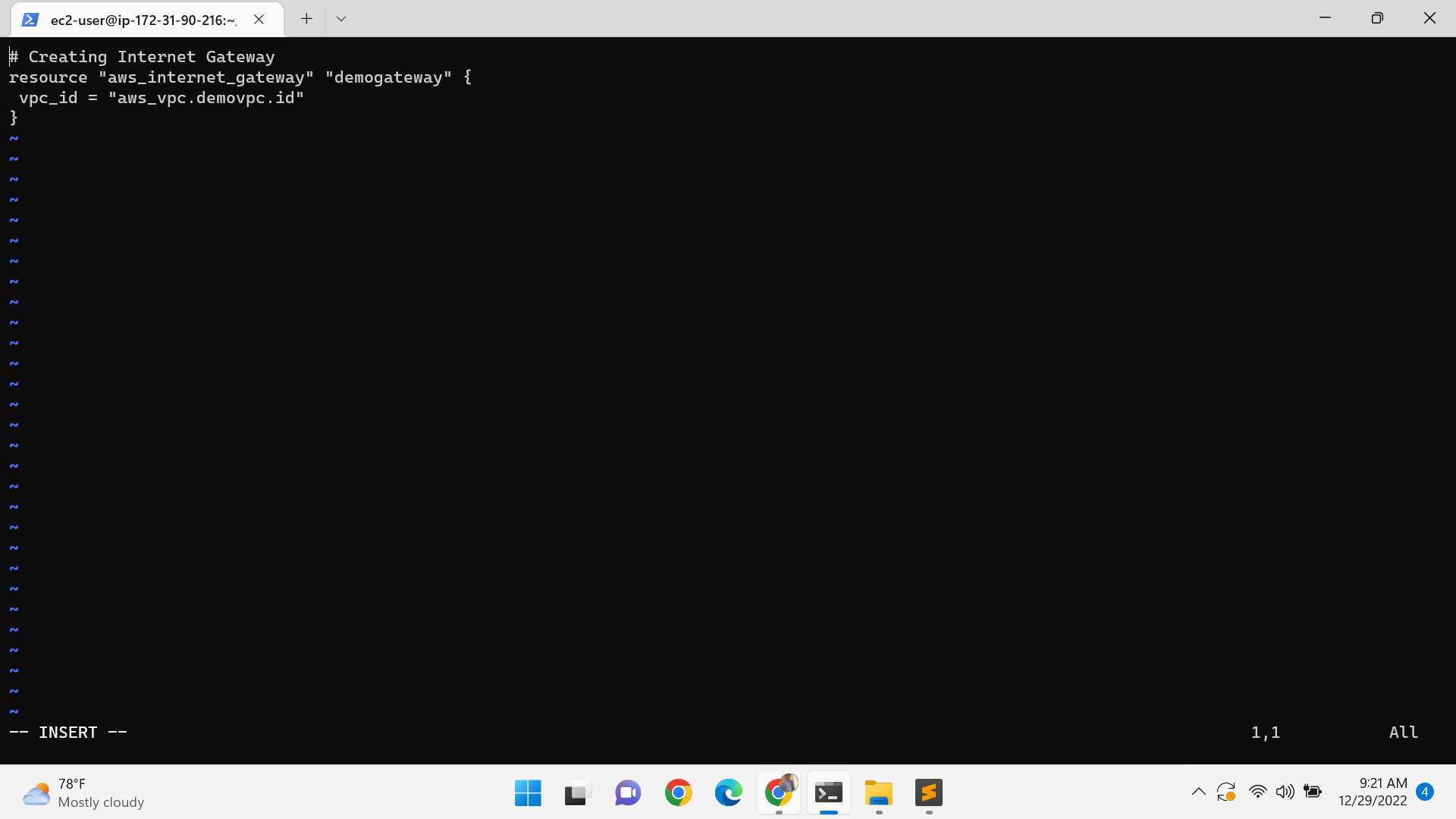


Save the script and init the terraform-terraform init

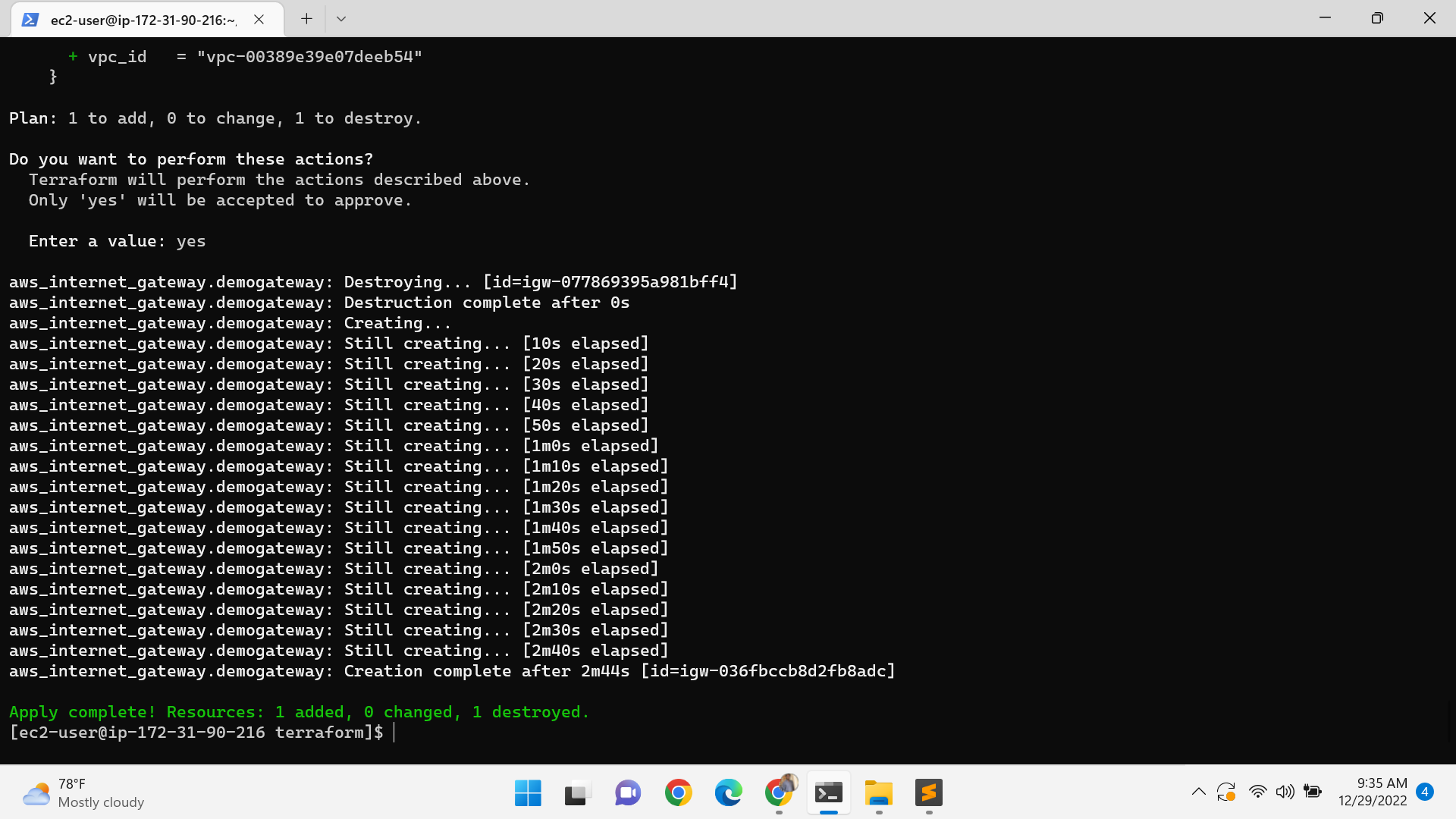
And terraform apply



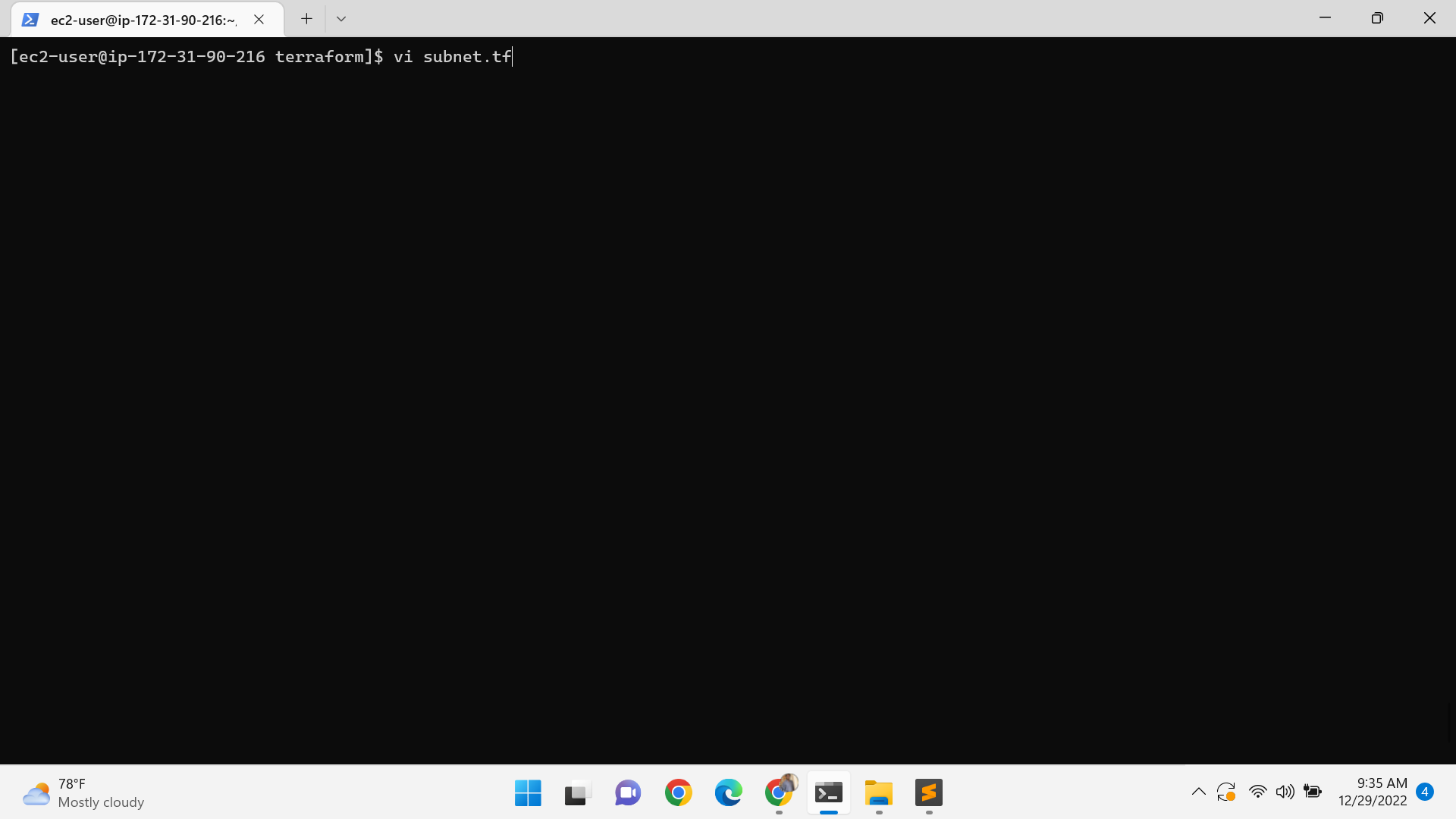
Create a internet gateway and the gateway is connected to the vpc and create file –vi igw.tf

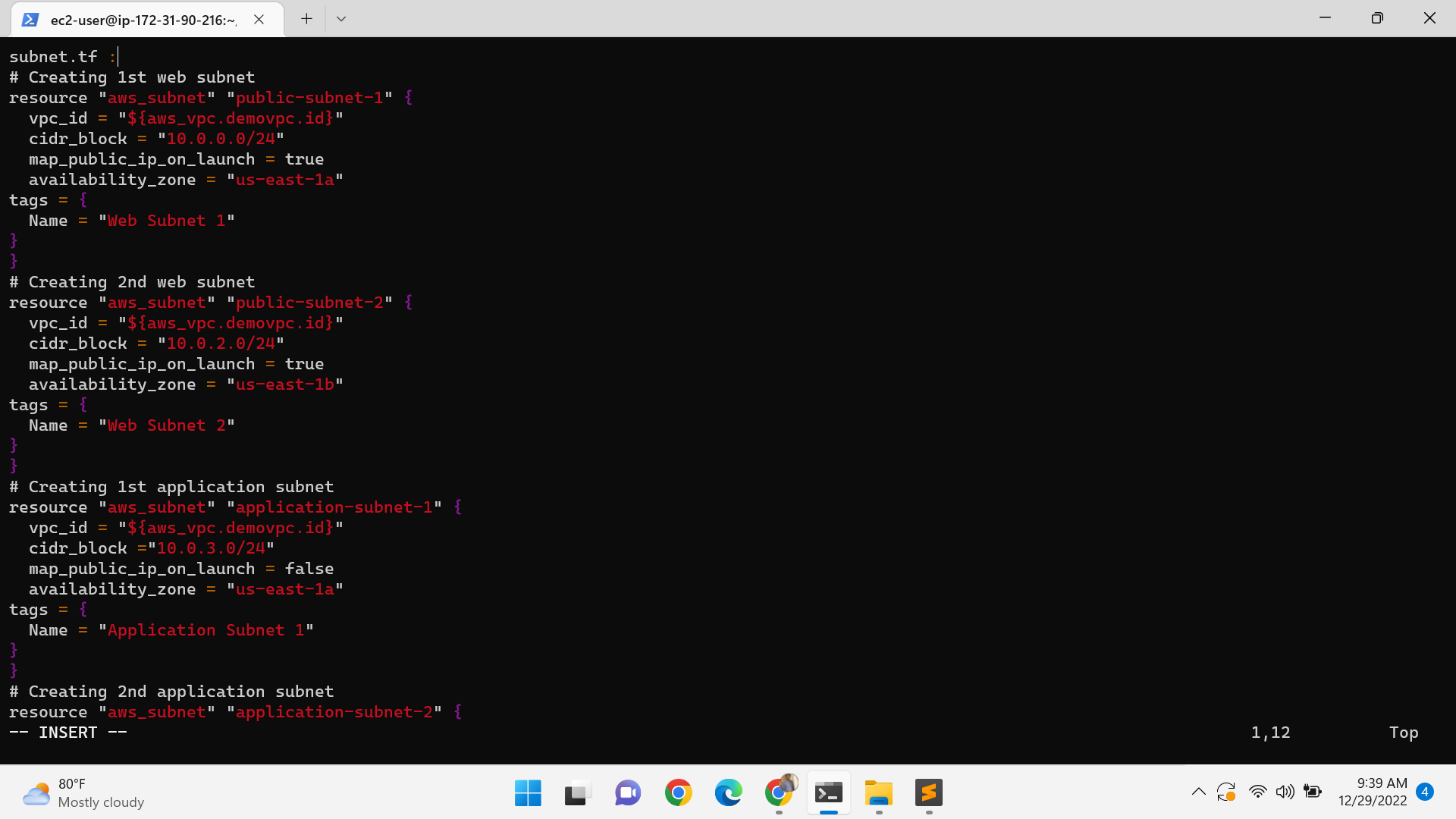


And again initialise the terraform and terraform apply the internet gateway created it is connected to vpc

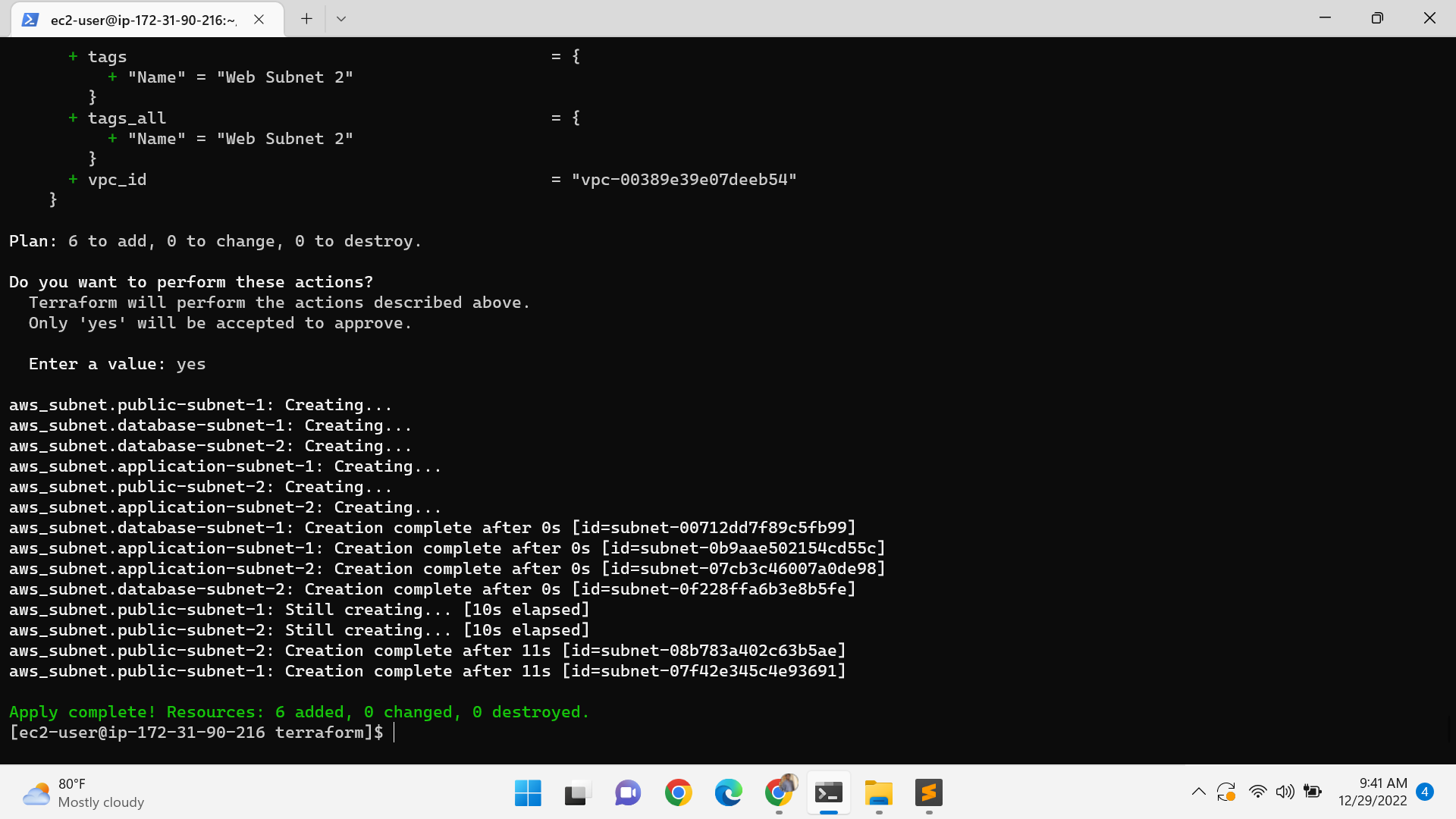


Create a subnet-vi subnet.tf



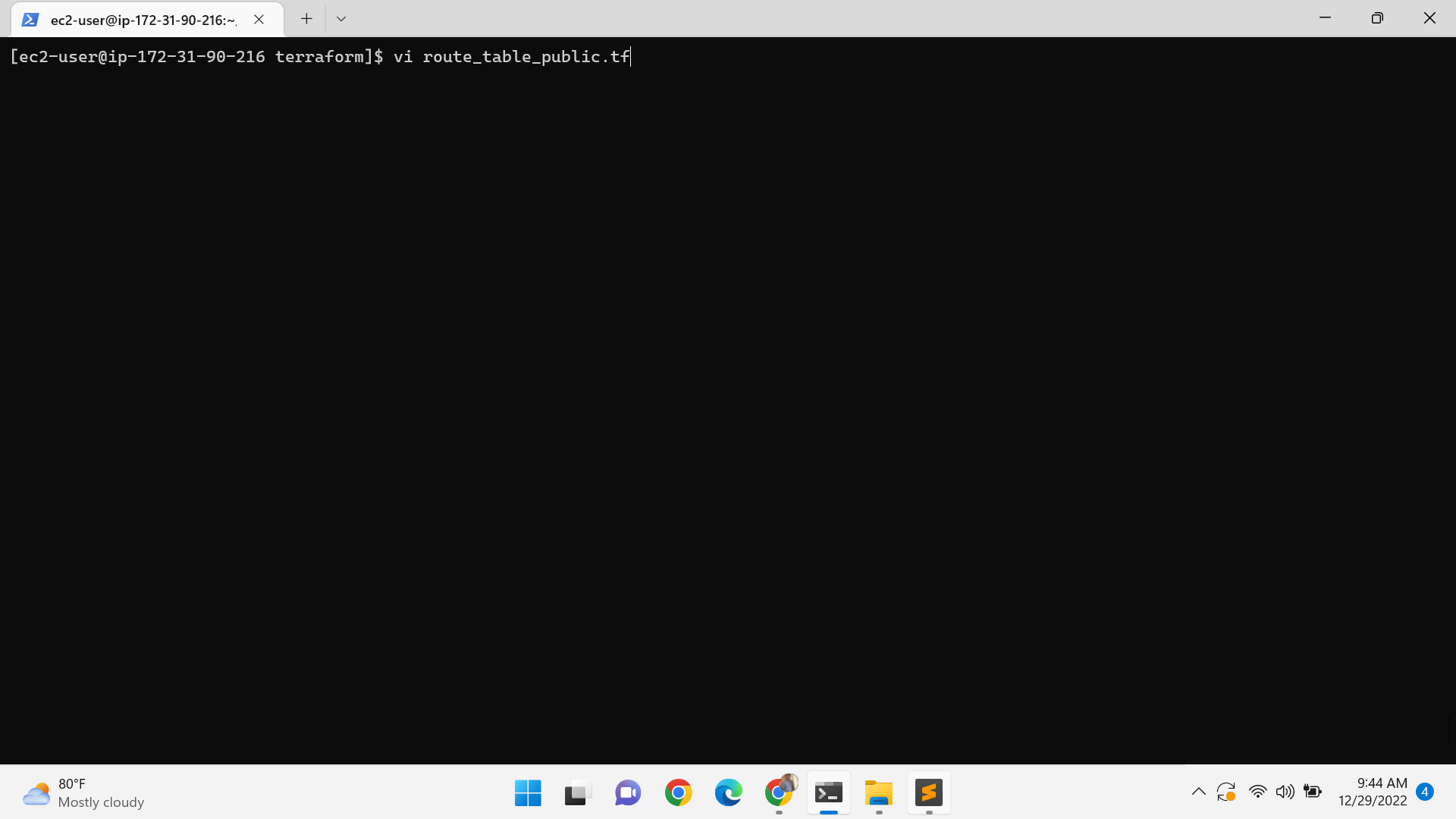


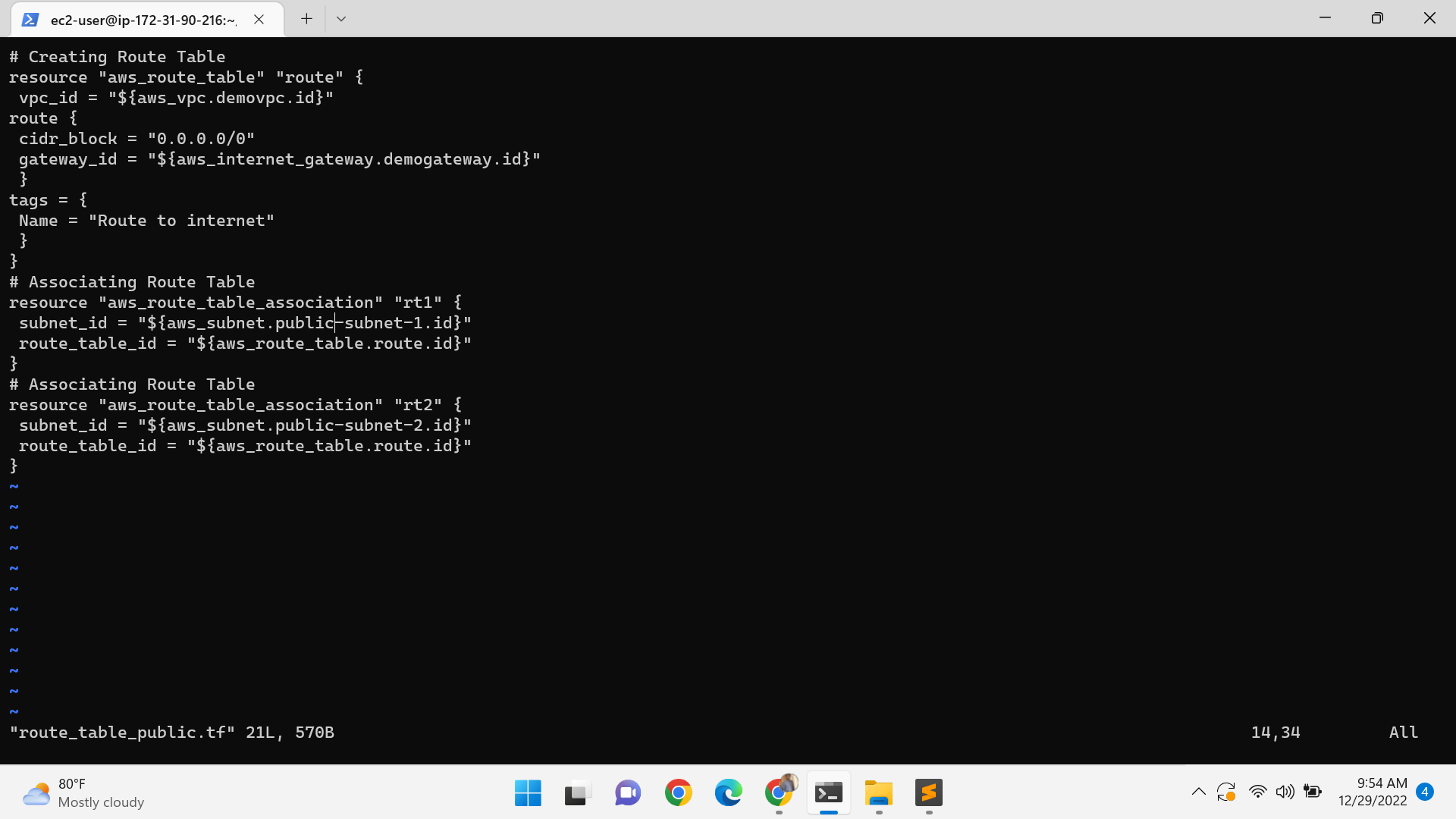
Terraform init and terraform apply subnets can be created



Create a route table with the name

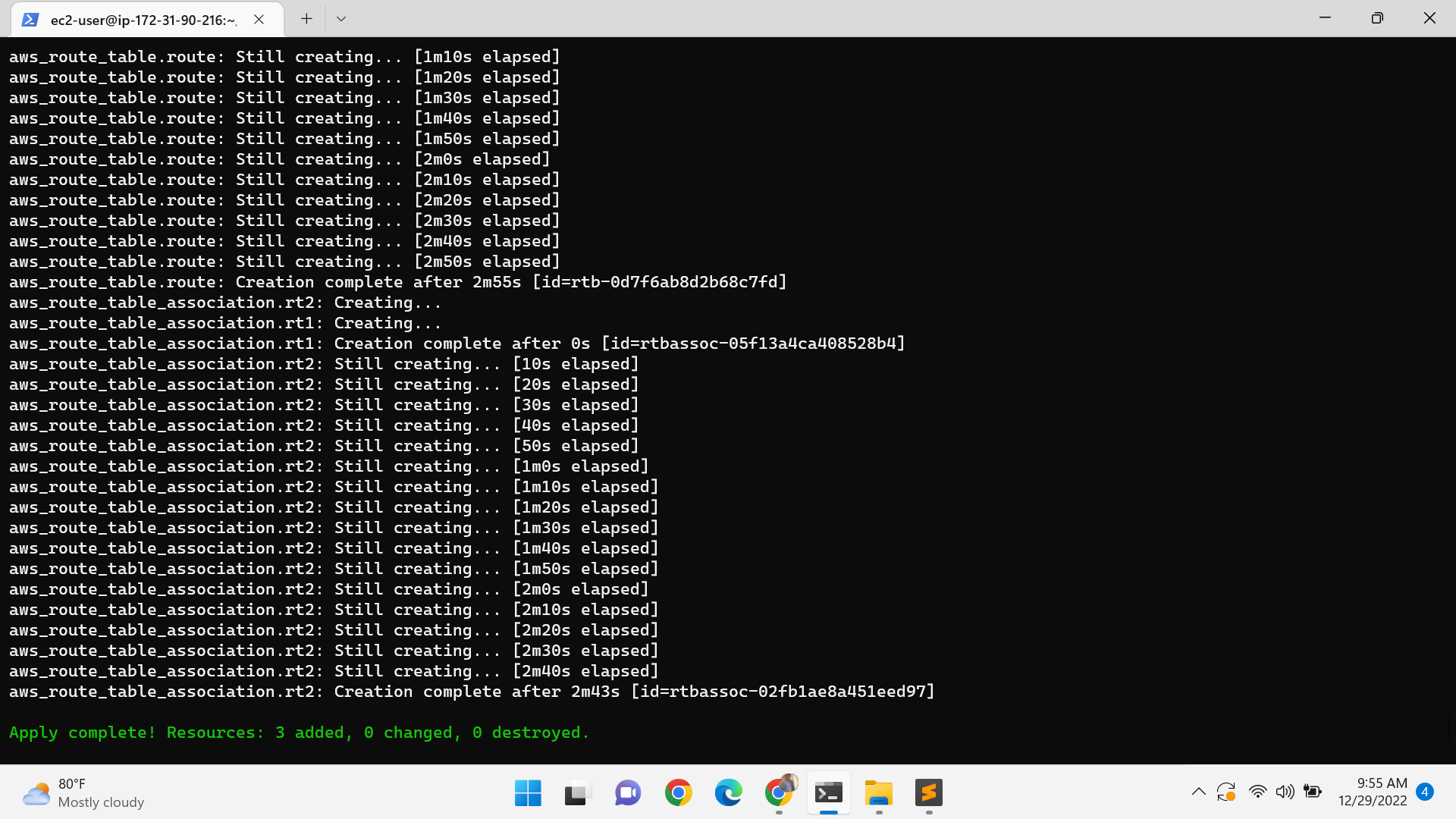
Vi route\_public\_table.tf



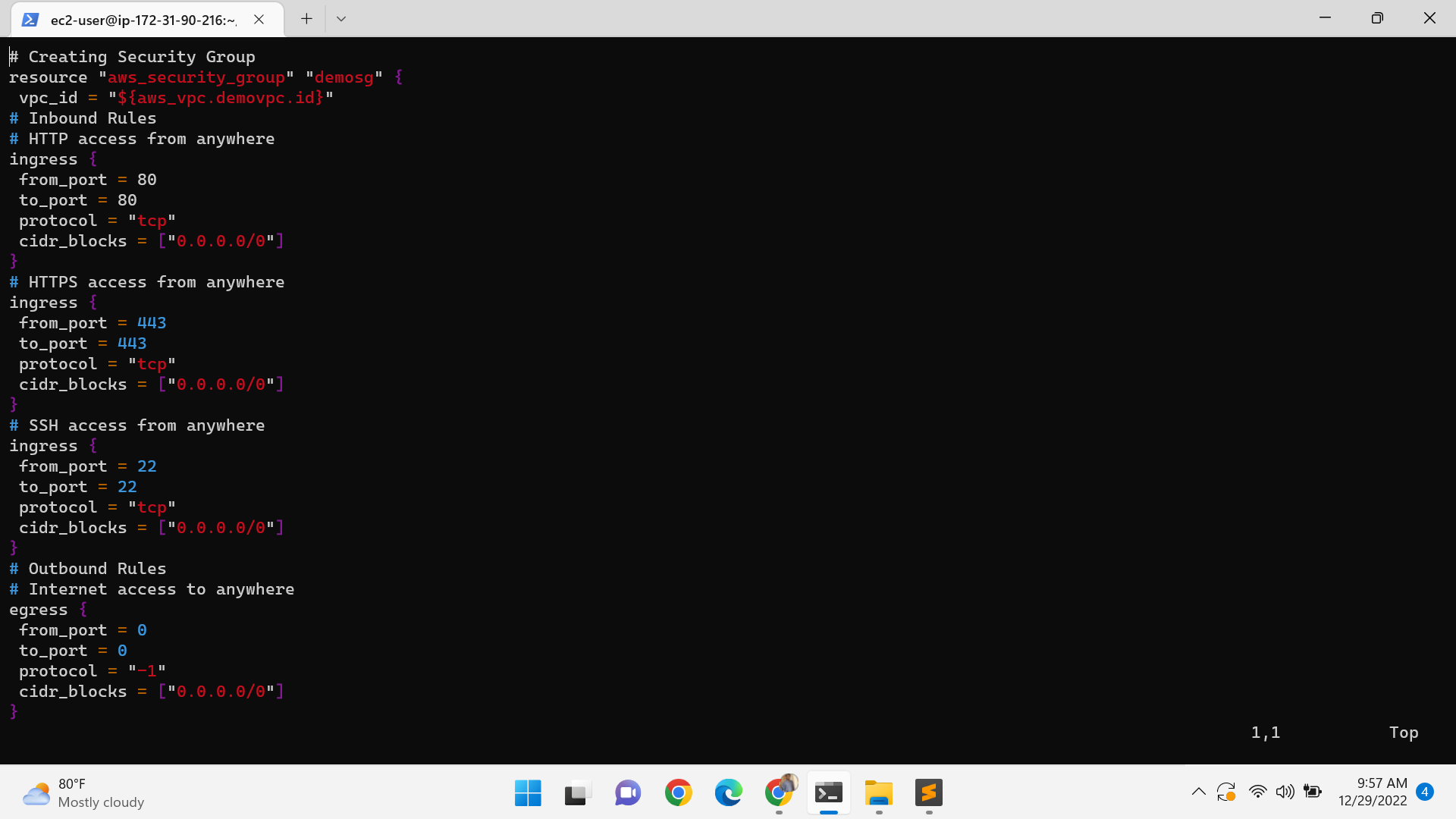


Initilaze the terraform and apply whether the script is write or wrong

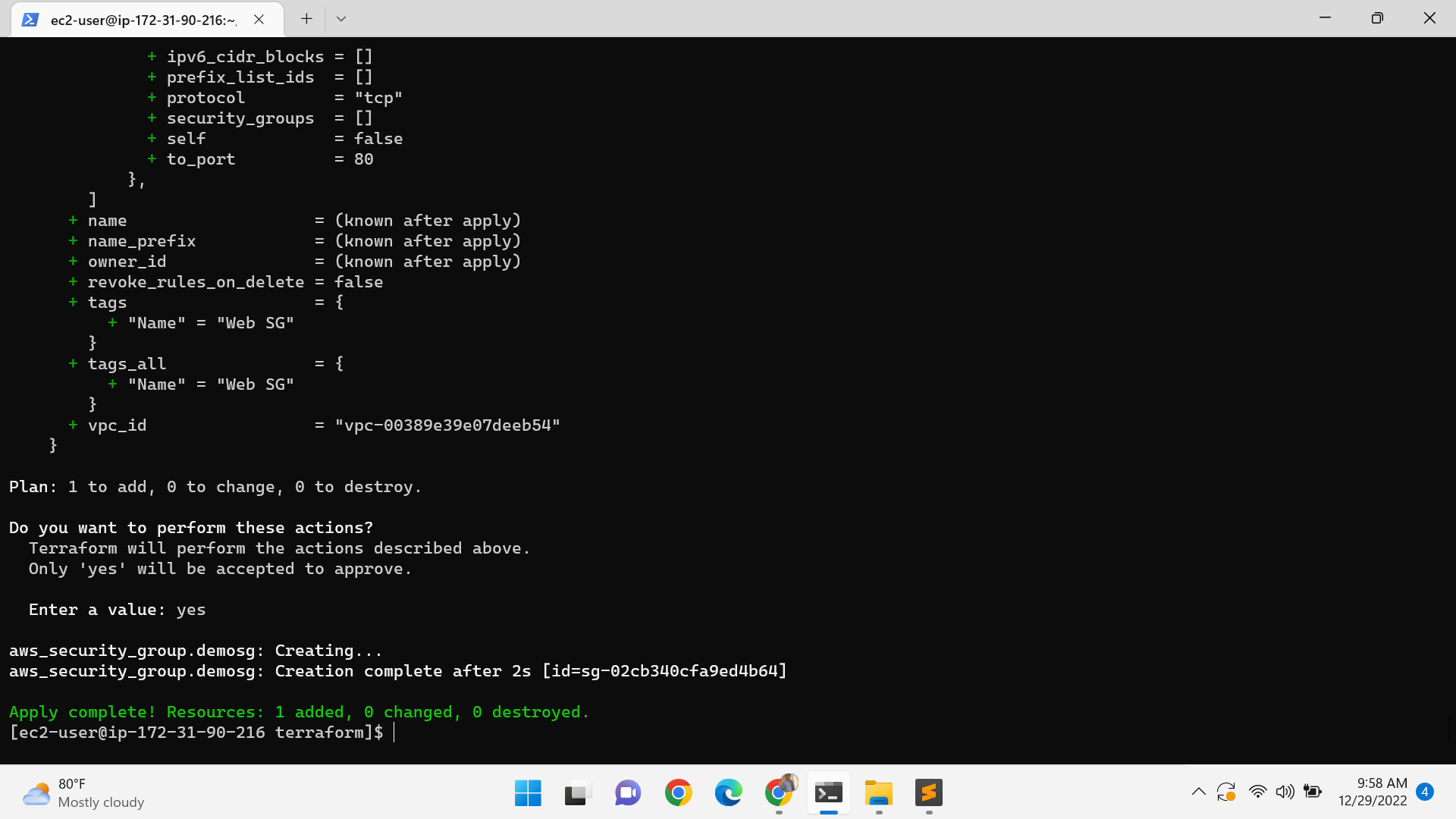
And it creates the route tables and associate with subnets and the route is connected to internet gateway



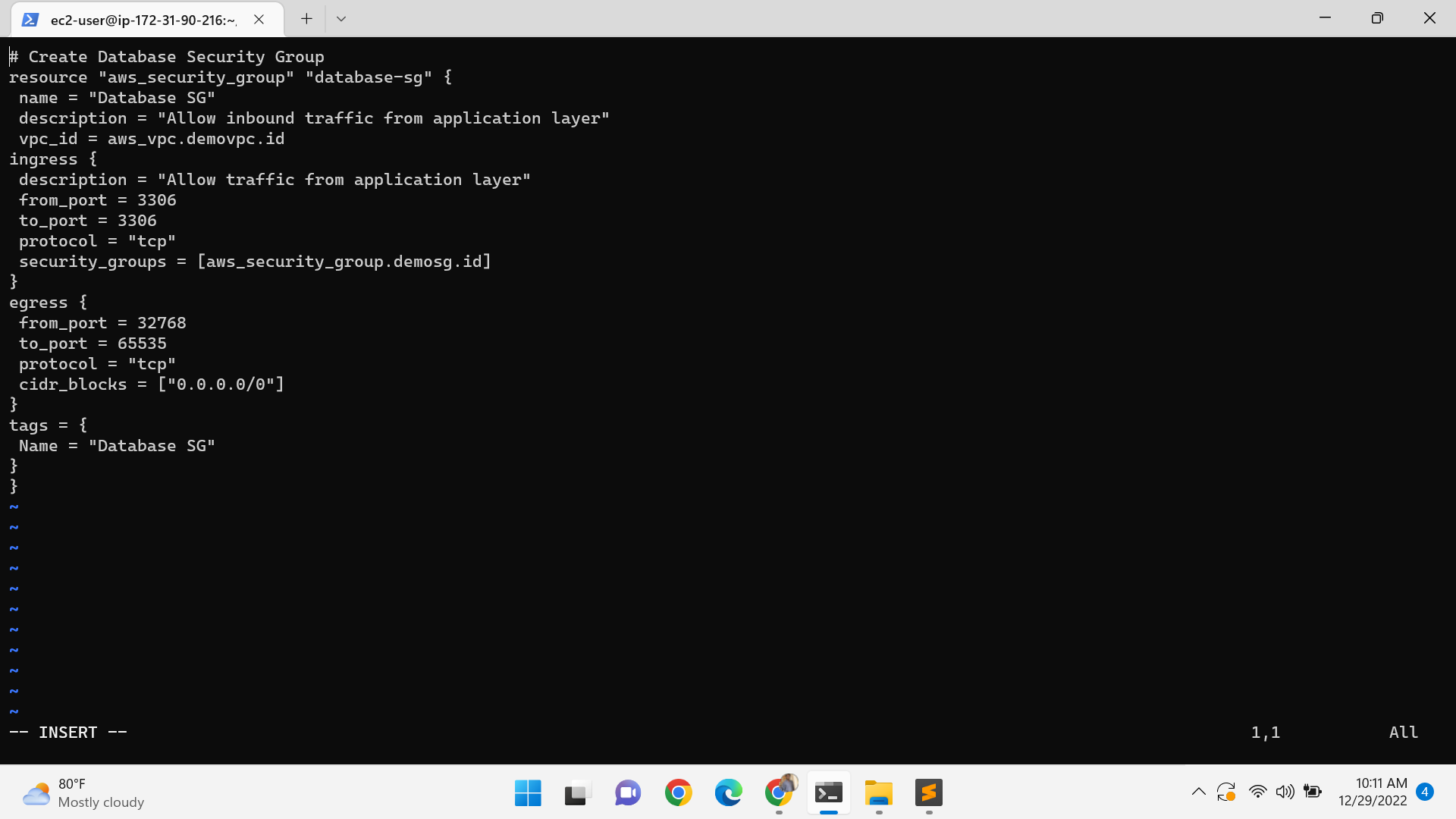
Create a security groups –vi web\_sg.tf

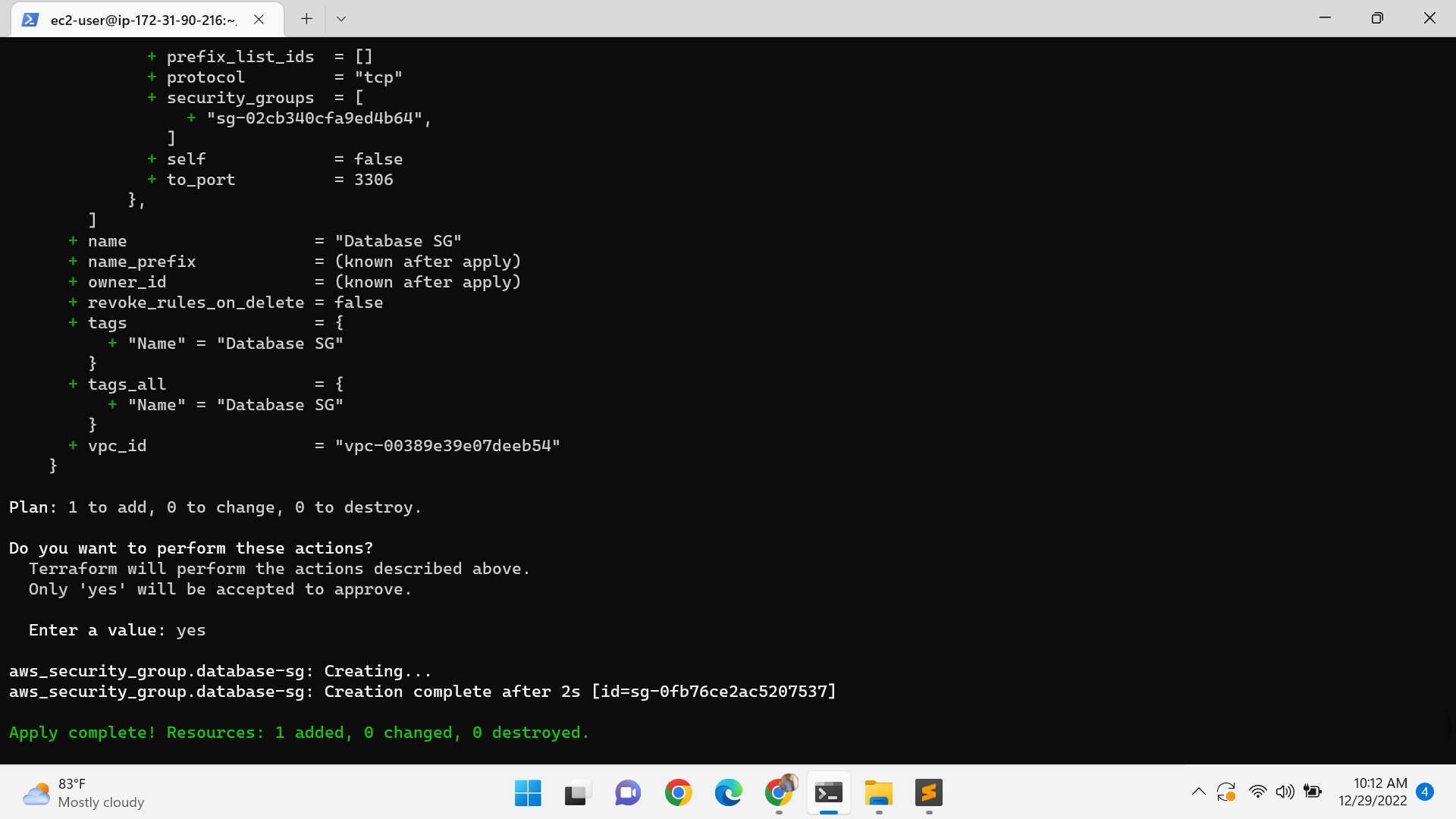


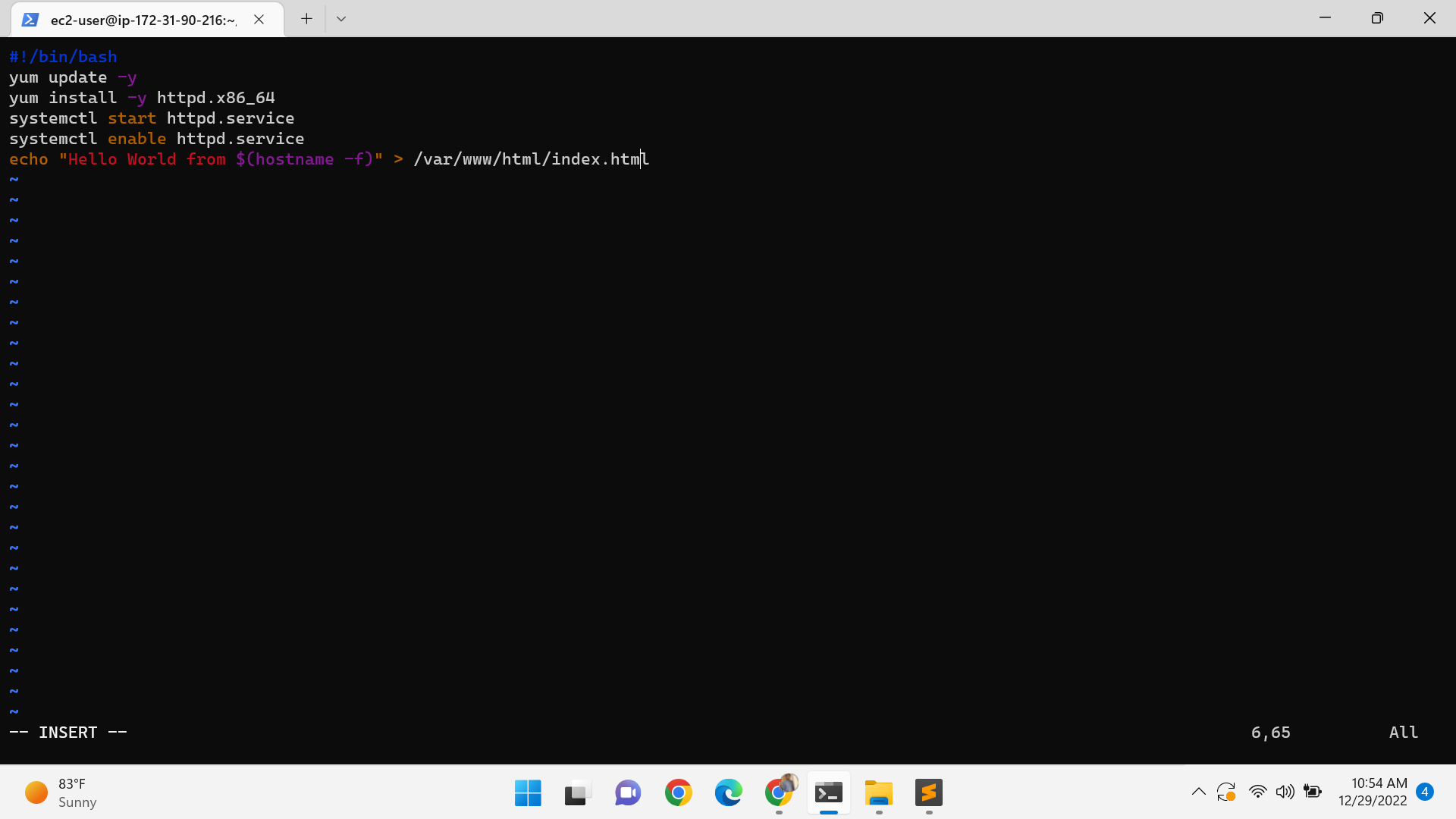
Terraform init and apply the terraform

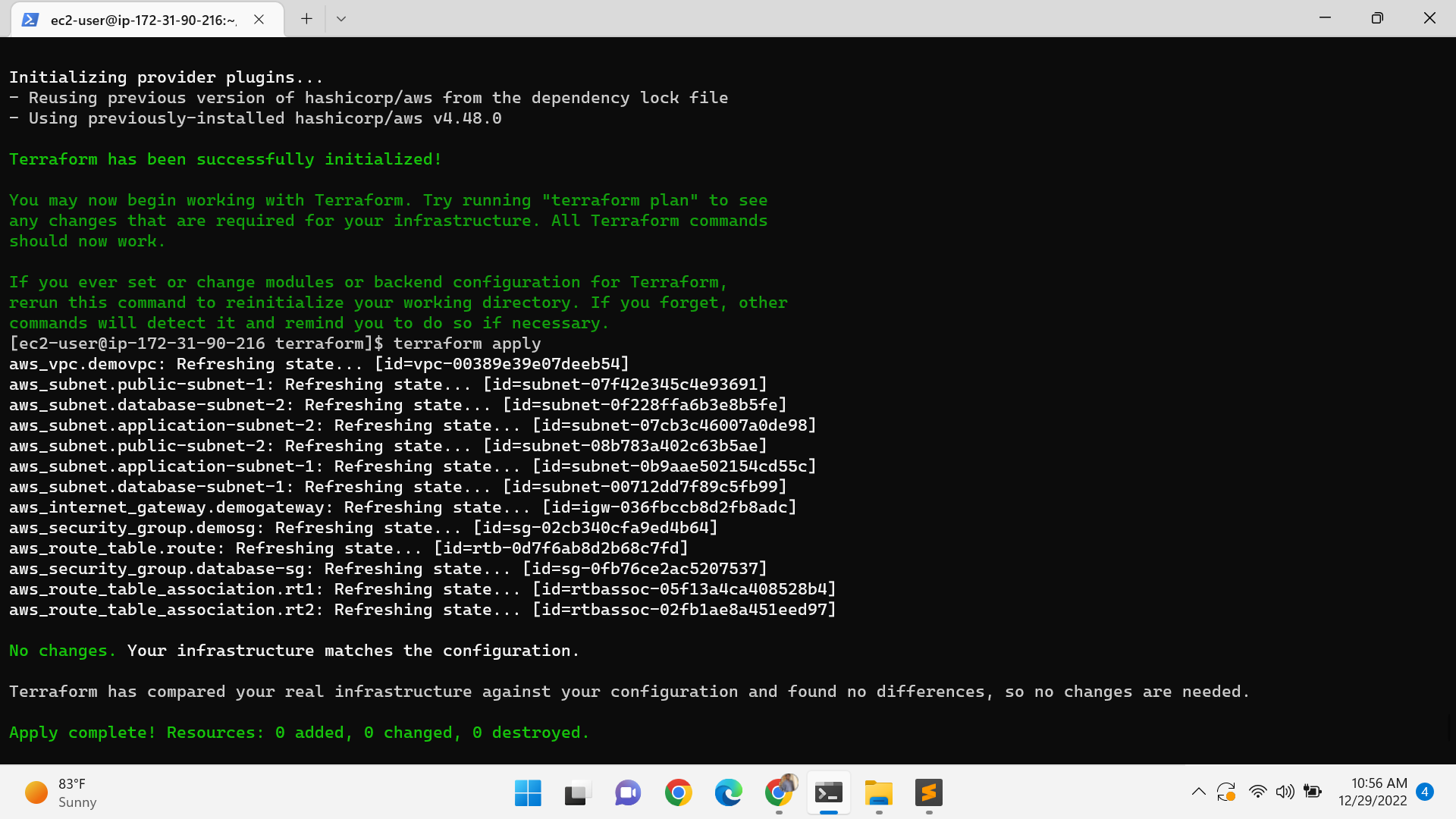


And again create a security group vi database\_sg.tf

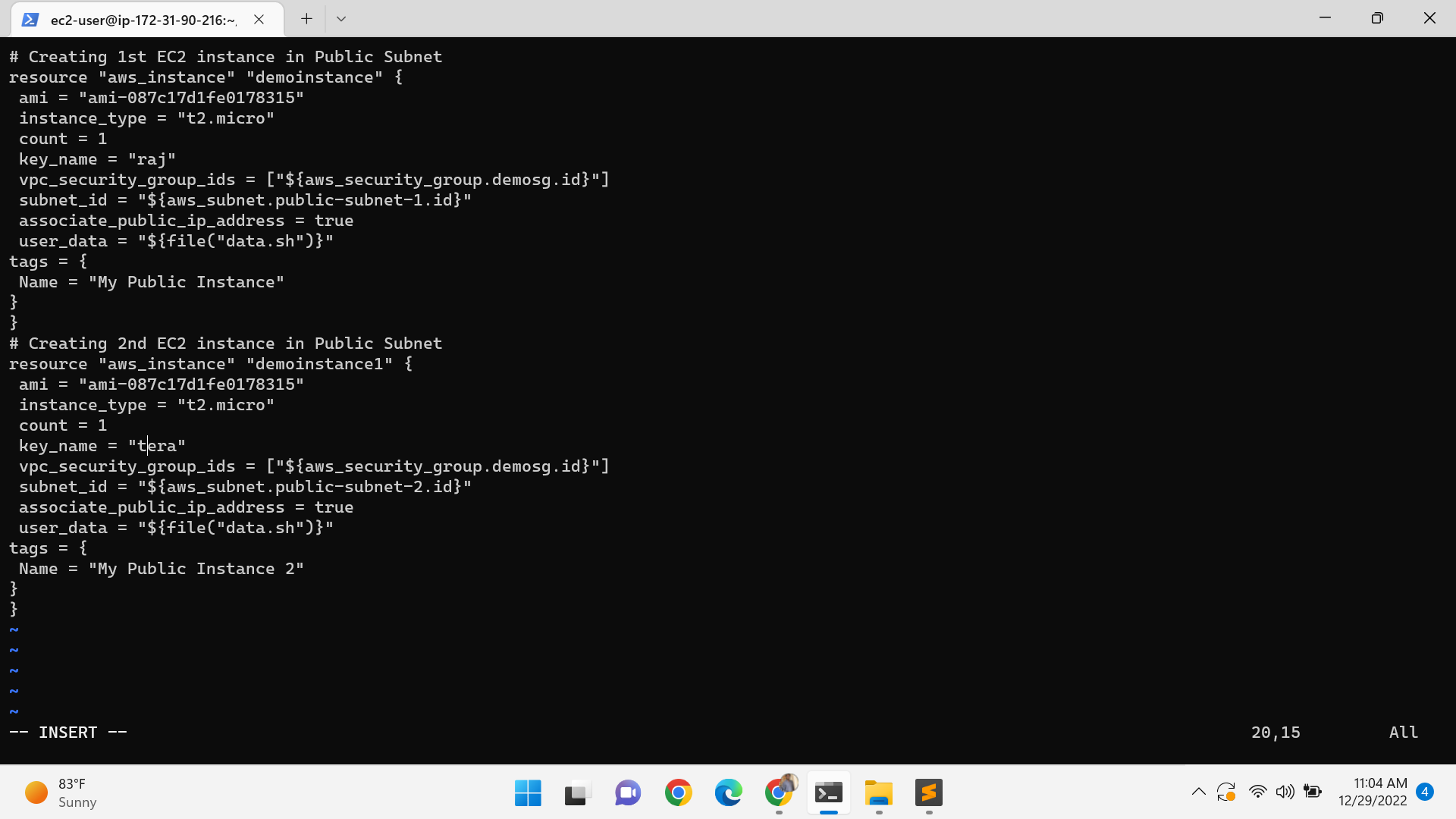
Terraform init and apply

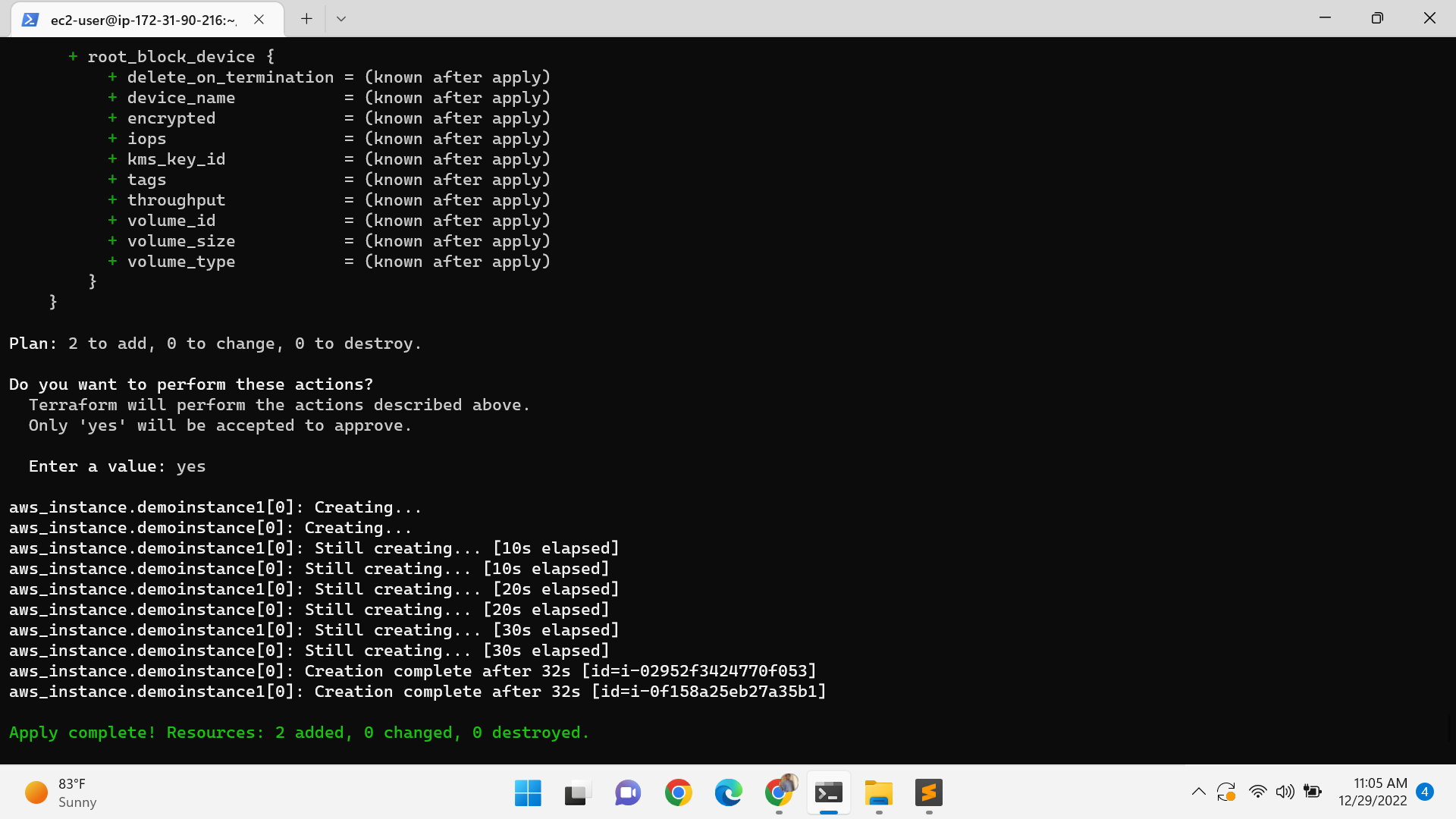
Create data.sh file-vi data.sh



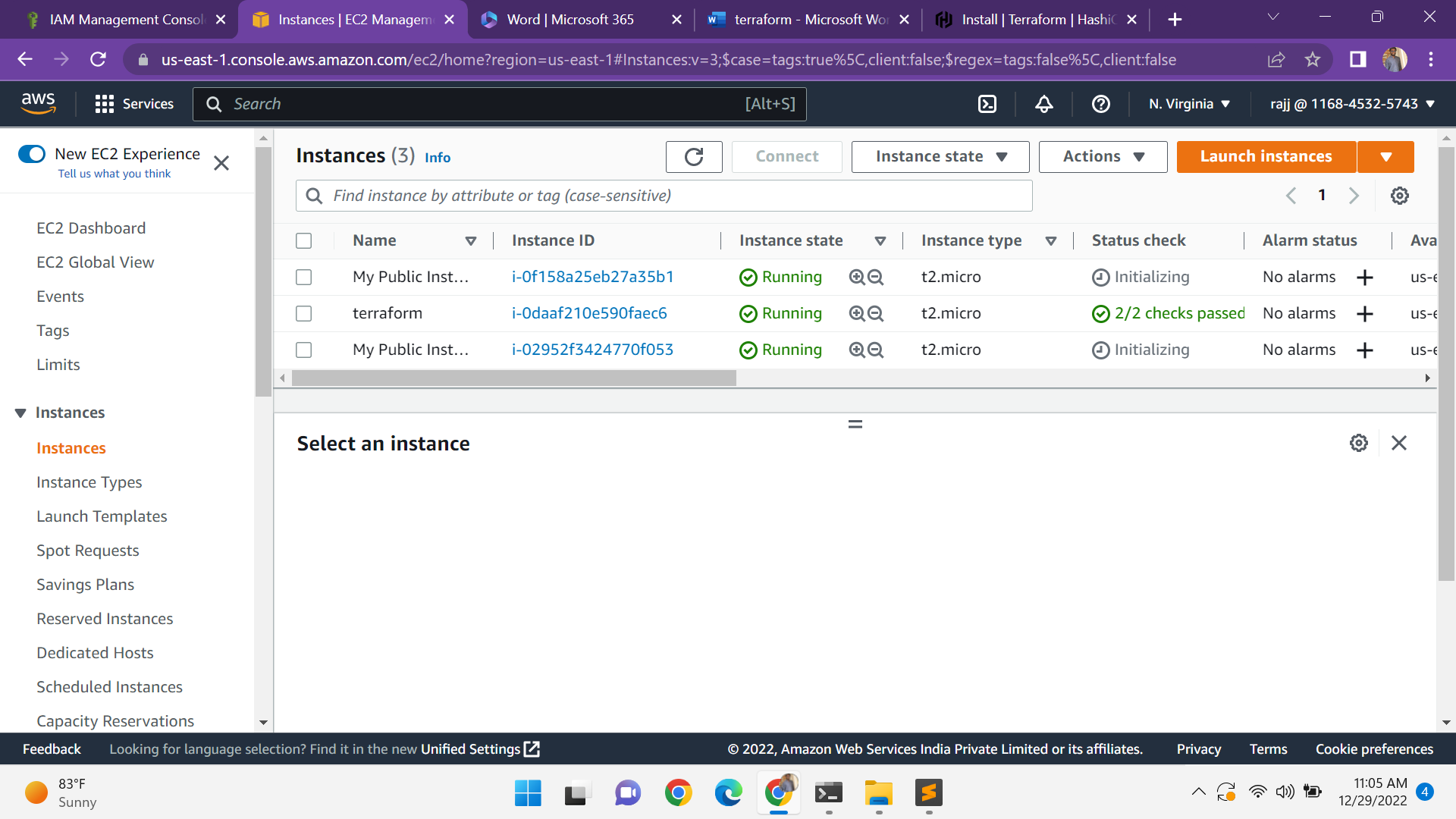


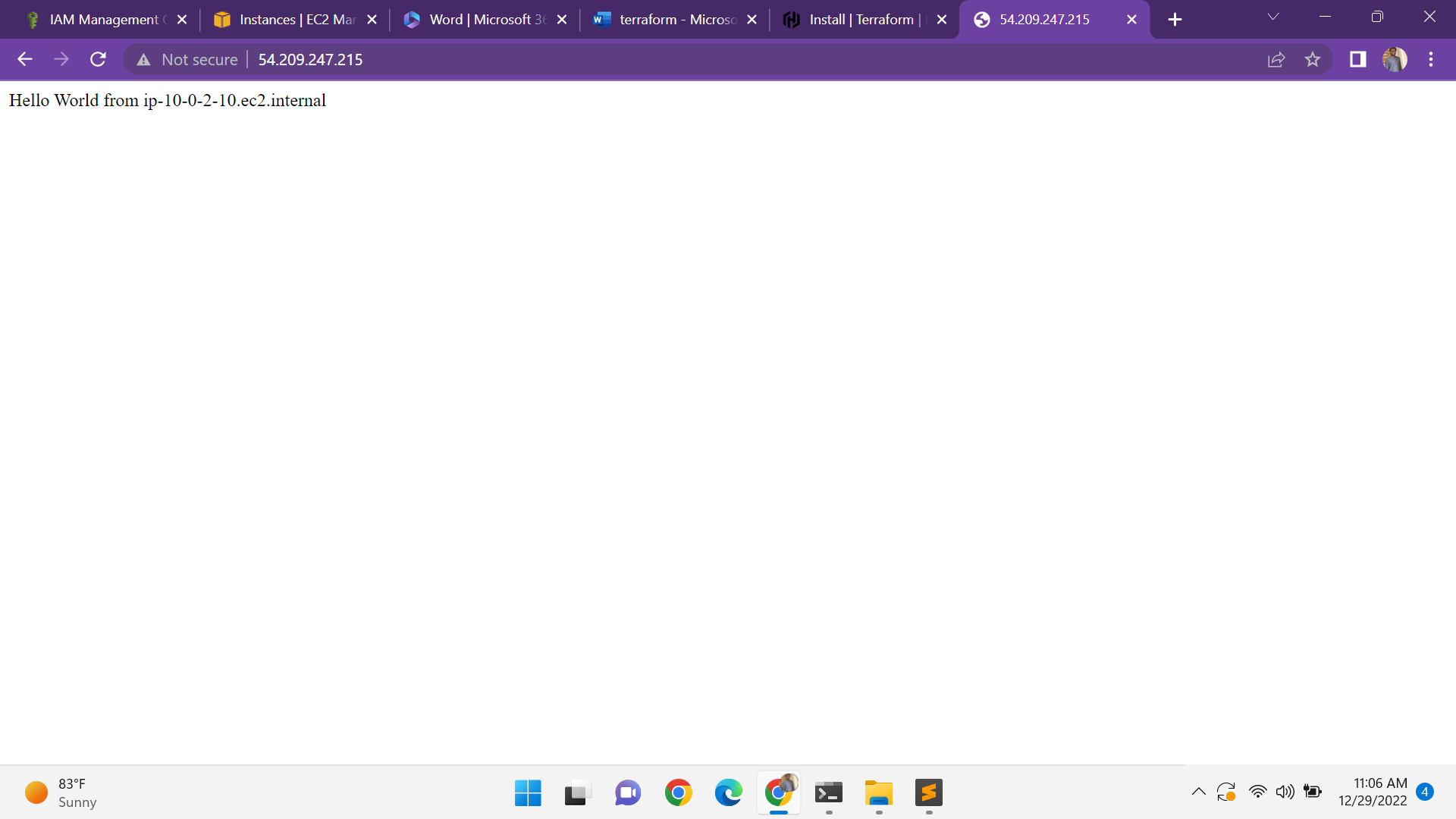
And then create the ec2 instance and the give the user data in this instance-vi ec2.tf

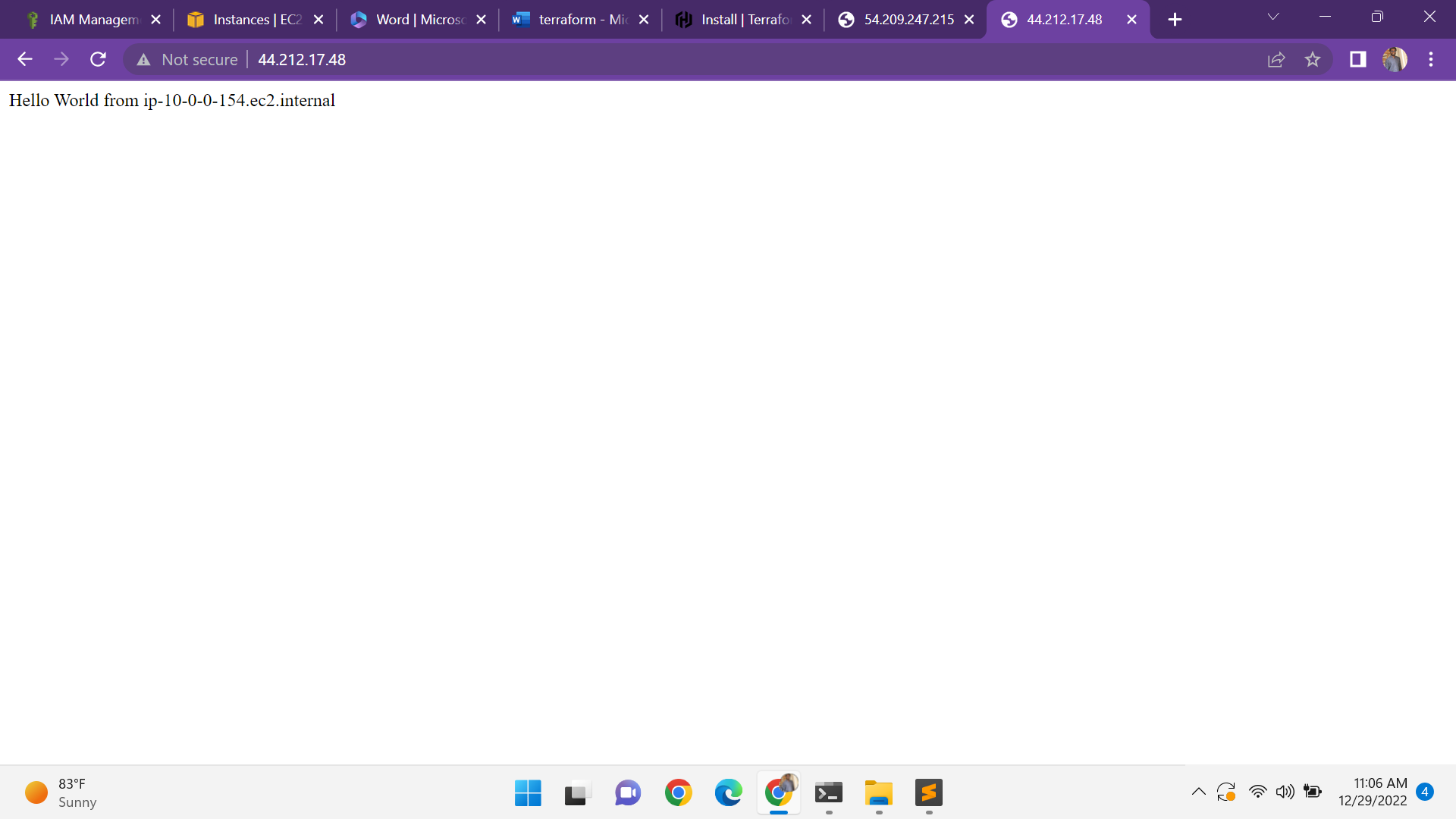




Go to the ec2 instances and copy the public ip adress and paste it on browser.

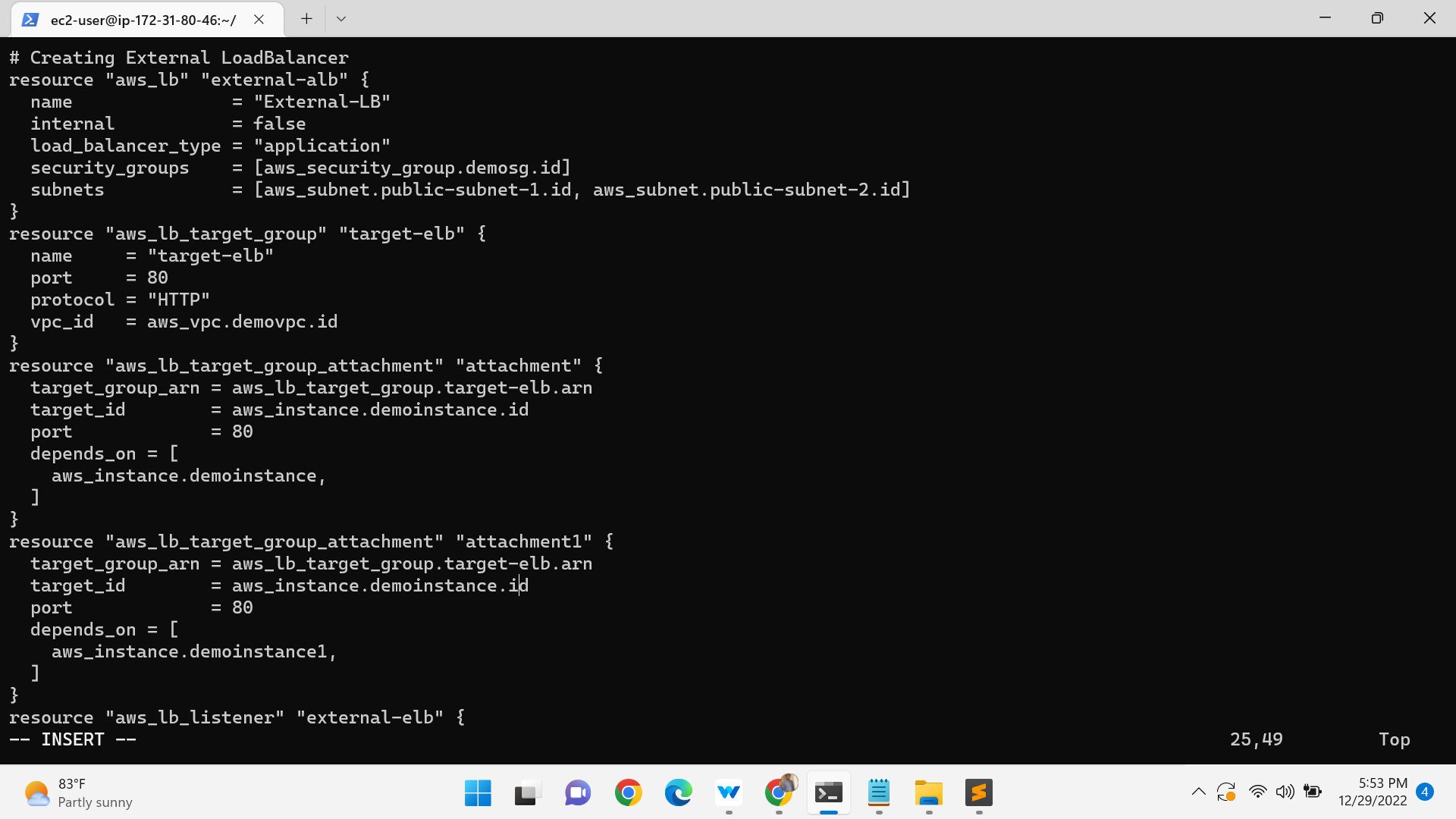




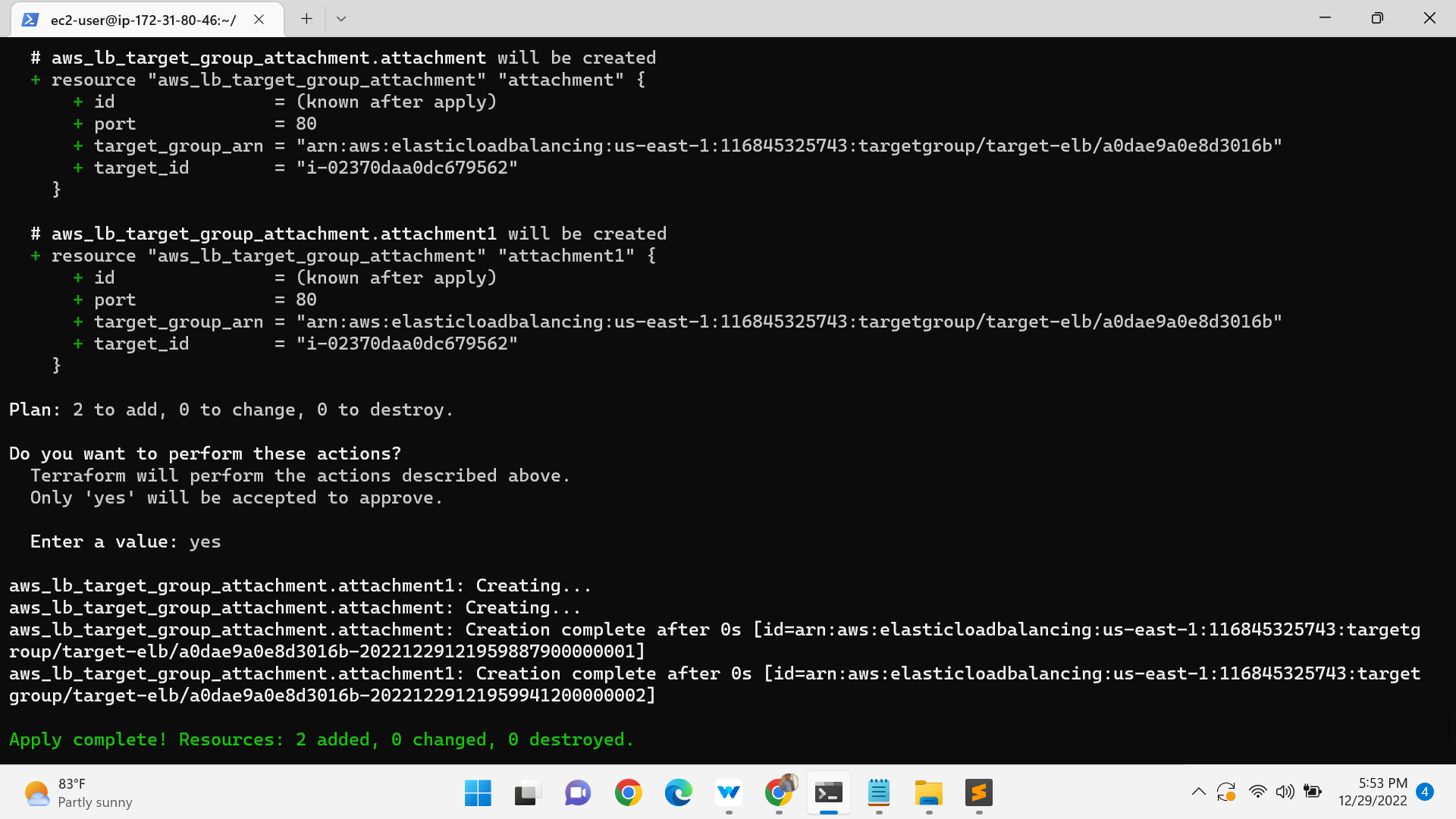


The above load balancer is of type external  Load balancer type is set to application  The aws\_lb\_target\_group\_attachment resource will attach our instances to the Target Group.  The load balancer will listen requests on port 80

Vi alb.tf

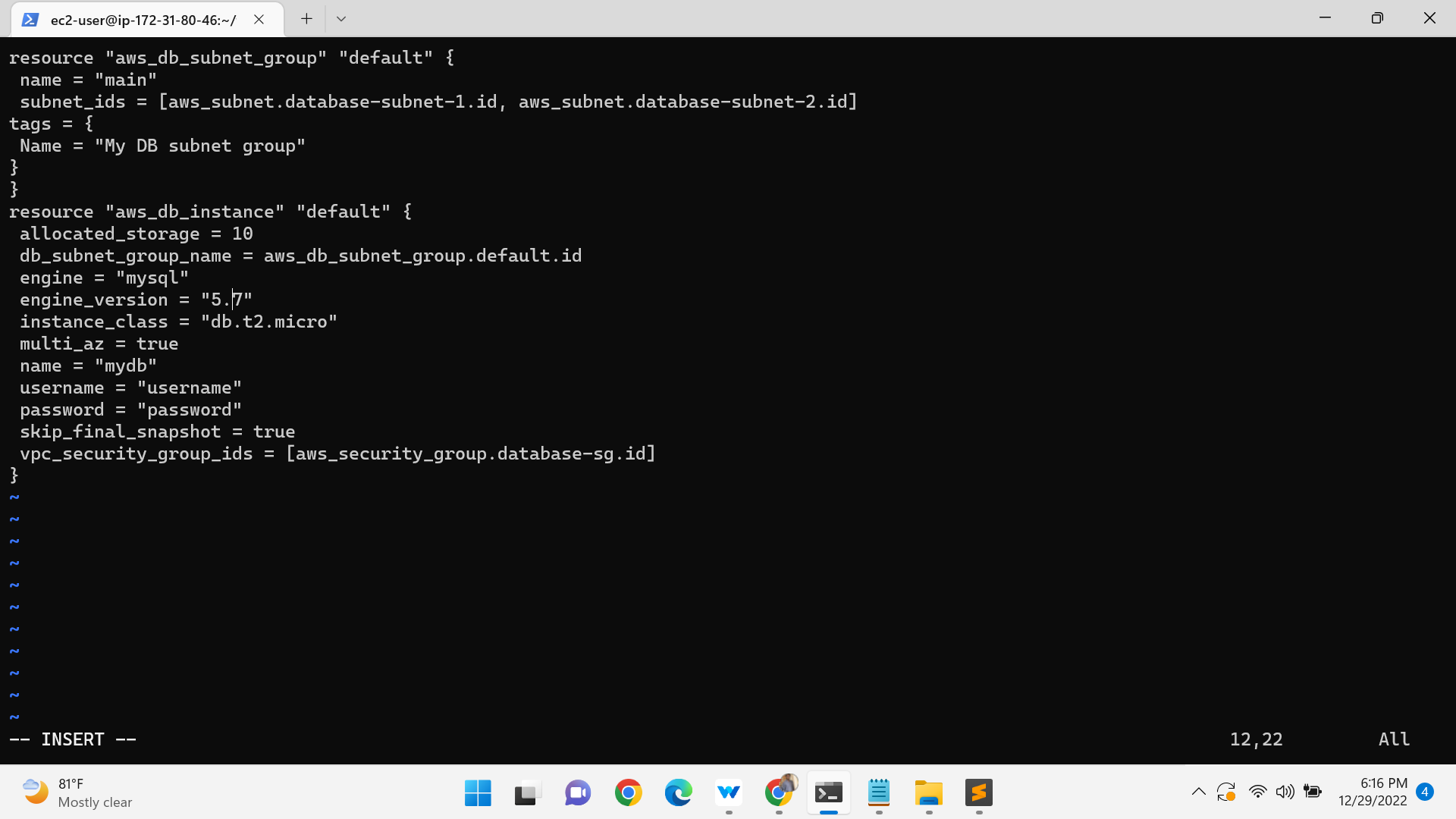


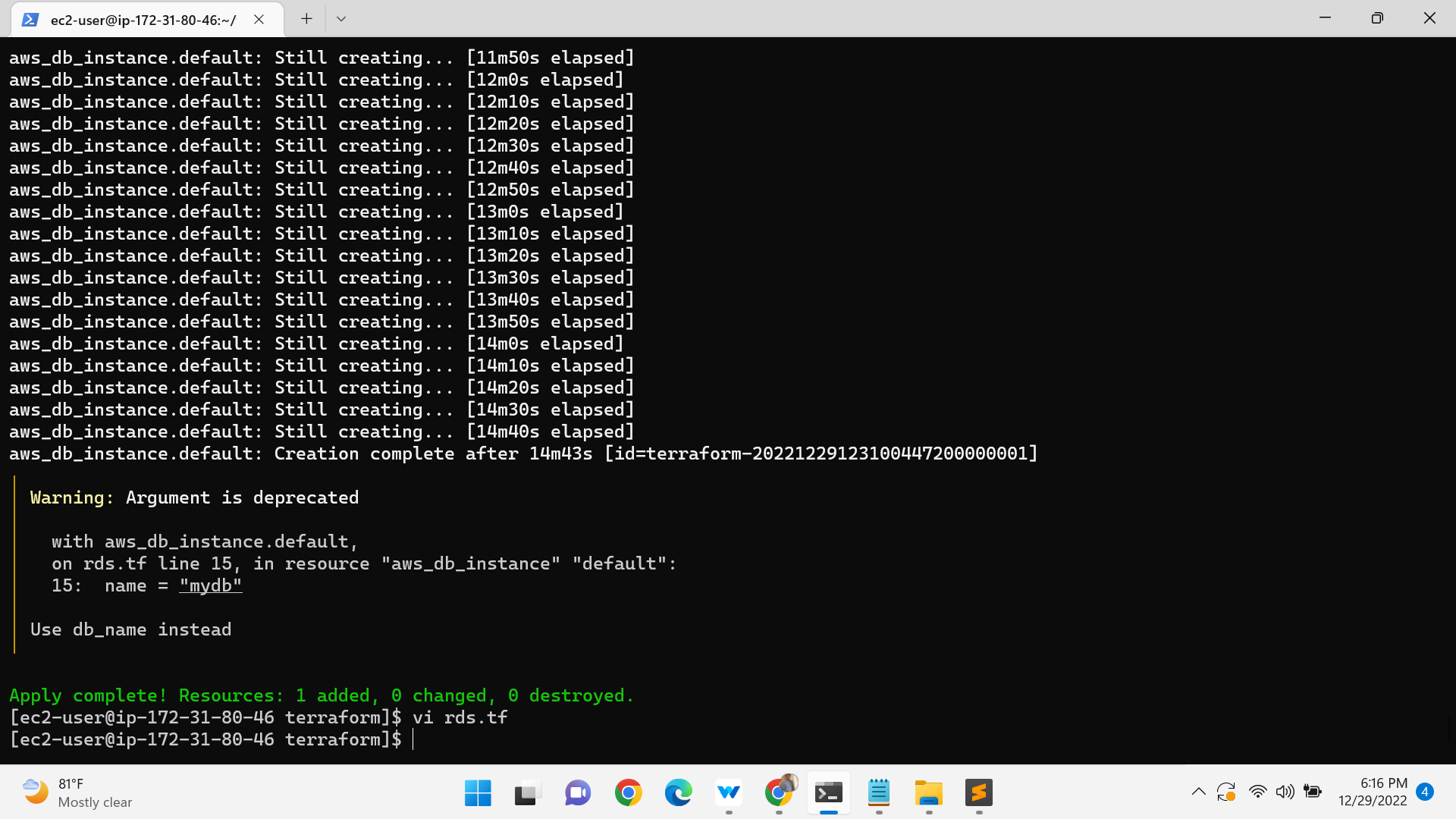
Terraform apply



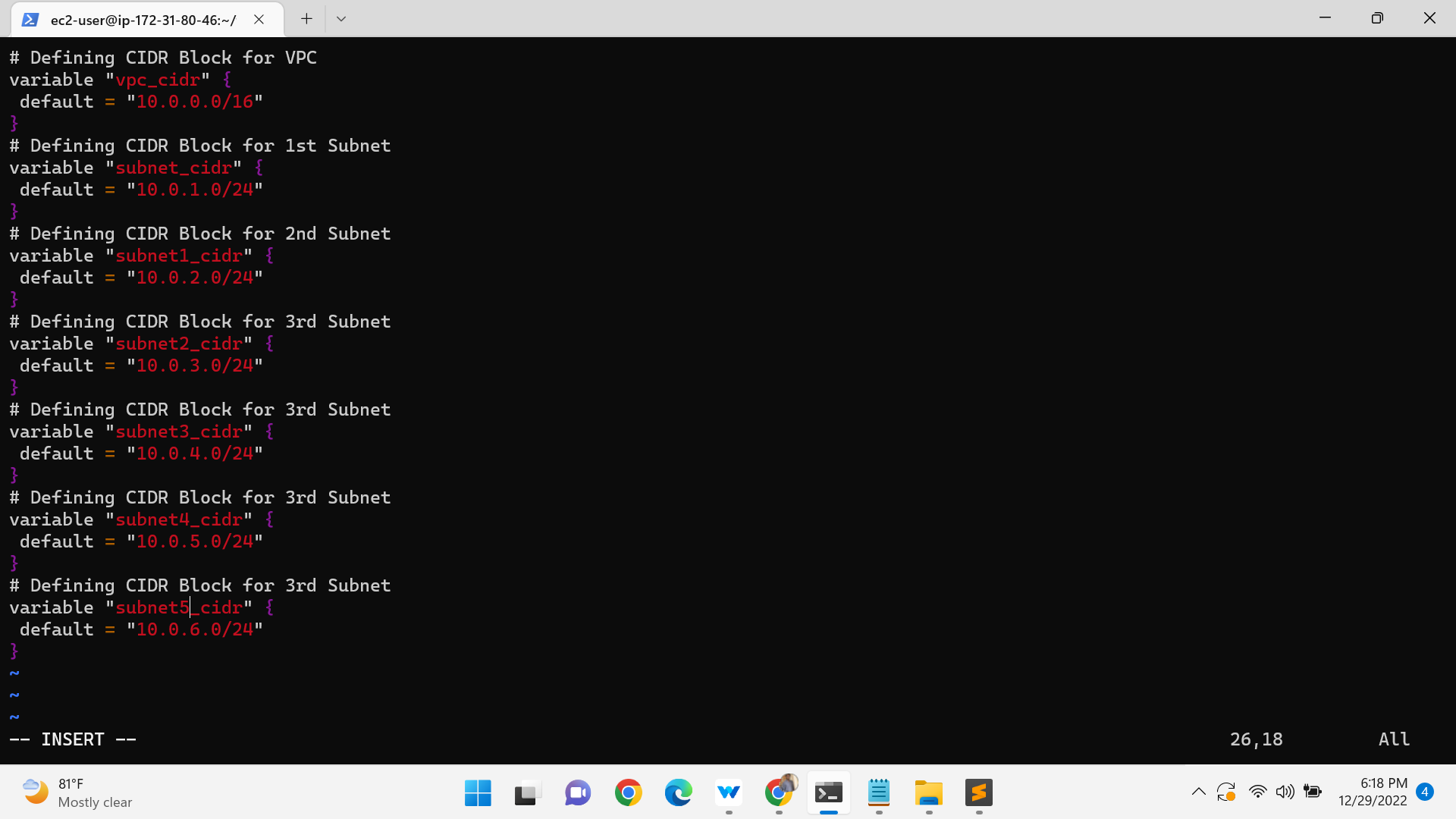
Create rds.tf and write the following script

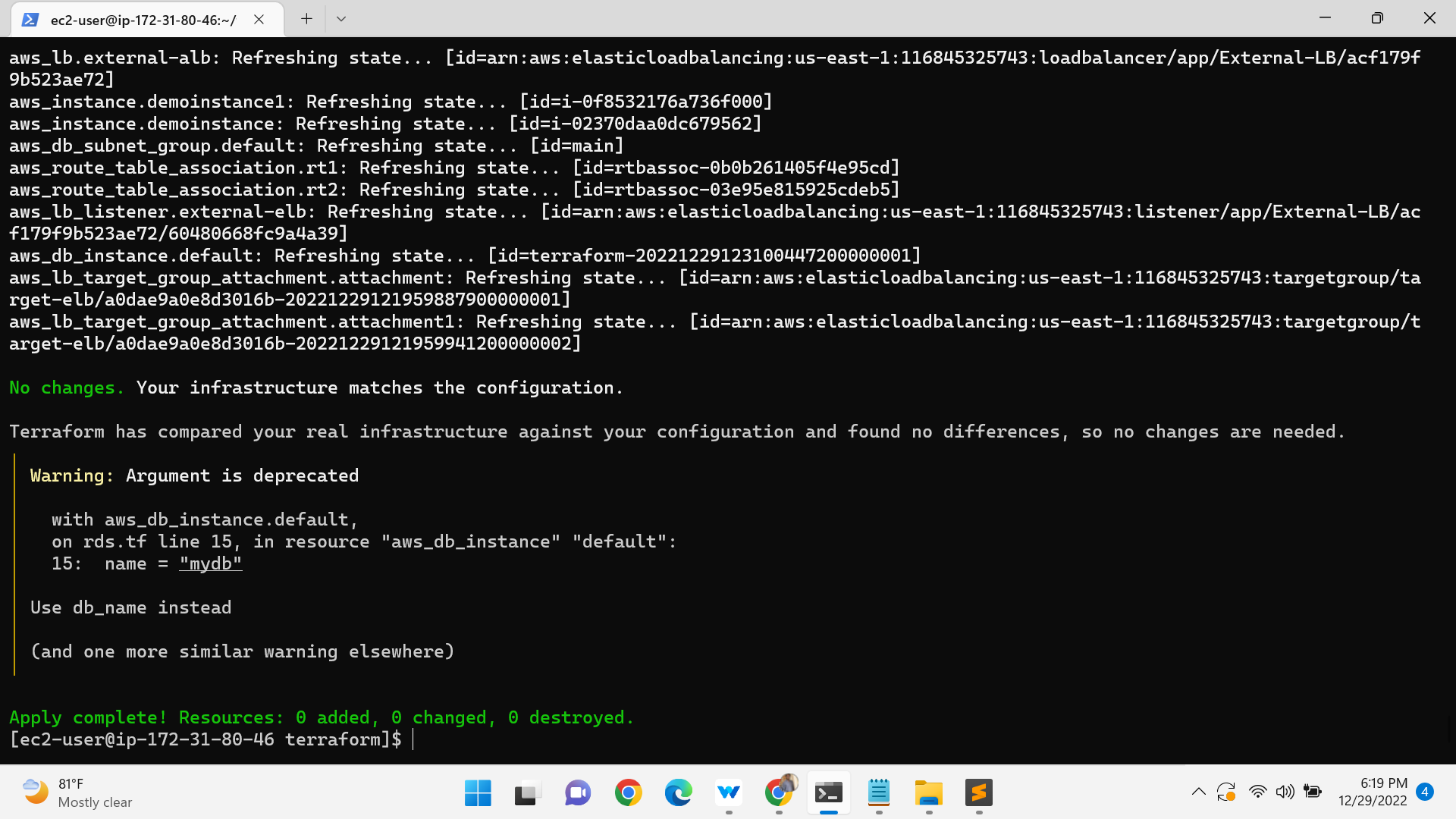
multi-az is set to true for the high availability





And create the file vi variable.tf





We will get the DNS of the application load balancer.

And create vi output.tf

