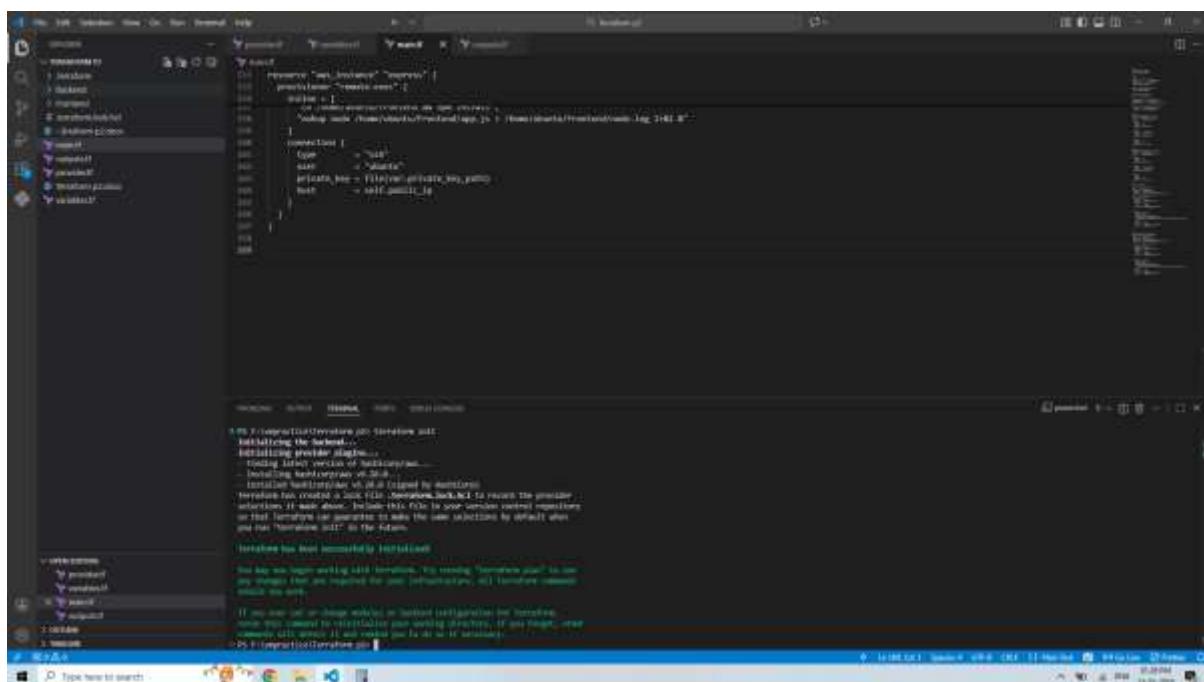


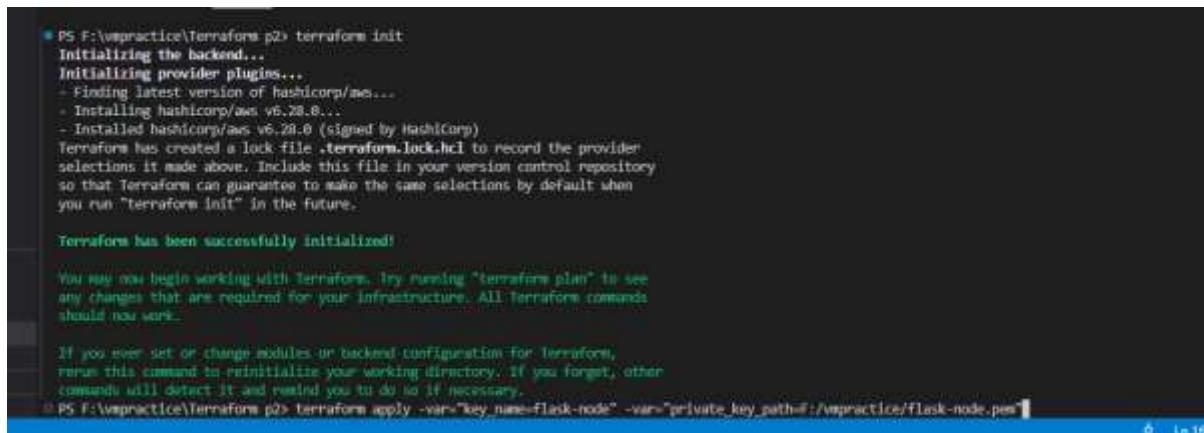
Terraform part 2



```
PS C:\Users\user\Documents\Terraform\aws> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.28.8...
- Installed hashicorp/aws v6.28.8 (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!
```

Initialize terraform working directory/folder



```
PS F:\mpRACTICE\Terraform p2> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.28.8...
- Installed hashicorp/aws v6.28.8 (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,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS F:\mpRACTICE\Terraform p2> terraform apply -var="key_name=flask-node" -var="private_key_path=F:\mpRACTICE\flask-node.pem"
```

Apply or run terraform it create and configure EC2,subnet,VPC,Inbound rules ,outbound rules, those specified in main.tf

EC2

Dashboard

AWS Global View

Events

Instances

- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations
- Capacity Manager

Images

- AMIs
- AMI Catalog

Elastic Block Store

- Volumes
- Snapshots
- Gateway Manager

Network & Security

- Security Groups
- Elastic IPs
- Placement Group
- Key Pairs
- Network Interfaces

Load Balancing

- Load Balancers

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region.

Instances (running)	2	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IP	0	Instances	2
Key pairs	2	Load Balances	0	Placement group	0
Security groups	5	Snapshots	0	Reserves	2

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance **Migrate a server**

Note: Your instances will launch in the Asia Pacific (Mumbai) Region.

Instance alarms

0 in alarm 0 DE 0 insufficient data

[View in CloudWatch Metrics](#)

[Instances in alarm](#)

Scheduled events

Asia Pacific (Mumbai)
No scheduled events

Migrate a server

Use AWS Application Migration Service to simplify and expedite migration from

EC2 cost

Date range: Costs in year: Credits remaining: \$116.78

Cost: \$0

Service health

[AWS Health Dashboard](#)

Region	Status
Asia Pacific (Mumbai)	The service is operating normally.

Zones

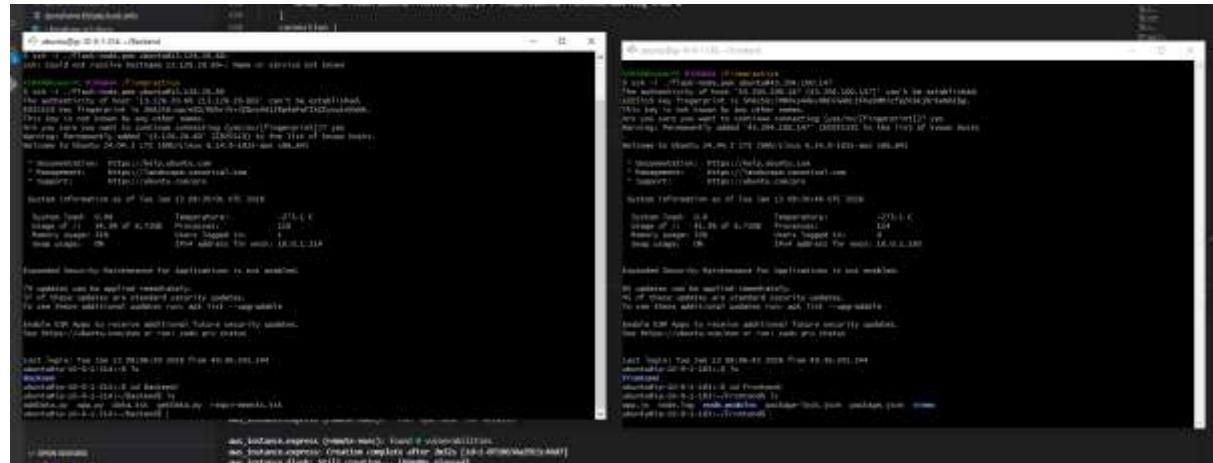
Zone name	Zone ID
ap-south-1a	ap-s1-aZ1
ap-south-1b	ap-s1-aZ2
ap-south-1c	ap-s1-aZ2

[Enable additional Zone](#) ap-south-1

Account

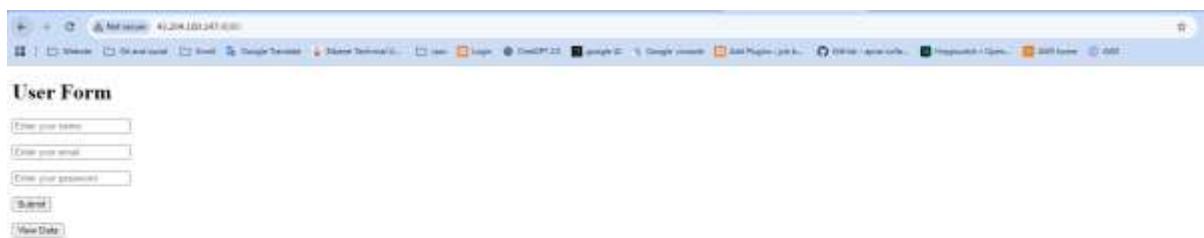
Default VPC: vpc-0cd10d4e

Two EC2 instance created and also started



Both instance configured and source code also added successfully by terraform we able to connect it with local using ssh

After completed create -launch – configure
in output we get public ip of both fronted and backend



Also able to connect with frontend service using public ip that it returns