SCENARIO:-
Mr. Boby is an IT Administrator of Alchemist group Pvt. Ltd. His organization is passionate about adopting IaaS using public cloud service providers. Their majority of clients are e-commerce and OTP service providers. Initially, they want to set up two virtual windows servers using Amazon EC2 which can be resizable and provide compute capacity along with a web-scale cloud computing solution.

Boby is planning to create IAAS as below for E-Commerce clients. You are required to provide the solution to Boby with proper step by step demonstration. Consider the following attached scenario and perform the following tasks using AWS EC2 Service:

- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your Amazon EC2 instance to scale
- Explore EC2 limits
- Test termination protection
- Terminate your EC2 instance

AIM:- To automate above Infrastructure using Infrastructure As A Code - Terraform.

Steps:-

1. **Configuring terraform code.**
   - **CODE:- (main.tf)**

```
terraform {
 required_providers {
   aws = {
     source  = "hashicorp/aws"
     version = "~> 3.0"
   }
 }
 required_version = ">= 0.13.5"
}

# Configure the AWS Provider
provider "aws" {
 region     = "us-east-1"
 access_key = "ACCESS_KEY"
 secret_key = "SECRET_KEY"
}

# Create a Instance
resource "aws_instance" "vm" {
```

```
  ami                     = "ami-0cff7528ff583bf9a"
  instance_type           = "t2.micro"
  key_name                = "MyKey"
  security_groups         = ["${aws_security_group.allow_http.name}"]
  disable_api_termination = true
  monitoring              = true

  tags = {
    Name = "Amazon Linux 2"
  }
}

resource "aws_security_group" "allow_http" {
  name = "allow_http"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port        = 0
    to_port          = 0
    protocol         = "-1"
    cidr_blocks      = ["0.0.0.0/0"]
    ipv6_cidr_blocks = ["::/0"]
  }

  egress {
    from_port        = 0
    to_port          = 0
    protocol         = "-1"
    cidr_blocks      = ["0.0.0.0/0"]
    ipv6_cidr_blocks = ["::/0"]
  }
}
```

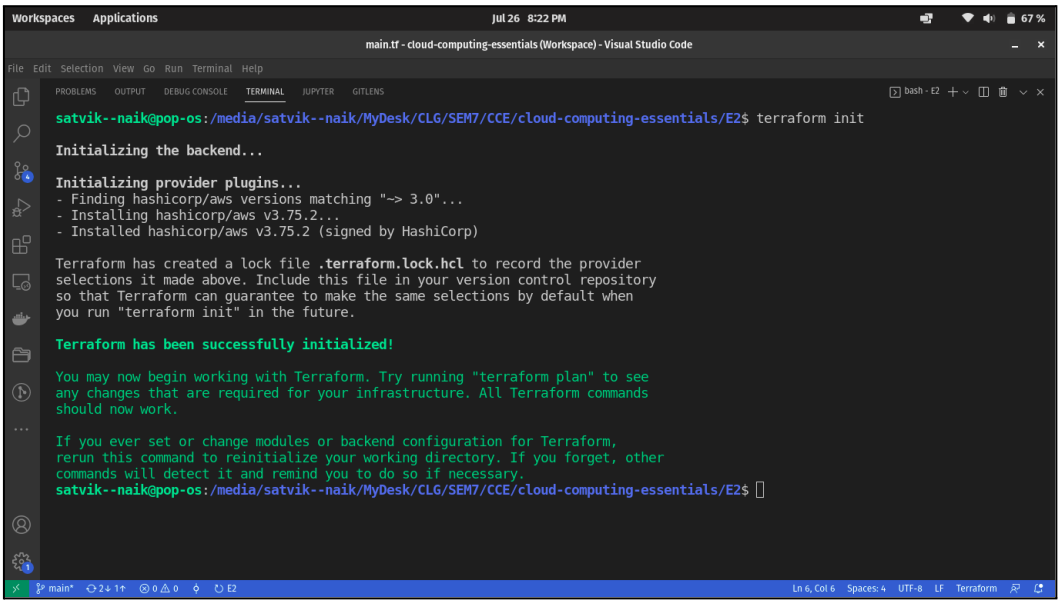
- ***disable_api_termination*** If true, enables EC2 Instance Termination Protection.

- ***monitoring*** If true, the launched EC2 instance will have detailed monitoring enabled.

2. Create an instance via terraform.

Terraform has 5 main commands, which are as follows:

- **terraform init**
  - This prepares your working directory for other commands.

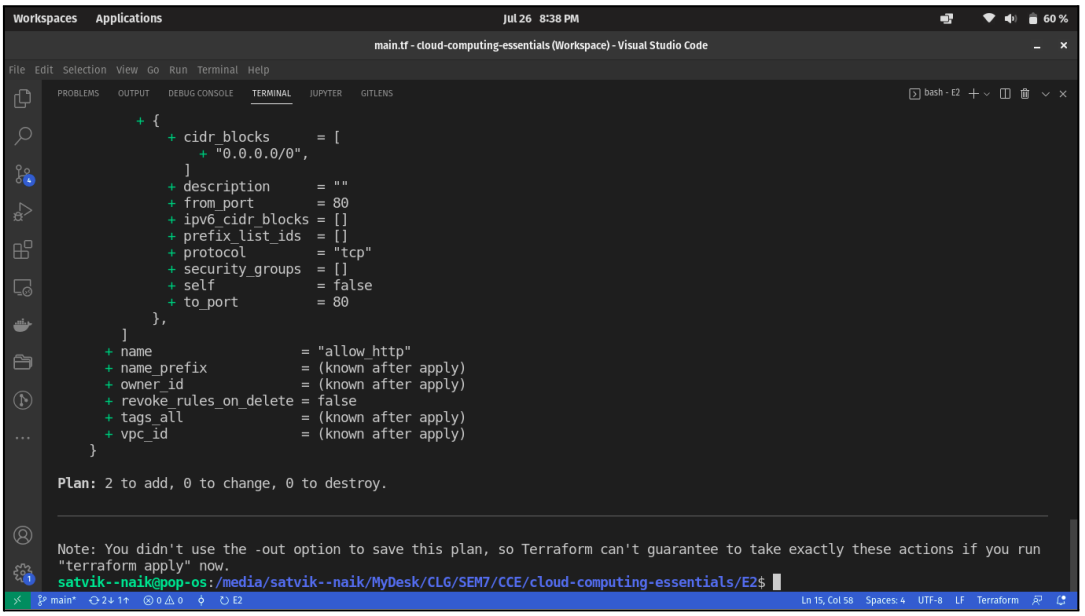


- **terraform validate**
  - Checks whether the configuration is valid or not.

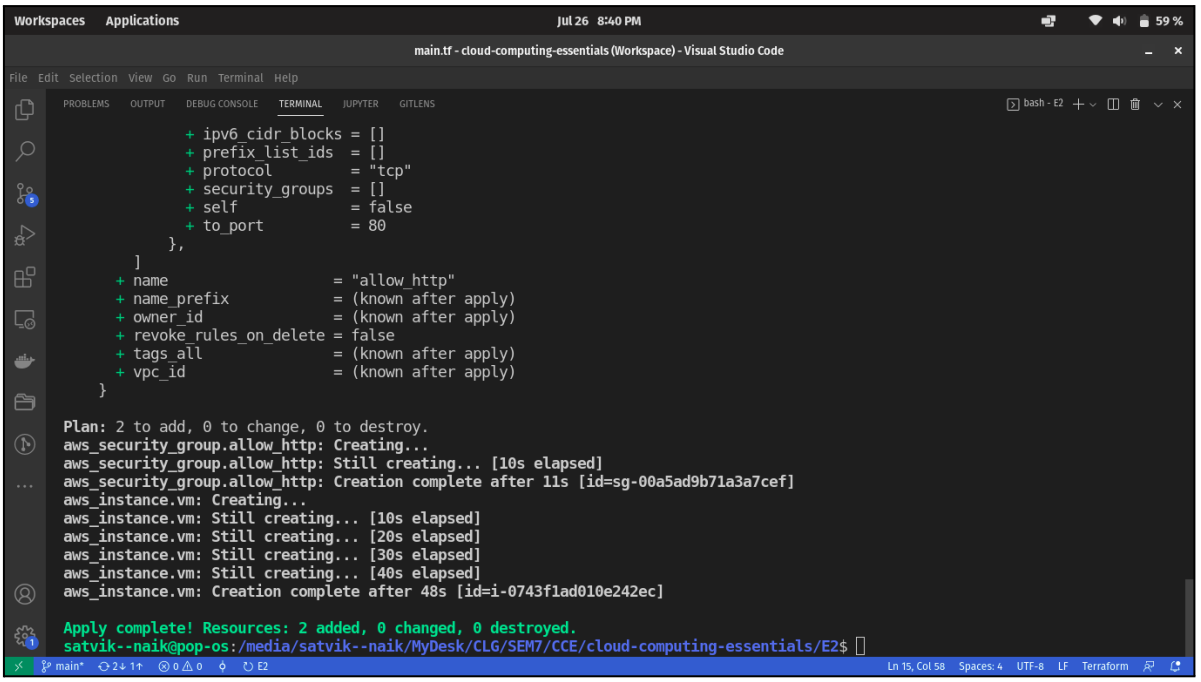
```
satvik--naik@pop-os:/media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E2$ terraform validate
Success! The configuration is valid.

satvik--naik@pop-os:/media/satvik--naik/MyDesk/CLG/SEM7/CCE/cloud-computing-essentials/E2$
```

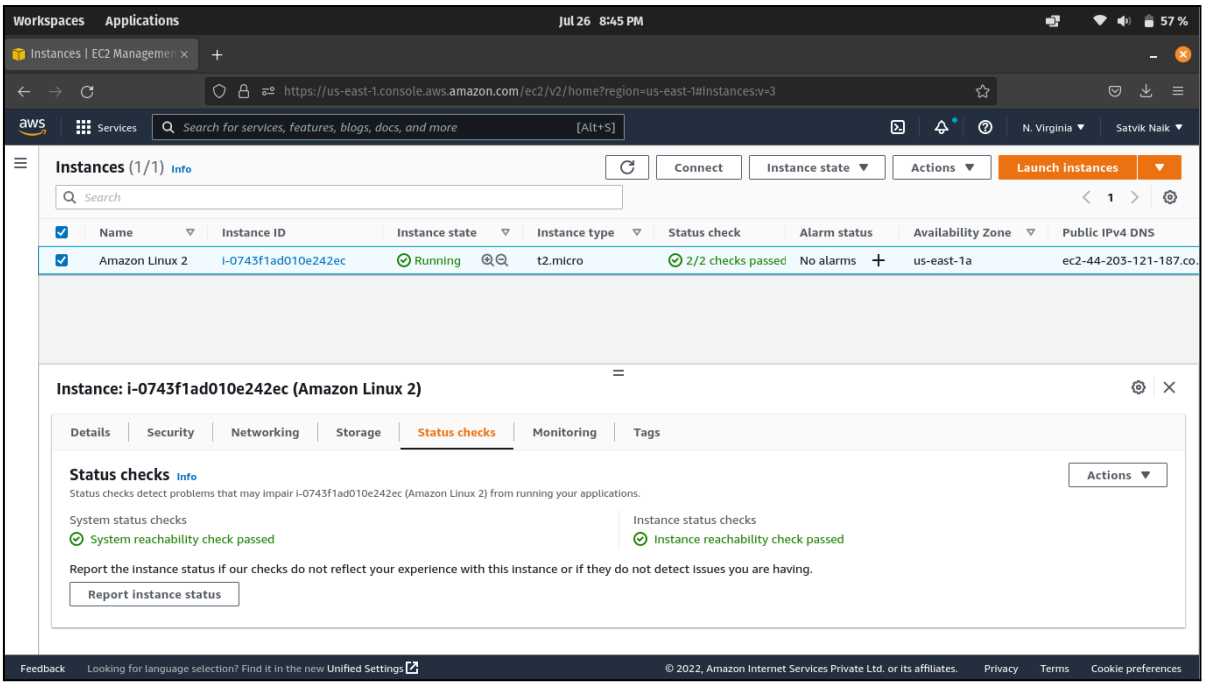
- **terraform plan**
  - It shows changes required by the current configuration.



- **terraform apply**
  - Creates or updates infrastructure.



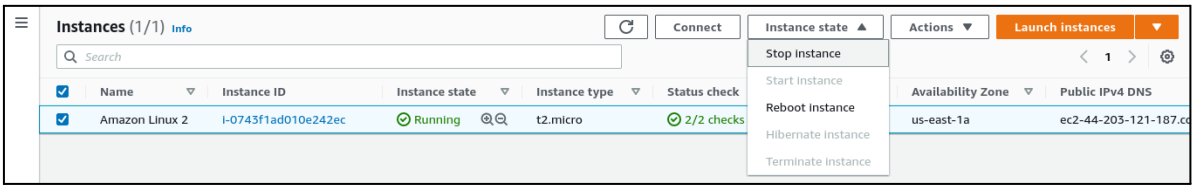
- After applying the code, the instance has been successfully ***created*** & is in a ***Running*** state.



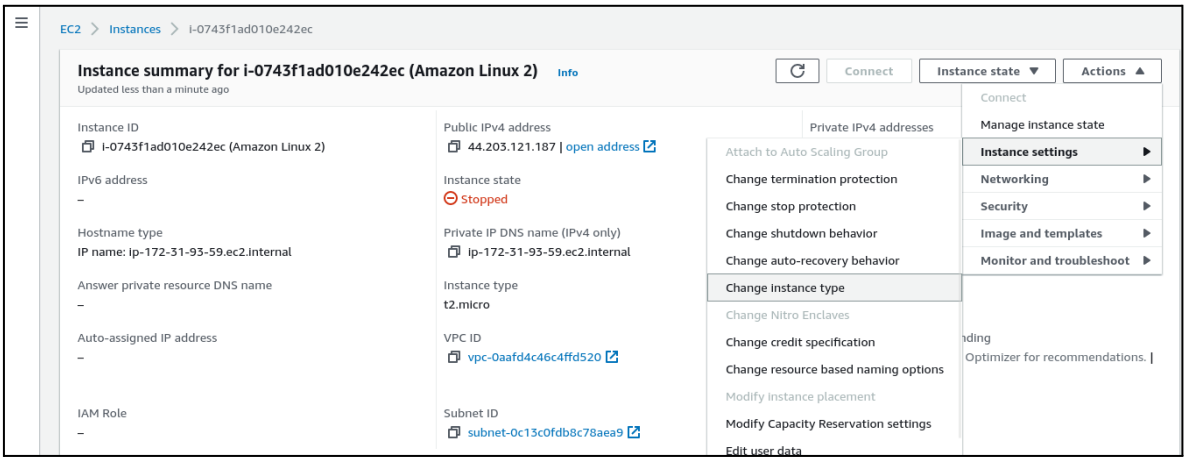
3. Resizing Amazon EC2 instance to scale.

- In order to resize amazon EC2 instance to scale, you must ***stop the existing instance*** and then you should ***change the instance type***.

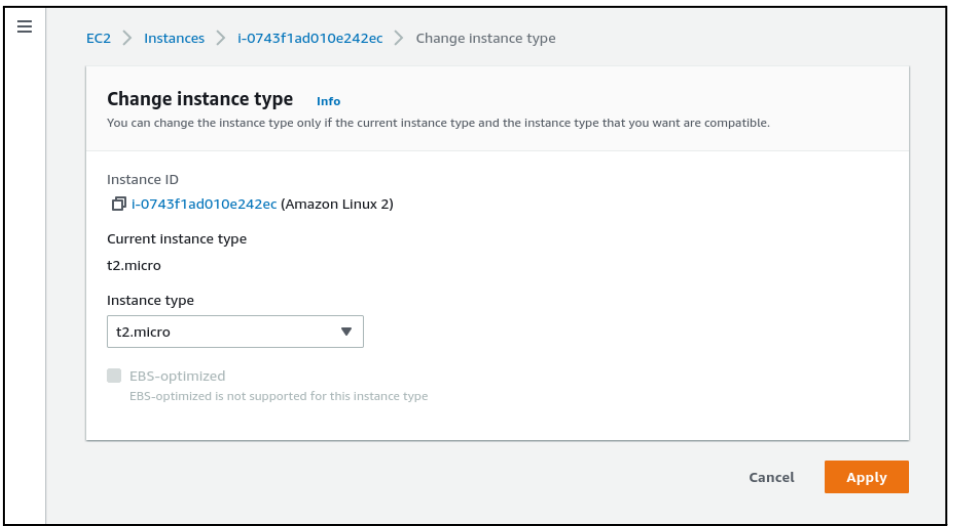
- Stop instance



- Change instance type



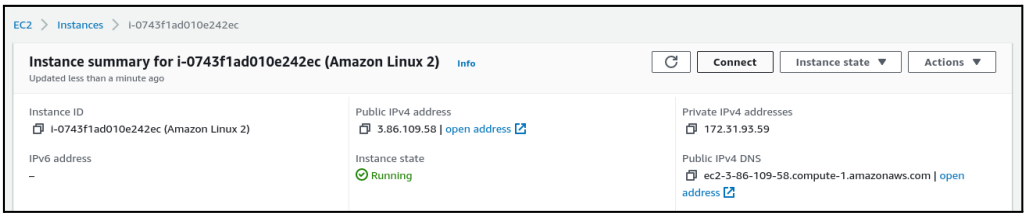
- By default t2.micro is selected, for scaling other instance types could be selected and further applied. (i.e. t2.xlarge, t2.2xlarge etc…)



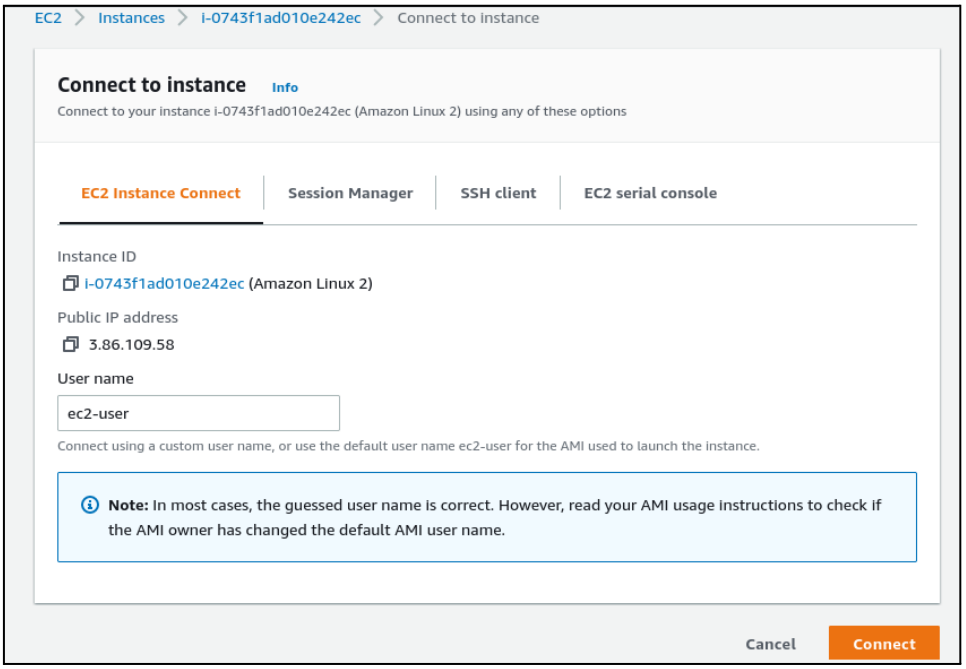
- Select the particular instance type and click on ***'Apply'*** & then start your instance.

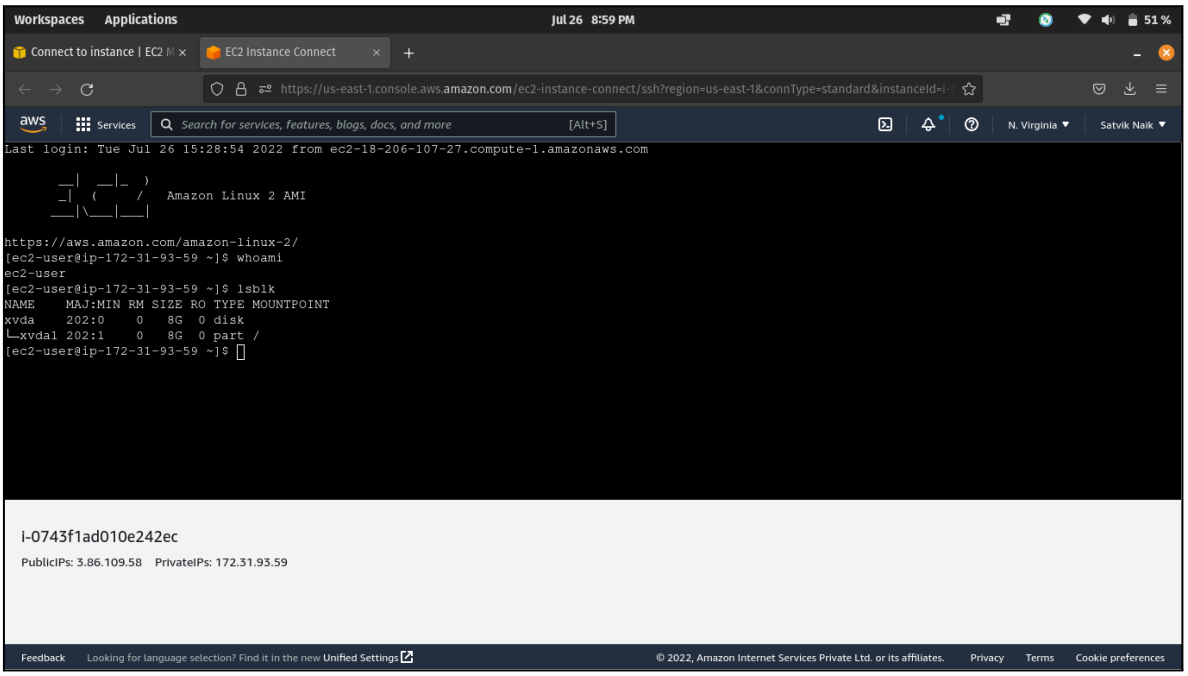
4. Connect to instance via EC2 instance Connect.

- Go to the instance and click on ***'Connect'***.



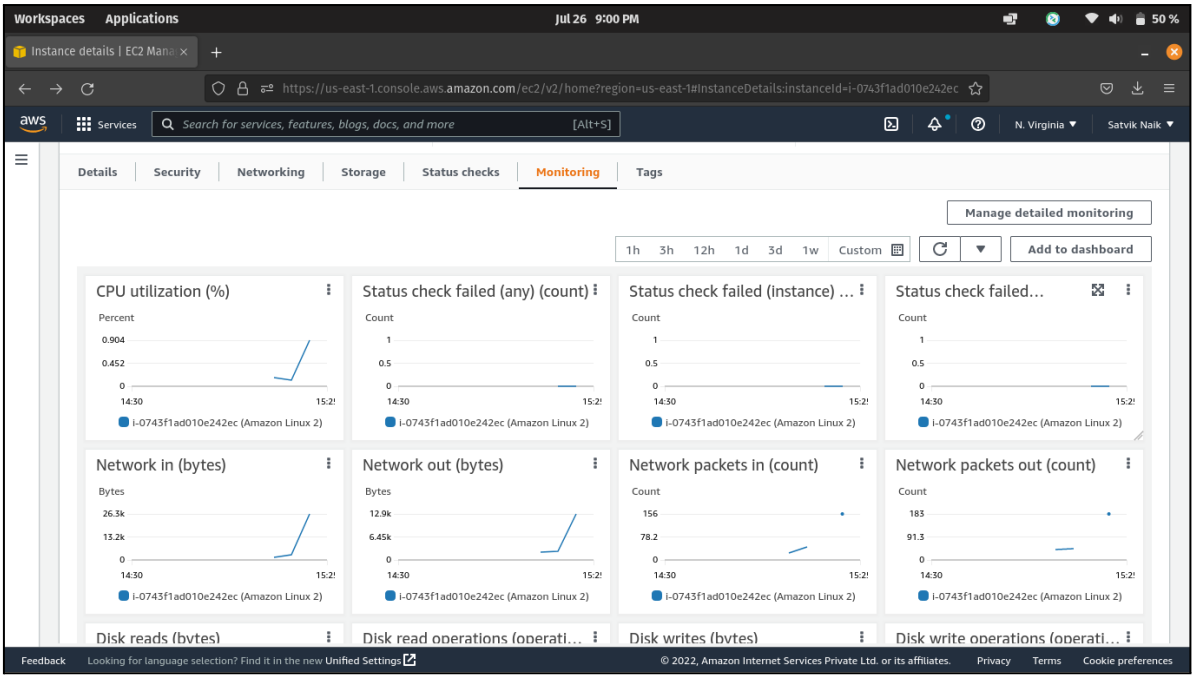
- Select EC2 instance connect.





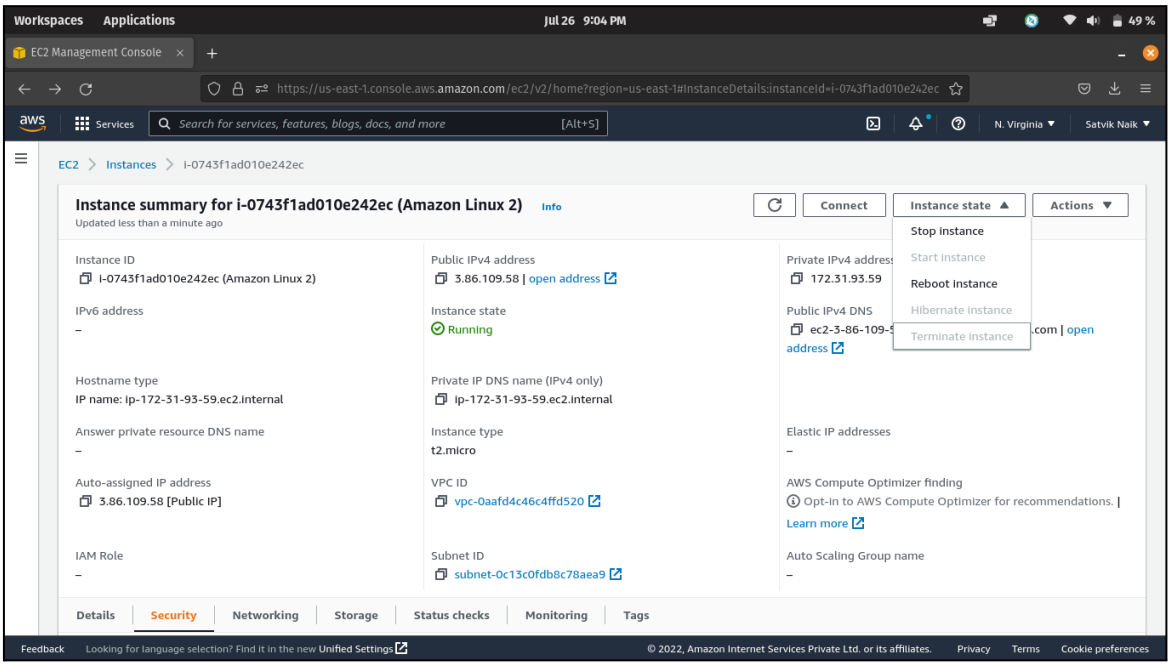
5. Monitoring EC2 Instance.

- EC2 instances can be monitored by visualizing various parameters such as disk R/W operations. CPU usage, network packets etc…

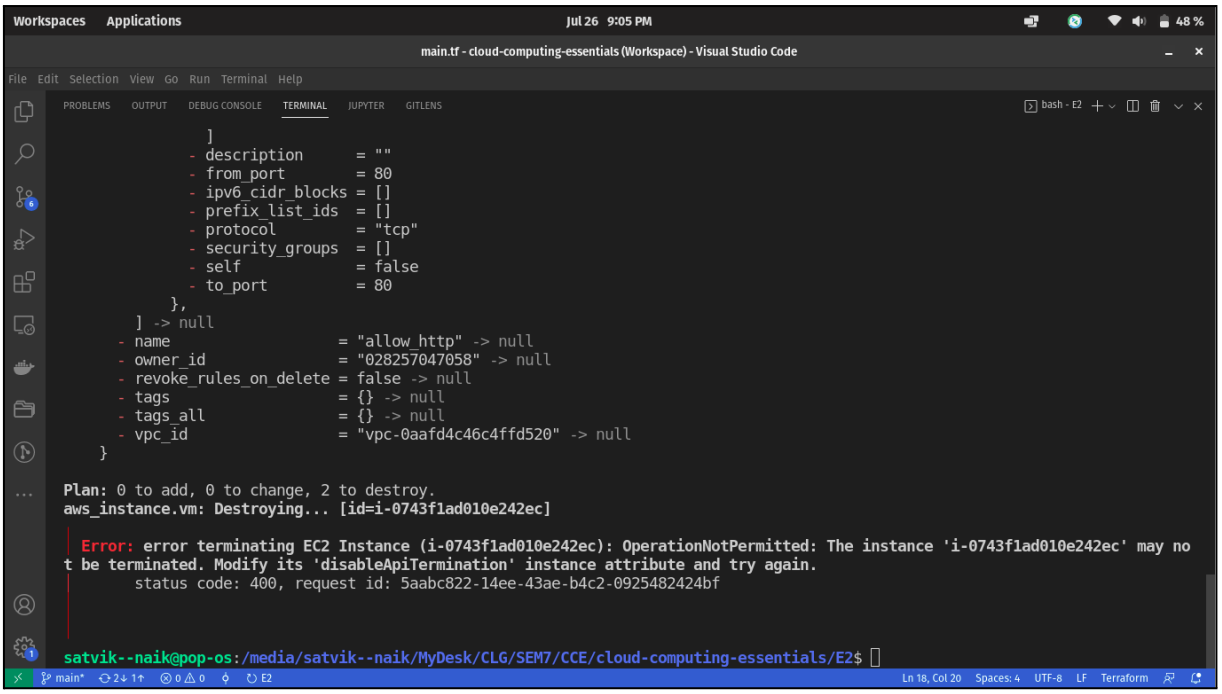


6. Test termination protection.

- In case you try to terminate the instance, it will prompt you with a message showing ***"Termination protection is enabled for one or more selected instances."***

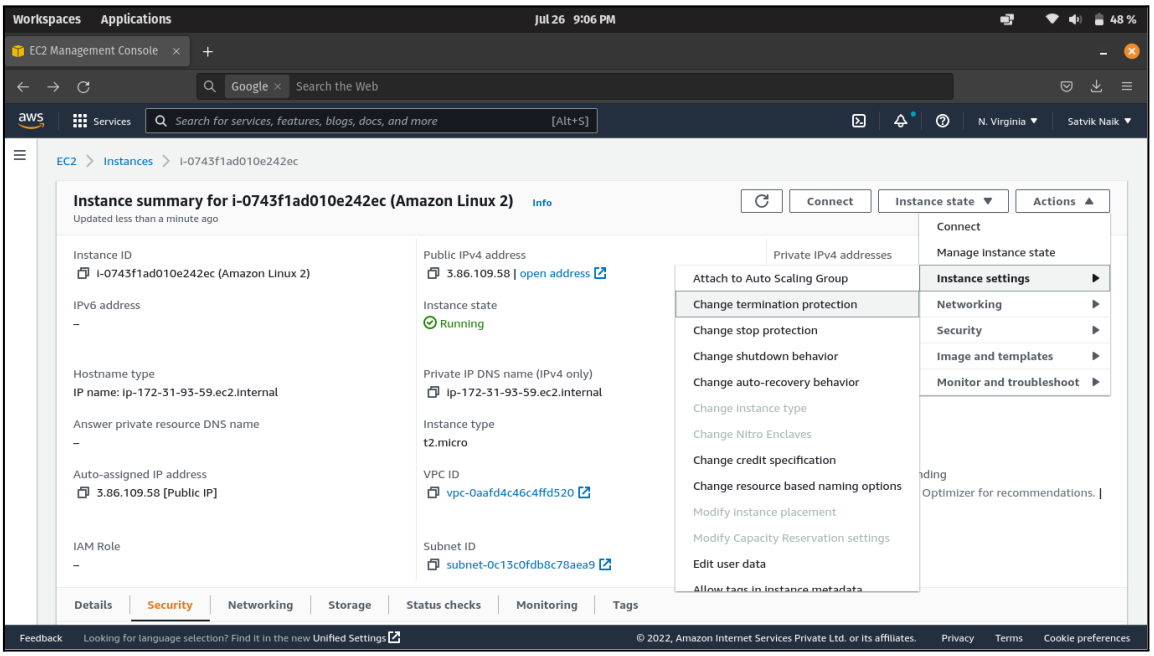


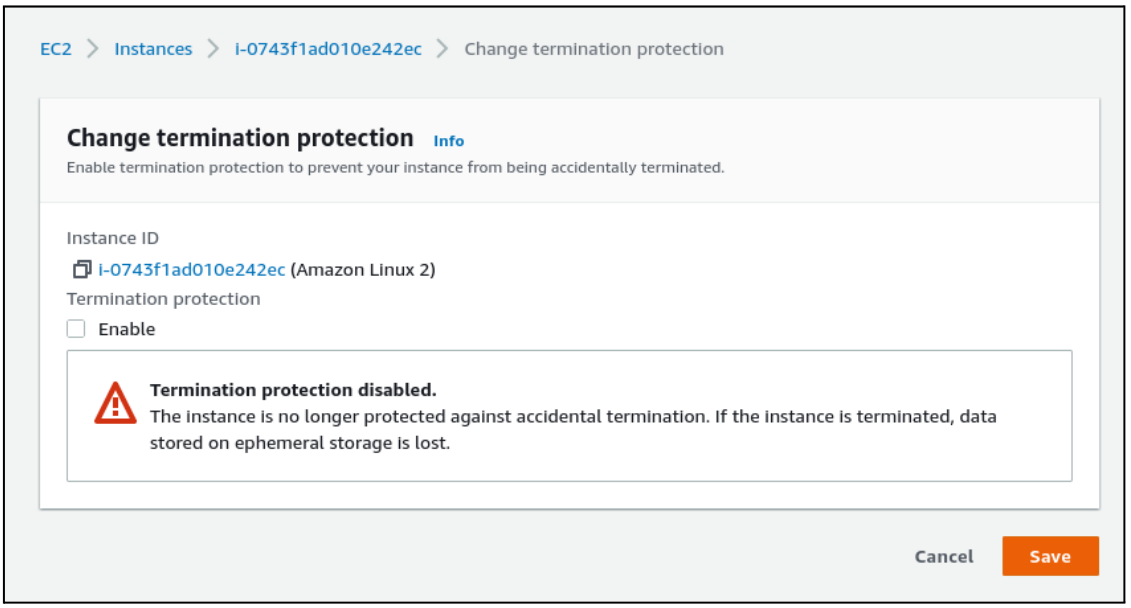
- In case if you try to do it via ***'terraform destroy'*** command then also it will show you the following error message. (i.e. OperationNotPermitter).



7. Terminate your EC2 Instance.

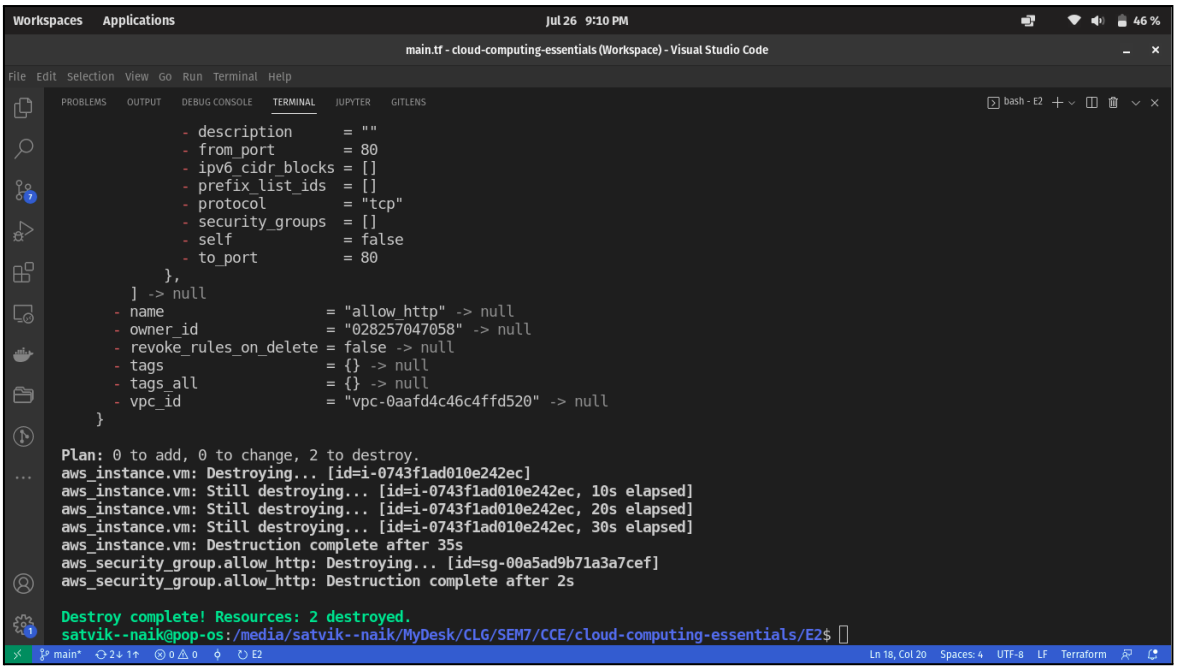
- In order to terminate your instance, first disable the termination protection.





Click on ***'Save'***.

- Now on trying to give the ***'terraform destroy'*** command, it will terminate the instance.



8. EC2 limits.

- 20 instances per region.
- 5 Elastic IP Addresses.
- Price variations.