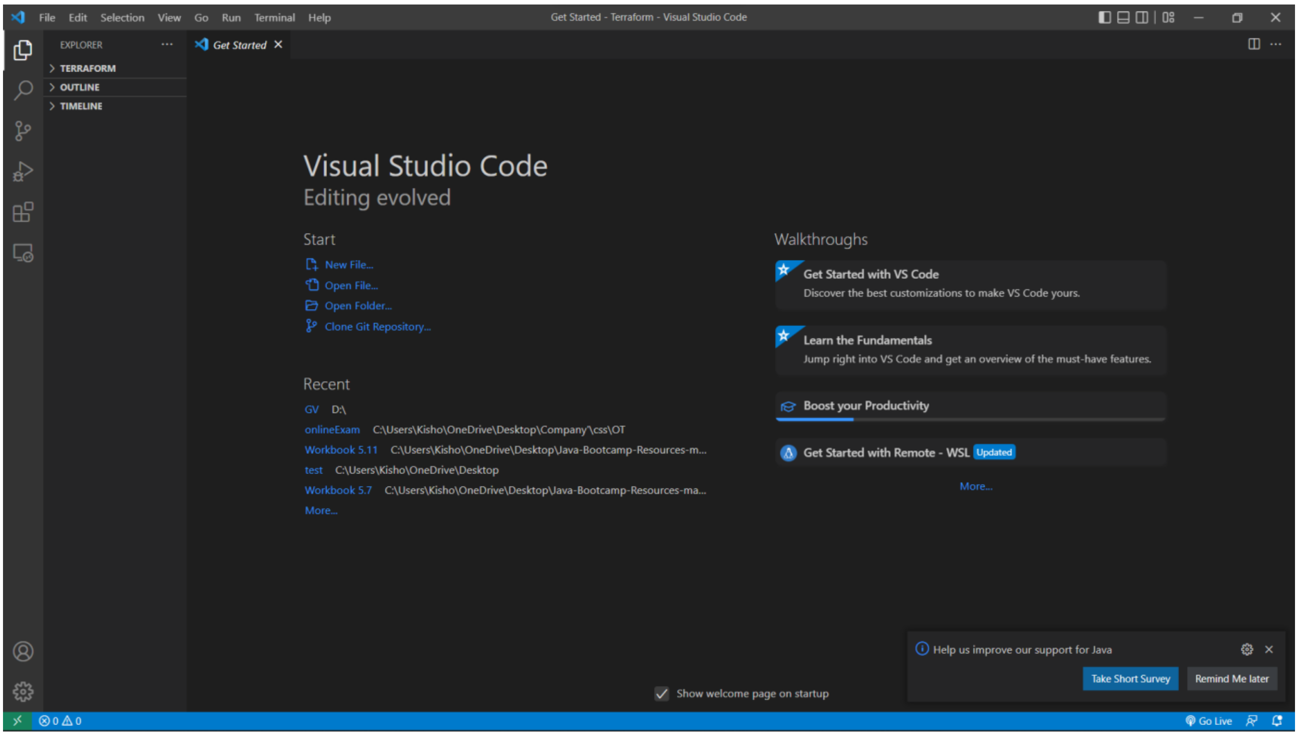
LAUNCHING VPC VIA TERRAFORM

1) Open the VS Code



Visual Studio Code
Editing evolved

GV D:\
onlineExam C:\Users\Kisho\OneDrive\Desktop\Company\css\OT
Workbook 5.11 C:\Users\Kisho\OneDrive\Desktop\Java-Bootcamp-Resources-m...
test C:\Users\Kisho\OneDrive\Desktop
Workbook 5.7 C:\Users\Kisho\OneDrive\Desktop\Java-Bootcamp-Resources-ma...

Get Started with VS Code
Discover the best customizations to make VS Code yours.

Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.

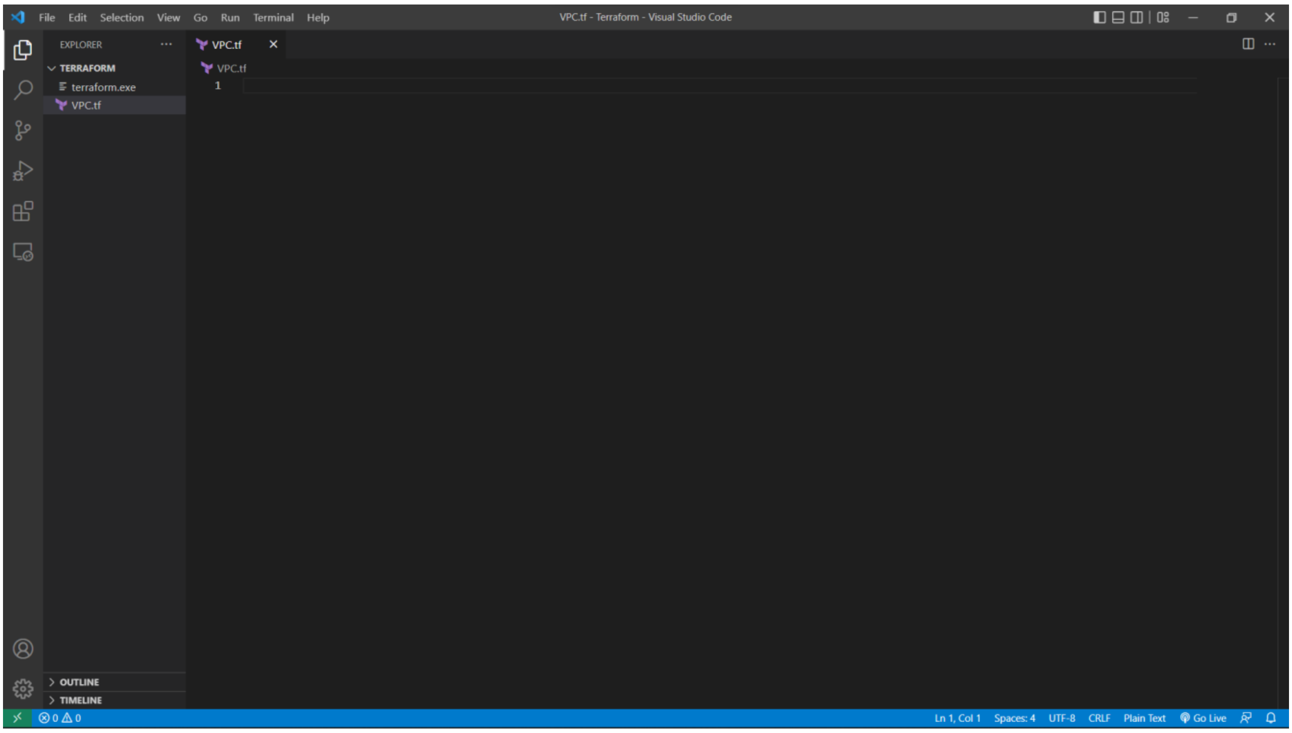
Boost your Productivity

Get Started with Remote - WSL Updated

Help us improve our support for Java

✓ Show welcome page on startup

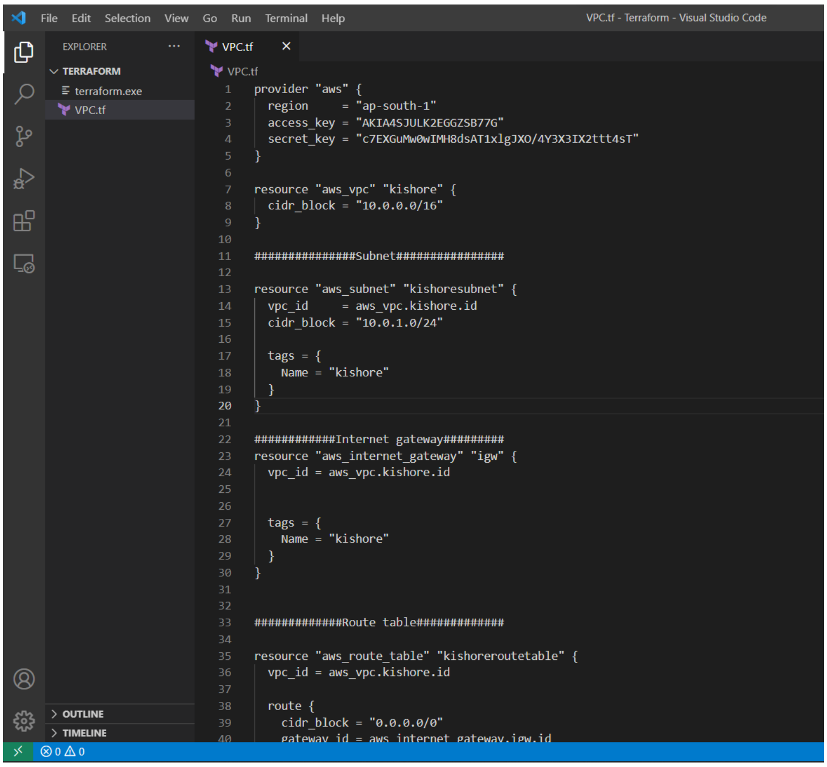
2) Click on File -> Open Folder (select the terraform.exe located folder) and create a new file and save it with an extension **.tf**



3) To create the access key in the AWS
   - **click on profile** -> **security credentials** -> **Access Keys** -> **create New Access Key**
   - download the access key file before closing the window

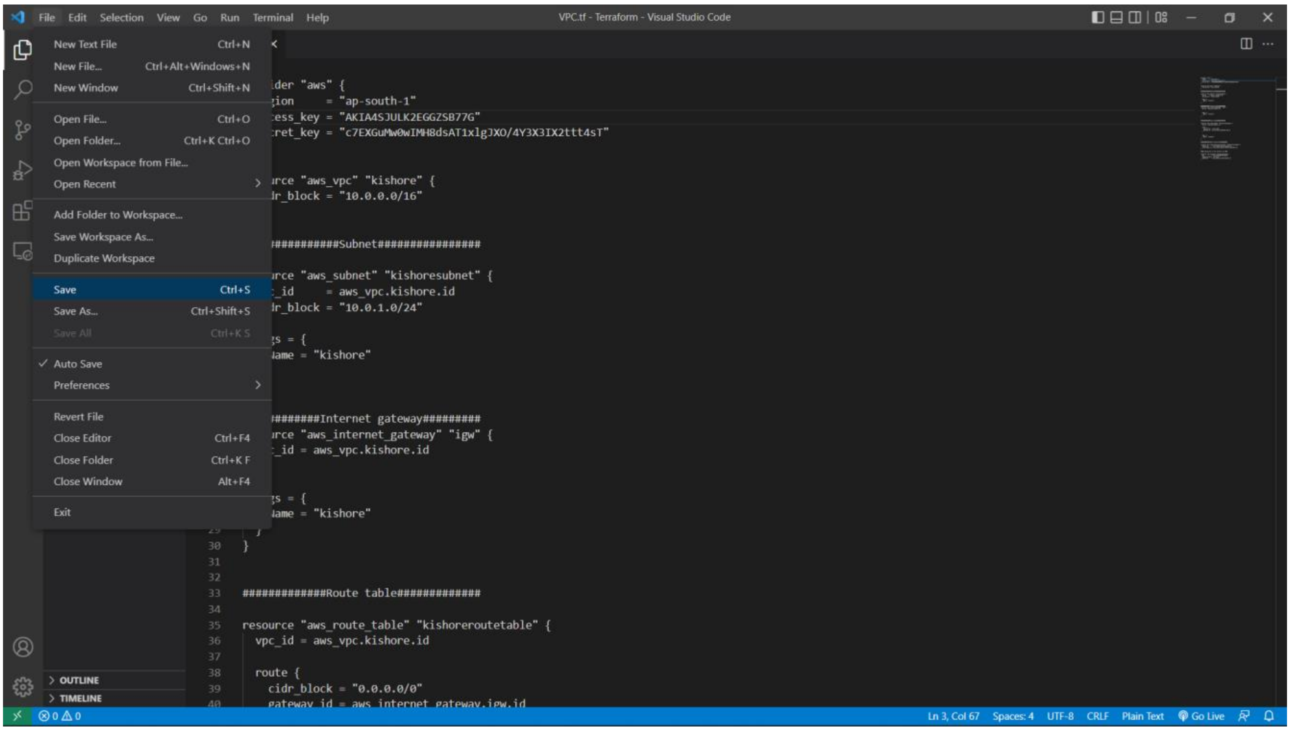
4) Paste the following code in VS code (**change the access key and secret key with your access key credentials**)

```
provider "aws" {
  region     = "ap-south-1"
  access_key = "AKIA4SJULK2EGGZSB77G"
  secret_key = "c7EXGuMw0wIMH8dsAT1xlgJXO/4Y3X3IX2ttt4sT"
}

resource "aws_vpc" "kishore" {
  cidr_block = "10.0.0.0/16"
}

###############Subnet################

resource "aws_subnet" "kishoresubnet" {
  vpc_id     = aws_vpc.kishore.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "kishore"
  }
}

############Internet gateway#########
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.kishore.id

  tags = {
    Name = "kishore"
  }
}

#############Route table#############

resource "aws_route_table" "kishoreroutetable" {
  vpc_id = aws_vpc.kishore.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }

  tags = {
    Name = "kishore"
  }
}

############Subnet association##########

resource "aws_route_table_association" "kishoreroutetable" {
  subnet_id      = aws_subnet.kishoresubnet.id
  route_table_id = aws_route_table.kishoreroutetable.id
}

####creating Ec2 in the kishore vpc ####

resource "aws_instance" "kishoreinstance" {
  ami           = "ami-076e3a557efe1aa9c"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.kishoresubnet.id
}
```



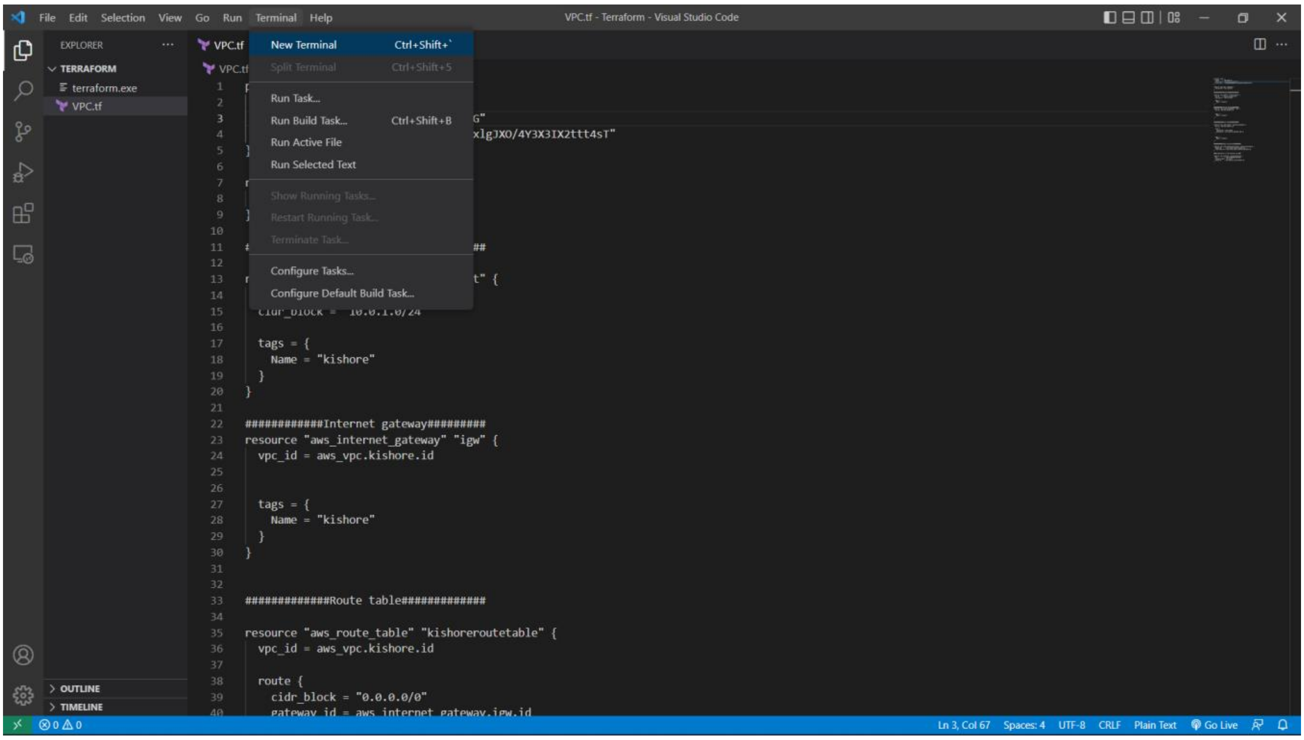
```
provider "aws" {
  region     = "ap-south-1"
  access_key = "AKIA4SJULK2EGGZSB77G"
  secret_key = "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"
}

resource "aws_vpc" "kishore" {
  cidr_block = "10.0.0.0/16"
}

###############Subnet################

resource "aws_subnet" "kishoresubnet" {
  vpc_id     = aws_vpc.kishore.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "kishore"
  }
}

############Internet gateway#########
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.kishore.id


  tags = {
    Name = "kishore"
  }
}


#############Route table#############

resource "aws_route_table" "kishoreroutetable" {
  vpc_id = aws_vpc.kishore.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
```

5) Save the file



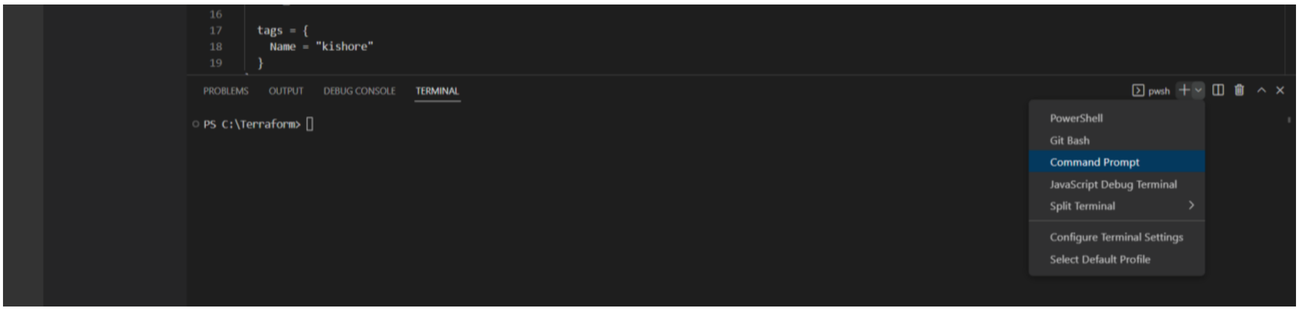
```
ider "aws" {
gion      = "ap-south-1"
cess_key = "AKIA4SJULK2EGGZSB77G"
cret_key = "c7EXGuMw0wIMH8dsAT1xlgJXO/4Y3X3IX2ttt4sT"


urce "aws_vpc" "kishore" {
dr_block = "10.0.0.0/16"


###########Subnet################

urce "aws_subnet" "kishoresubnet" {
_id      = aws_vpc.kishore.id
dr_block = "10.0.1.0/24"

gs = {
Name = "kishore"



########Internet gateway#########
urce "aws_internet_gateway" "igw" {
_id = aws_vpc.kishore.id


gs = {
Name = "kishore"
 }
}


#############Route table#############

resource "aws_route_table" "kishoreroutetable" {
  vpc_id = aws_vpc.kishore.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
```

6) Click on Terminal -> New Terminal



```
  tags = {
    Name = "kishore"
  }
}

############Internet gateway#########
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.kishore.id


  tags = {
    Name = "kishore"
  }
}


#############Route table#############

resource "aws_route_table" "kishoreroutetable" {
  vpc_id = aws_vpc.kishore.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
```

7) To Open the terminal as Command Prompt (follow the directions as per the following image)

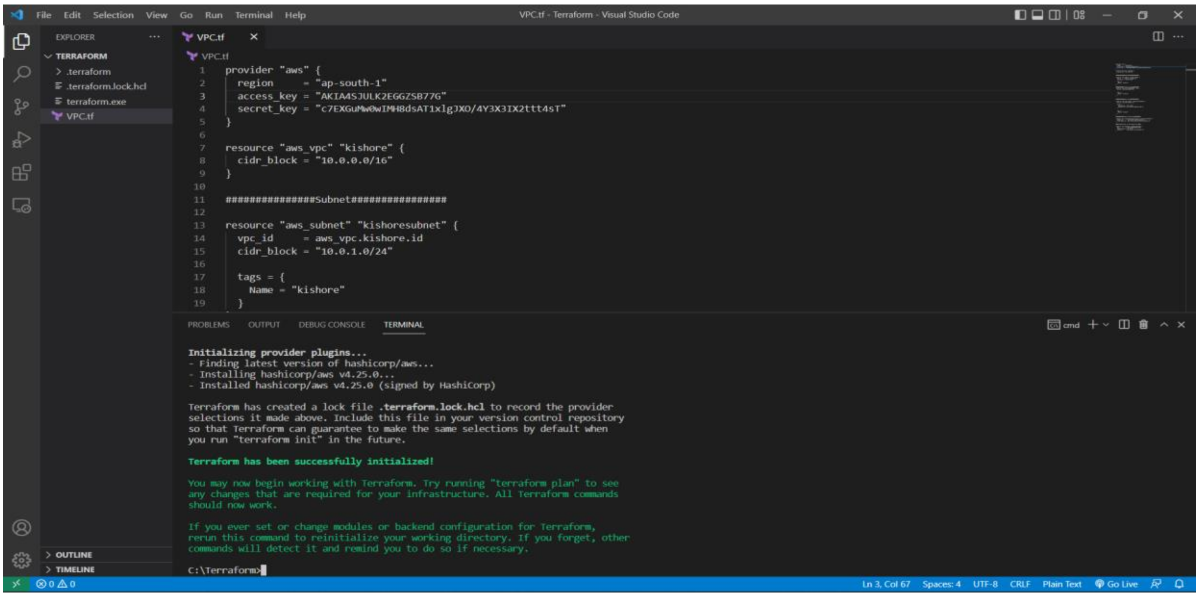


```
  tags = {
    Name = "kishore"
  }
```

```
PS C:\Terraform>
```

8) Enter the following Commands in the terminal

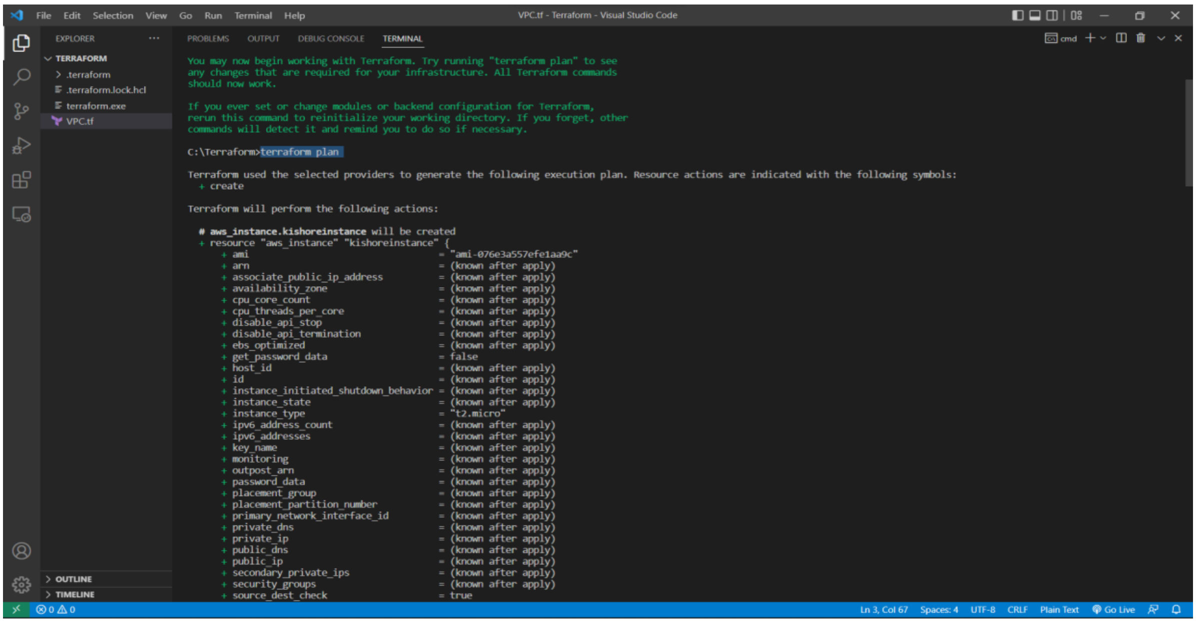
i. To initialize -> **terraform init**


File Edit Selection View Go Run Terminal Help
VPC.tf - Terraform - Visual Studio Code
EXPLORER
TERRAFORM
.terraform
.terraform.lock.hcl
terraform.exe
VPC.tf
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cmd
Ln 3, Col 67 Spaces: 4 UTF-8 CRLF Plain Text Go Live

```
provider "aws" {
  region     = "ap-south-1"
  access_key = "AKIA4SJULK2EGGZSB77G"
  secret_key = "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"
}

resource "aws_vpc" "kishore" {
  cidr_block = "10.0.0.0/16"
}

###############Subnet################

resource "aws_subnet" "kishoresubnet" {
  vpc_id     = aws_vpc.kishore.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "kishore"
  }
```

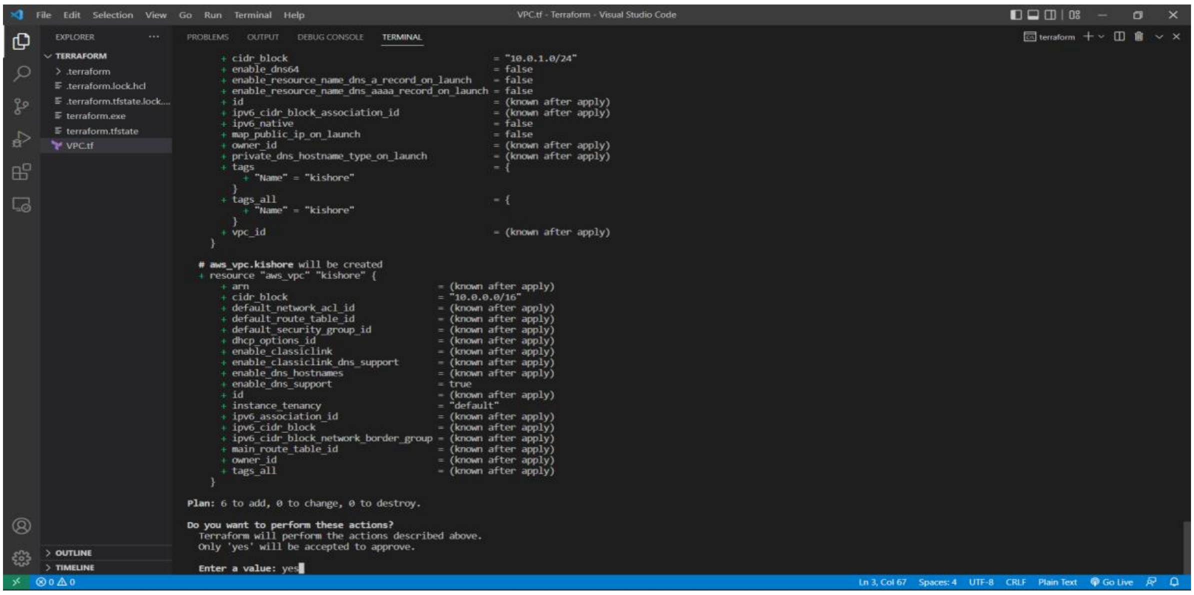
```
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.25.0...
- Installed hashicorp/aws v4.25.0 (signed by HashiCorp)

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.

C:\Terraform>
```

ii. To see any changes that are required for your infrastructure -> **terraform plan**


File Edit Selection View Go Run Terminal Help
VPC.tf - Terraform - Visual Studio Code
EXPLORER
TERRAFORM
.terraform
.terraform.lock.hcl
terraform.exe
VPC.tf
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cmd
Ln 3, Col 67 Spaces: 4 UTF-8 CRLF Plain Text Go Live

```
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.

C:\Terraform>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.kishoreinstance will be created
  + resource "aws_instance" "kishoreinstance" {
      + ami                                  = "ami-076e3a557efe1aa9c"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
```

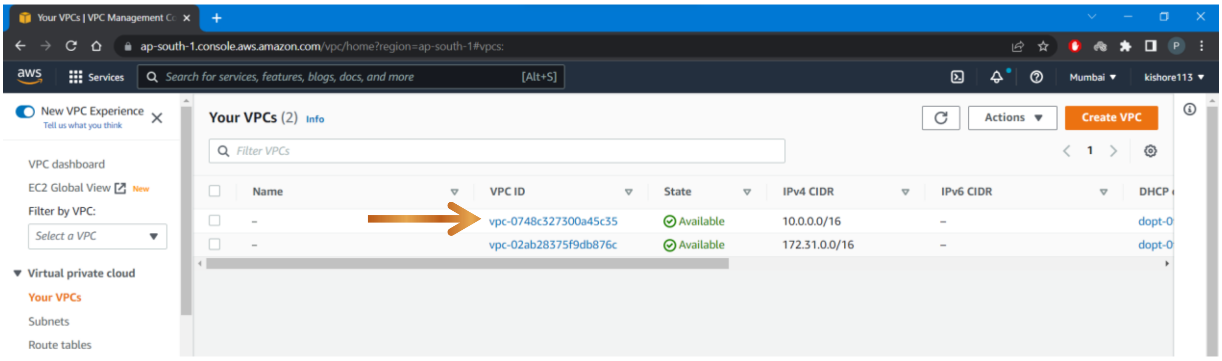
iii. Now to apply -> **terraform apply** -> it will ask you **enter the value:** type **yes** and press enter


File Edit Selection View Go Run Terminal Help
VPC.tf - Terraform - Visual Studio Code
EXPLORER
TERRAFORM
.terraform
.terraform.lock.hcl
.terraform.tfstate.lock....
terraform.exe
terraform.tfstate
VPC.tf
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
terraform
Ln 3, Col 67 Spaces: 4 UTF-8 CRLF Plain Text Go Live

```
      + cidr_block                                     = "10.0.1.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags                                           = {
          + "Name" = "kishore"
        }
      + tags_all                                       = {
          + "Name" = "kishore"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.kishore will be created
  + resource "aws_vpc" "kishore" {
      + 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_classiclink                   = (known after apply)
      + enable_classiclink_dns_support       = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + 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_all                             = (known after apply)
    }

Plan: 6 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
```

9) The resources will be created in AWS account, check for it

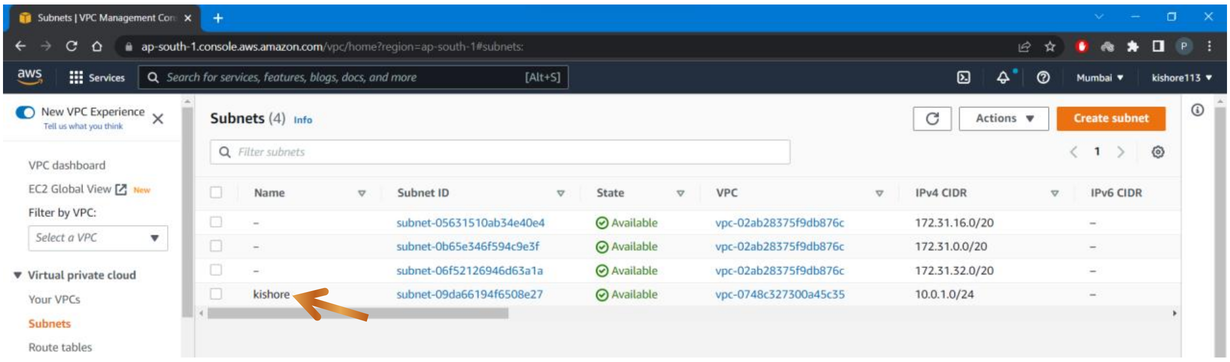
I. VPC created with given CIDR range



Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
–	vpc-0748c327300a45c35	Available	10.0.0.0/16	–	dopt-0
–	vpc-02ab28375f9db876c	Available	172.31.0.0/16	–	dopt-0

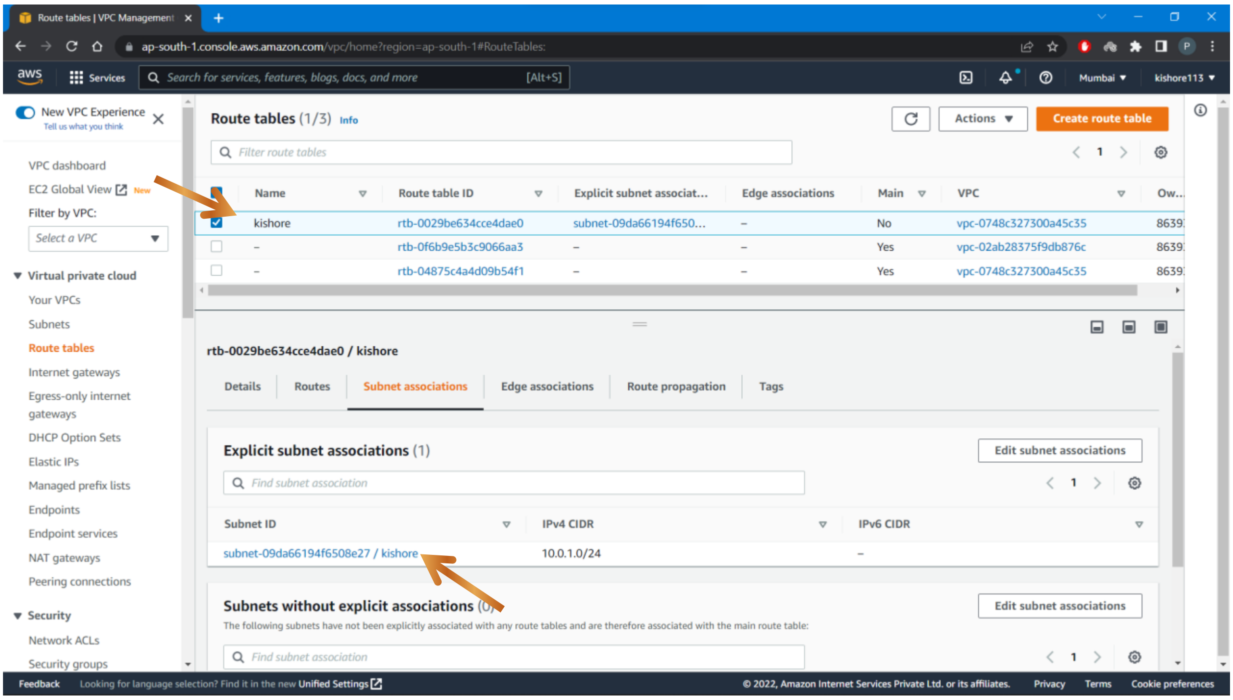
II. Subnet



Subnets (4) Info

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
–	subnet-05631510ab34e40e4	Available	vpc-02ab28375f9db876c	172.31.16.0/20	–
–	subnet-0b65e346f594c9e3f	Available	vpc-02ab28375f9db876c	172.31.0.0/20	–
–	subnet-06f52126946d63a1a	Available	vpc-02ab28375f9db876c	172.31.32.0/20	–
kishore	subnet-09da66194f6508e27	Available	vpc-0748c327300a45c35	10.0.1.0/24	–

III. Route table with subnet association



Route tables (1/3) Info

	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Ow...
☑	kishore	rtb-0029be634cce4dae0	subnet-09da66194f650...	–	No	vpc-0748c327300a45c35	8639
□	–	rtb-0f6b9e5b3c9066aa3	–	–	Yes	vpc-02ab28375f9db876c	8639
□	–	rtb-04875c4a4d09b54f1	–	–	Yes	vpc-0748c327300a45c35	8639

rtb-0029be634cce4dae0 / kishore

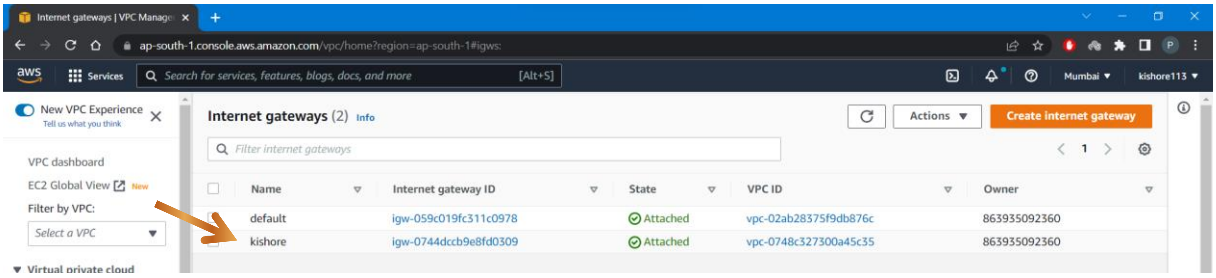
Explicit subnet associations (1)

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-09da66194f6508e27 / kishore	10.0.1.0/24	–

Subnets without explicit associations (0)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

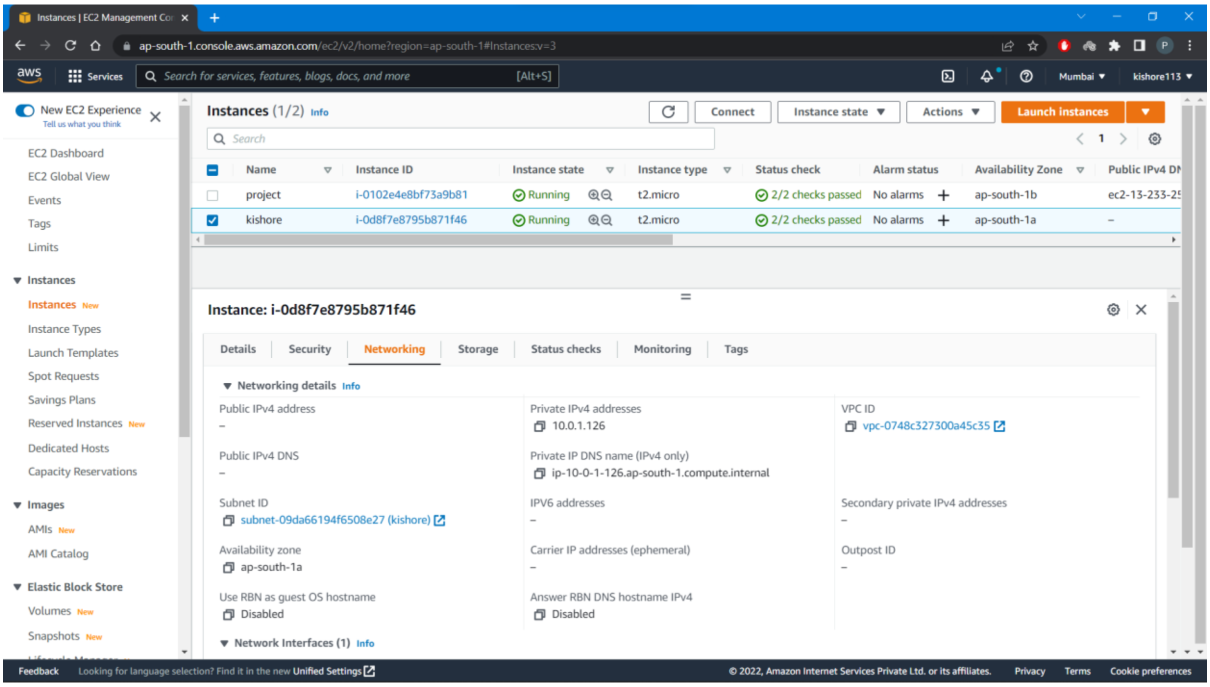
IV. Internet Gateway



Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID	Owner
default	igw-059c019fc311c0978	Attached	vpc-02ab28375f9db876c	863935092360
kishore	igw-0744dccb9e8fd0309	Attached	vpc-0748c327300a45c35	863935092360

V. EC2 instance within the new custom VPC



Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
□	project	i-0102e4e8bf73a9b81	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-233-2
☑	kishore	i-0d8f7e8795b871f46	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–

Instance: i-0d8f7e8795b871f46

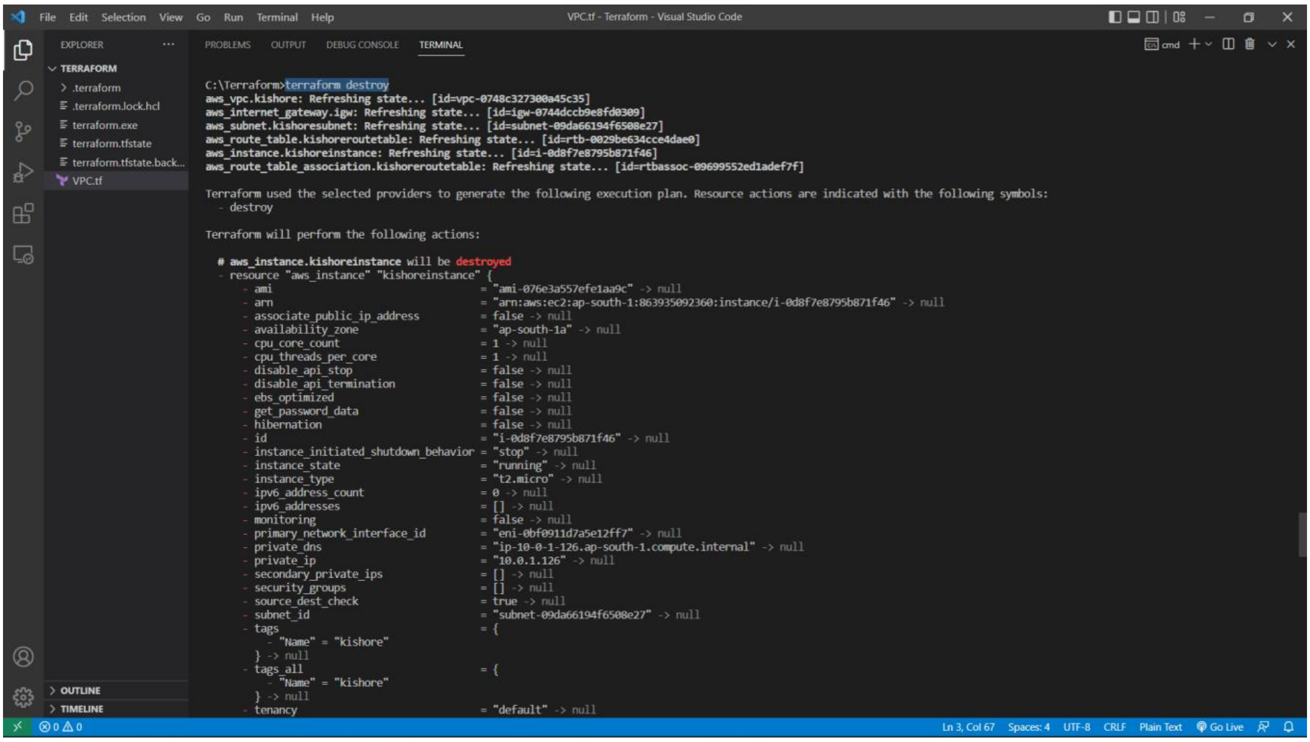
Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Networking details Info

Public IPv4 address –	Private IPv4 addresses 10.0.1.126	VPC ID vpc-0748c327300a45c35
Public IPv4 DNS –	Private IP DNS name (IPv4 only) ip-10-0-1-126.ap-south-1.compute.internal	
Subnet ID subnet-09da66194f6508e27 (kishore)	IPV6 addresses –	Secondary private IPv4 addresses –
Availability zone ap-south-1a	Carrier IP addresses (ephemeral) –	Outpost ID –
Use RBN as guest OS hostname Disabled	Answer RBN DNS hostname IPv4 Disabled	

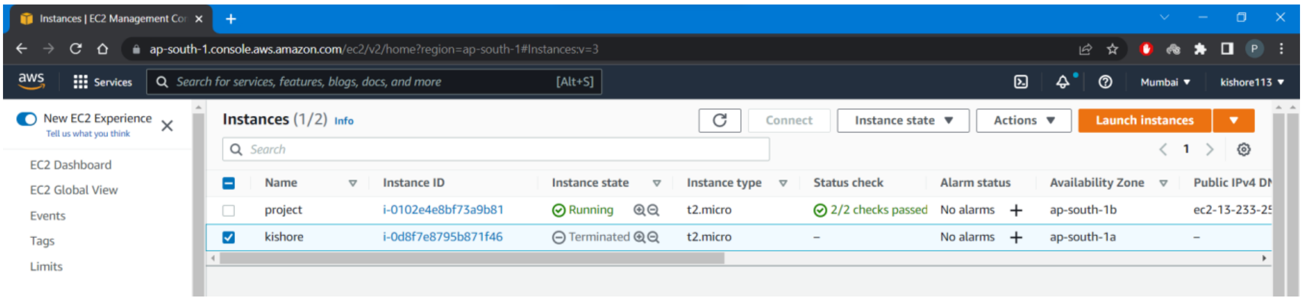
Network Interfaces (1) Info

10) To delete all the created resources, enter the command -> **terraform destroy** in VS code terminal



```
C:\Terraform>terraform destroy
aws_vpc.kishore: Refreshing state... [id=vpc-0748c327300a45c35]
aws_internet_gateway.igw: Refreshing state... [id=igw-0744dccb9e8fd0309]
aws_subnet.kishoresubnet: Refreshing state... [id=subnet-09da66194f6508e27]
aws_route_table.kishoreroutetable: Refreshing state... [id=rtb-0029be634cce4dae0]
aws_instance.kishoreinstance: Refreshing state... [id=i-0d8f7e8795b871f46]
aws_route_table_association.kishoreroutetable: Refreshing state... [id=rtbassoc-09699552ed1adef7f]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.kishoreinstance will be destroyed
  - resource "aws_instance" "kishoreinstance" {
      - ami                                  = "ami-076e3a557efe1aa9c" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:863935092360:instance/i-0d8f7e8795b871f46" -> null
      - associate_public_ip_address          = false -> null
      - availability_zone                    = "ap-south-1a" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-0d8f7e8795b871f46" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
      - instance_type                        = "t2.micro" -> null
      - ipv6_address_count                   = 0 -> null
      - ipv6_addresses                       = [] -> null
      - monitoring                           = false -> null
      - primary_network_interface_id         = "eni-0bf0911d7a5e12ff7" -> null
      - private_dns                          = "ip-10-0-1-126.ap-south-1.compute.internal" -> null
      - private_ip                           = "10.0.1.126" -> null
      - secondary_private_ips                = [] -> null
      - security_groups                      = [] -> null
      - source_dest_check                    = true -> null
      - subnet_id                            = "subnet-09da66194f6508e27" -> null
      - tags                                 = {
          - "Name" = "kishore"
        } -> null
      - tags_all                             = {
          - "Name" = "kishore"
        } -> null
      - tenancy                              = "default" -> null
```

- Instance was terminated



Instances | EC2 Management Co
ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:v=3
Services
Search for services, features, blogs, docs, and more [Alt+S]
Mumbai
kishore113
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Connect
Instance state
Actions
Launch instances
Search

Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
□	project	i-0102e4e8bf73a9b81	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-13-233-2
☑	kishore	i-0d8f7e8795b871f46	Terminated	t2.micro	–	No alarms	ap-south-1a	–