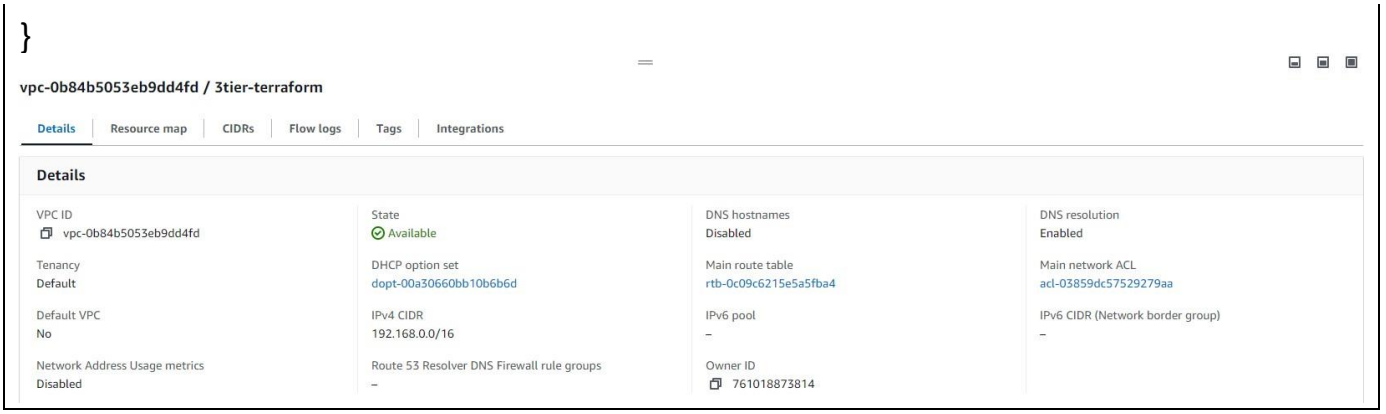
DEVOPS

No:-	Content
1.	**Terraform** • **VPC** • **Subnet** • **IGW** • **Route Table** • **Subnet Association** • **Create Security Group** • **Launch instance**
2.	• **Variables** • **Output variables**

Create VPC

```
resource "aws_vpc" "tier-3-project" { cidr_block =

"192.168.0.0/16" tags = {

Name ="3tier-terraform"



}
```

```
}
```



Create Subnet

```
resource "aws_subnet" "sub1" {

vpc_id = aws_vpc.tier-3-project.id

cidr_block = "192.168.1.0/24"

availability_zone = "us-east-1a"

map_public_ip_on_launch = "true"



}
```

Create IGW

```
resource "aws_internet_gateway" "int" {

vpc_id=aws_vpc.tier-3-project.id tags = {

Name="igw-40-40"
```

```
}

}
```

Create Route Table

```
resource "aws_route_table" "Public-RT" {

vpc_id = aws_vpc.tier-3-project.id

tags = {

Name="Public-RT-40"

}
```

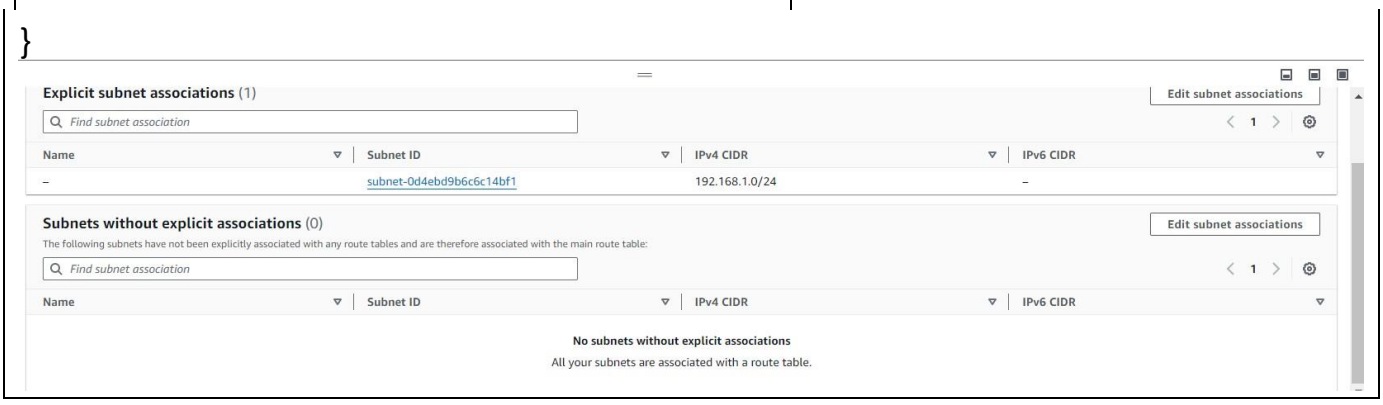

```
#   route = [{

#     gateway_id=aws_internet_gateway.int.id

#     cidr_block="0.0.0.0/0"


#   }]

}
```

Routing

```
resource "aws_route" "route_to_internet" {

route_table_id         = aws_route_table.Public-RT.id

destination_cidr_block = "0.0.0.0/0" gateway_id

= aws_internet_gateway.int.id

}
```

Subnet association

```
resource "aws_route_table_association" "demo" {

subnet_id = aws_subnet.sub1.id route_table_id =

aws_route_table.Public-RT.id



}
```



Create Security Group

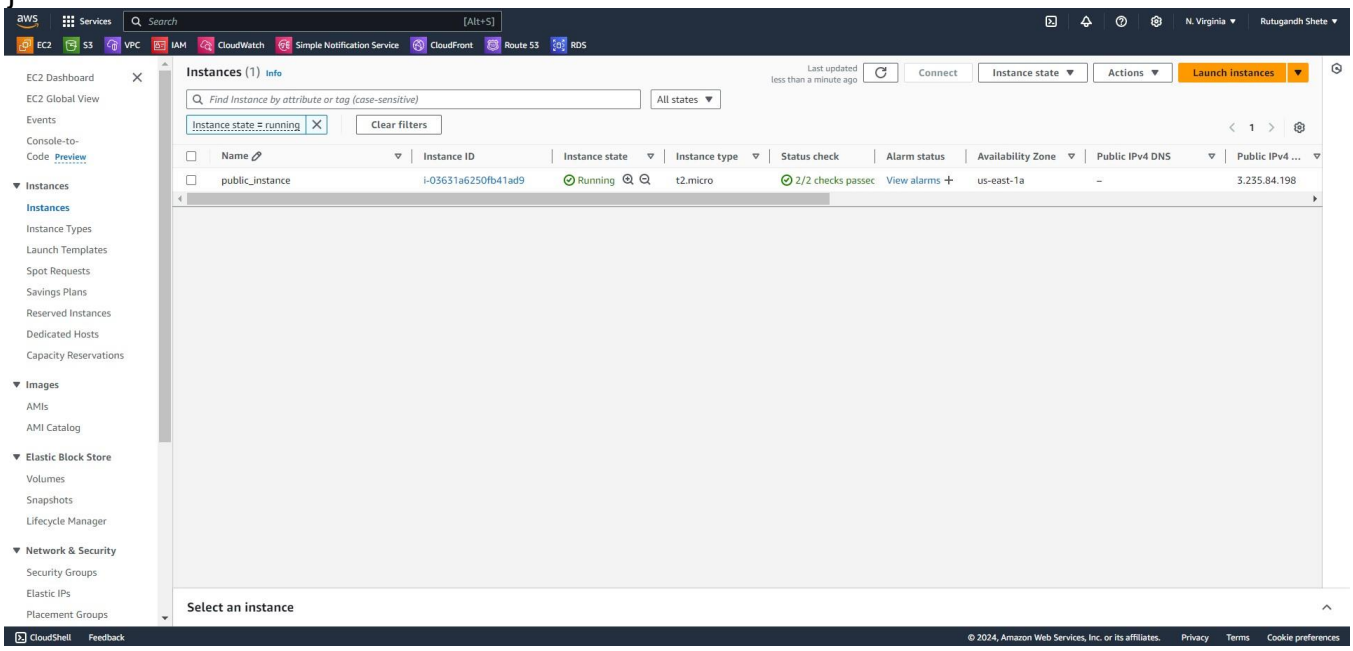
```
resource "aws_security_group" "sg" {

vpc_id = aws_vpc.tier-3-project.id tags

= {

Name="sg_3tier"

}

ingress = [{ from_port     = 22 to_port       = 22
protocol      = "tcp" cidr_blocks   = ["0.0.0.0/0"]
description   = "Allow SSH access from anywhere"
ipv6_cidr_blocks  = [] prefix_list_ids   = []
```

```
security_groups   = []
self              = false
}]
egress = [{
from_port         = 0
to_port           = 0
protocol          = "-1"
cidr_blocks       = ["0.0.0.0/0"]
description       = "Allow all outbound traffic"
ipv6_cidr_blocks  = []
prefix_list_ids   = []
security_groups   = []
self              = false
}]
```


```
}
```

Create Security Group

```
resource "aws_instance" "inst" {

ami = "ami-0866a3c8686eaeeba"

instance_type = "t2.micro"

key_name="new_key"

subnet_id = aws_subnet.sub1.id

vpc_security_group_ids = [aws_security_group.sg.id]

tags = {

Name="public_instance"

}



}
```



Variables

What are Variables in Terraform?

- **Variables** allow you to define values once and use them in multiple places.
- They make your Terraform scripts more dynamic, as you can change the variable values without modifying the actual code.
- Variables can be defined in a file called variables.tf, or they can be specified directly in the Terraform configuration file.

Data Types for Variables

Here are the common data types you can use for variables in Terraform:

String

Description: A sequence of characters (text).

Example: "Hello, World!"

Number

Description: A numeric value (integer or float).

Example: 42 or 3.14

Boolean

Description: Represents a true or false value.

Example: true or false

List

Description: An ordered collection of values (can be of the same or different types).

Example: ["value1", "value2", "value3"]

Map

Description: A collection of key-value pairs.

: {"key1" = "value1", "key2" = "value2"}

What are Output Variables?

- **Outputs** provide a way to return values from your Terraform configurations.
- They can be displayed in the terminal after you run terraform apply.
- Outputs can be used by other Terraform configurations if you're using modules.

```
output "publicip" {
  value = aws_instance.instance_prac.public_ip
}
```

```
resource "aws_vpc" "vpc_prac" {
  cidr_block = "192.168.0.0/16"
  tags = {
    Name="VPC_prac"
  }
}

resource "aws_subnet" "s1" {
  vpc_id = aws_vpc.vpc_prac.id
  cidr_block = var.subnet[0]
  availability_zone = var.az[0]
  map_public_ip_on_launch = true
  tags = {
    Name="subnet_prac"
  }
}
resource "aws_internet_gateway" "int_prac" {
  vpc_id = aws_vpc.vpc_prac.id
  tags={
    Name="IGW_prac"
  }
}
resource "aws_route_table" "rt" {
  vpc_id = aws_vpc.vpc_prac.id
  tags={
    Name="RT_prac"
  }
}
resource "aws_route" "route" {
  route_table_id = aws_route_table.rt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id = aws_internet_gateway.int_prac.id
}
resource "aws_route_table_association" "name" {
  route_table_id = aws_route_table.rt.id
  subnet_id = aws_subnet.s1.id
```

```
}
resource "aws_security_group" "sg_prac" {
  vpc_id = aws_vpc.vpc_prac.id
  tags = {
    Name="sg_prac"
  }
  ingress {
    from_port=22
    to_port=22
    protocol="tcp"
    cidr_blocks=["0.0.0.0/0"]
    description="allow ssh port"
  }
  egress {
    from_port=0
    to_port=0
    protocol="-1"
    cidr_blocks=["0.0.0.0/0"]

  }
}

resource "aws_instance" "instance_prac" {
  ami = var.ami_id
  instance_type = var.instance_type
  subnet_id = aws_subnet.s1.id
  key_name = var.key_pair[0]
  vpc_security_group_ids = [aws_security_group.sg_prac.id]
  tags = {
    Name="public_instance"
  }
}
output "publicip" {
  value = aws_instance.instance_prac.public_ip
}
```

```
variable "subnet"{
    type=list(string)
    default = [ "192.168.1.0/24","192.168.2.0/24" ]
}
variable "az" {
  type = list(string)
  default = ["us-east-1a","us-east-2"]
}
# variable "port" {
#   type = list(number)
#   default = [  ]
# }
```

```
variable "ami_id" {
  type = string
  description = "ami_id"
  default = "ami-0866a3c8686eaeeba"
}
variable "instance_type" {
  type=string
  default = "t2.micro"
}
variable "key_pair" {
  type=list(string)
  default = ["new_key","key"]
}
```