Complete Terraform Documentation for AWS VPC + Subnet + Route Table + Security Group + EC2 (Step-by-Step Guide)

1. Project Title

Terraform Infrastructure Setup on AWS (VPC, Subnets, Internet Gateway, Route Tables, Security Group, and EC2 Instance)

---

2. Project Objective

The main objective of this project is to create a complete AWS infrastructure using Terraform. This includes:

- Custom VPC
- Public and Private Subnets
- Internet Gateway
- Route Tables and Associations
- Security Group (SSH Access)
- EC2 Instance Deployment

This project is useful for DevOps, Cloud Computing, and MCA final year practical implementation.

---

4. Prerequisites (Before Starting)

You must install and configure the following:

4.1 Install Terraform

Check installation:

```
terraform -version
```

4.2 Install AWS CLI

Check:

```
aws --version
```

4.3 Configure AWS Credentials

Command:

aws configure

Enter:

- Access Key
- Secret Key
- Region (example: us-west-1)
- Output format: json

---

5. Project Folder Structure

Recommended structure:

```
terraform-project/
├── provider.tf
├── vpc.tf
├── ec2.tf
```

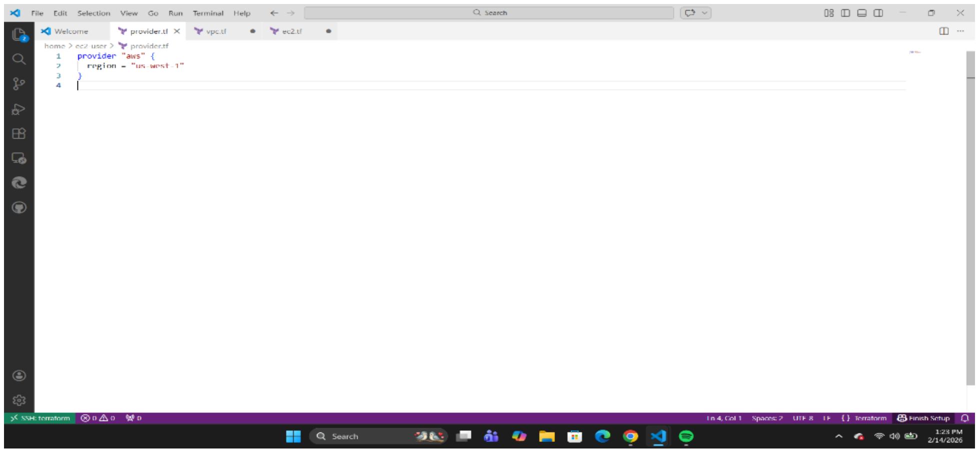
---

6. Provider Configuration (Very Important)

File: provider.tf

Correct Code:

```
provider "aws" {
  region = "us-west-1"
}
```



File Edit Selection View Go Run Terminal Help
Search
Welcome
provider.tf
vpc.tf
ec2.tf
home > ec2-user > provider.tf

```
provider "aws" {
  region = "us-west-1"
}
```

Explanation:

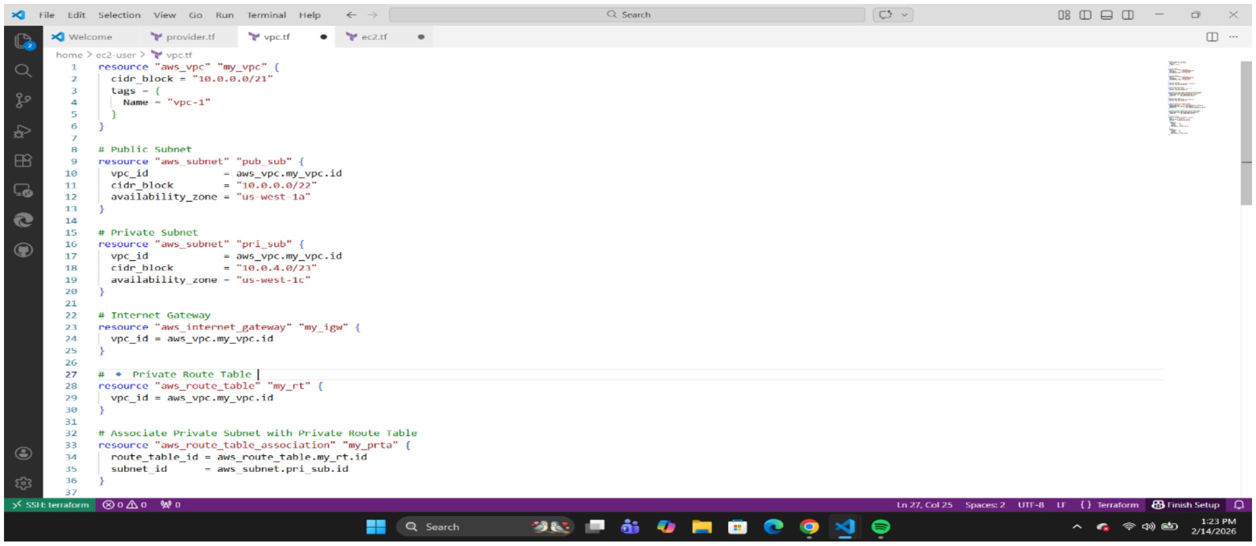
- Defines AWS region where resources will be created
- Region must be inside quotes
- If region mismatch, resources will not appear in AWS console

---

7. VPC Configuration

File: vpc.tf

```
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/21"
  tags = {
    Name = "vpc-1"
  }
}
```



```
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/21"
  tags = {
    Name = "vpc-1"
  }
}

# Public Subnet
resource "aws_subnet" "pub_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.0.0/22"
  availability_zone = "us-west-1a"
}

# Private Subnet
resource "aws_subnet" "pri_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.4.0/23"
  availability_zone = "us-west-1c"
}

# Internet Gateway
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id
}

#  •  Private Route Table
resource "aws_route_table" "my_rt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Associate Private Subnet with Private Route Table
resource "aws_route_table_association" "my_prta" {
  route_table_id = aws_route_table.my_rt.id
  subnet_id      = aws_subnet.pri_sub.id
}
```

SSH: terraform
Ln 27, Col 25 Spaces: 2 UTF-8 LF Terraform Finish Setup
1:23 PM
2/14/2026

Explanation:

- Creates custom VPC
- CIDR block defines IP range of network
- /21 provides large IP pool

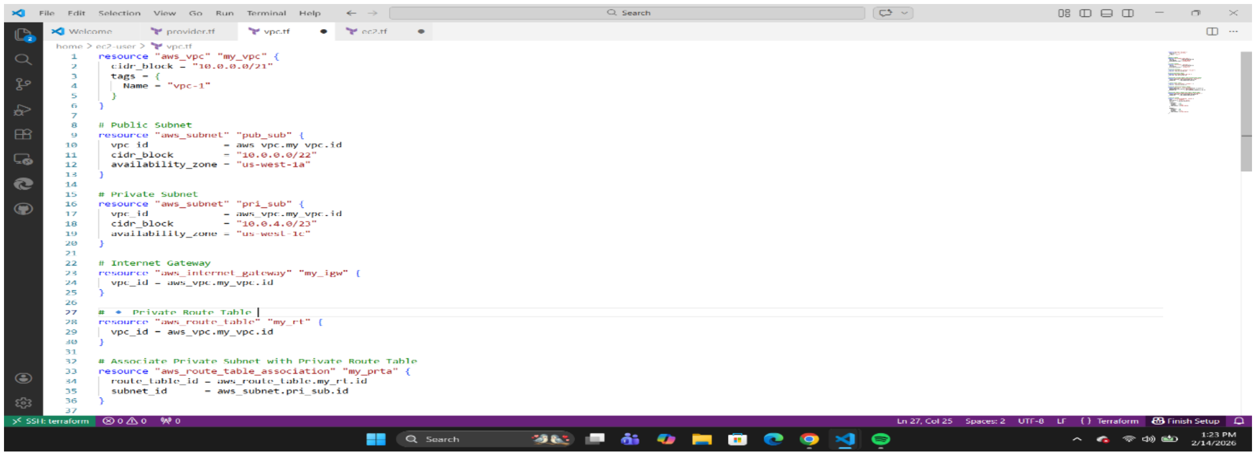
---

8. Subnet Configuration

Subnets divide VPC into smaller networks.

8.1 Public Subnet

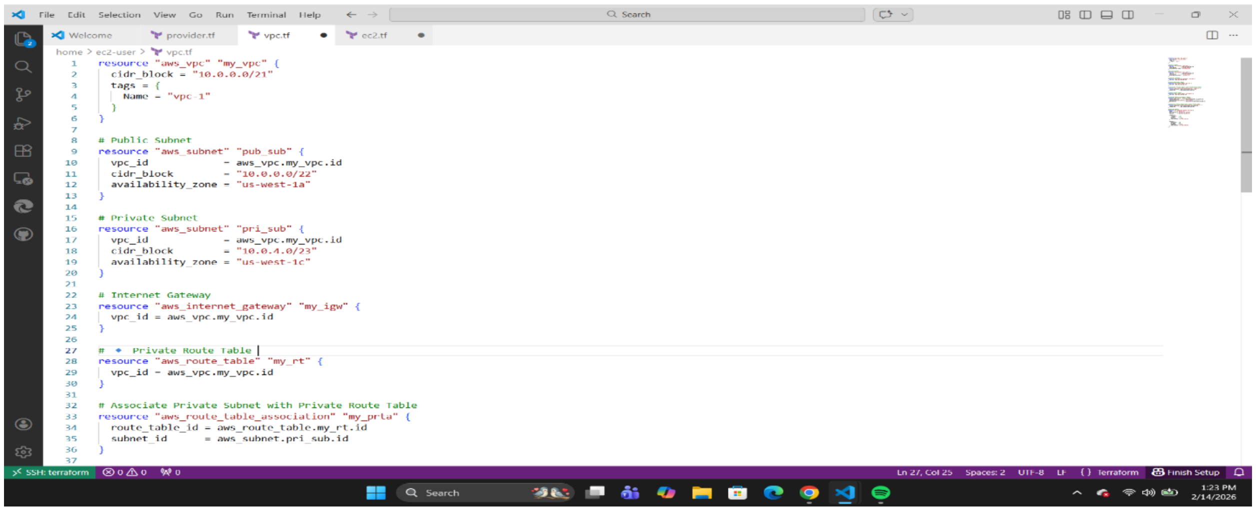
```
resource "aws_subnet" "pub_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.0.0/22"
  availability_zone = "us-west-1a"
  tags = {
    Name = "public-subnet"
  }
}
```



```
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/21"
  tags = {
    Name = "vpc-1"
  }
}

# Public Subnet
resource "aws_subnet" "pub_sub" {
  vpc_id                 = aws_vpc.my_vpc.id
  cidr_block             = "10.0.0.0/22"
  availability_zone = "us-west-1a"
}

# Private Subnet
resource "aws_subnet" "pri_sub" {
  vpc_id                 = aws_vpc.my_vpc.id
  cidr_block             = "10.0.4.0/23"
  availability_zone = "us-west-1c"
}

# Internet Gateway
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id
}

#  ◆  Private Route Table
resource "aws_route_table" "my_rt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Associate Private Subnet with Private Route Table
resource "aws_route_table_association" "my_prta" {
  route_table_id = aws_route_table.my_rt.id
  subnet_id      = aws_subnet.pri_sub.id
}
```

8.2 Private Subnet

```
resource "aws_subnet" "pri_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.4.0/23"
  availability_zone = "us-west-1c"
  tags = {
    Name = "private-subnet"
  }
}
```



```
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/21"
  tags = {
    Name = "vpc-1"
  }
}

# Public Subnet
resource "aws_subnet" "pub_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.0.0/22"
  availability_zone = "us-west-1a"
}

# Private Subnet
resource "aws_subnet" "pri_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.4.0/23"
  availability_zone = "us-west-1c"
}

# Internet Gateway
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id
}

# ◆ Private Route Table
resource "aws_route_table" "my_rt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Associate Private Subnet with Private Route Table
resource "aws_route_table_association" "my_prta" {
  route_table_id = aws_route_table.my_rt.id
  subnet_id      = aws_subnet.pri_sub.id
}
```

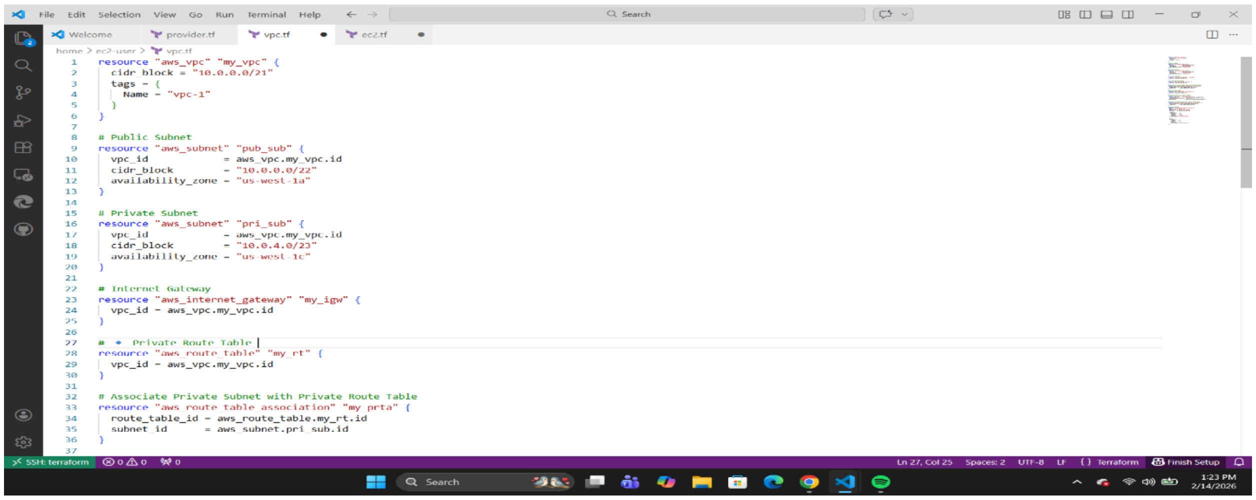
Important:

- Availability Zone must match region
- us-west-1 supports only us-west-1a and us-west-1c

---

9. Internet Gateway Configuration

```
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id
  tags = {
    Name = "my-igw"
  }
}
```



```
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/21"
  tags = {
    Name = "vpc-1"
  }
}

# Public Subnet
resource "aws_subnet" "pub_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.0.0/22"
  availability_zone = "us-west-1a"
}

# Private Subnet
resource "aws_subnet" "pri_sub" {
  vpc_id            = aws_vpc.my_vpc.id
  cidr_block        = "10.0.4.0/23"
  availability_zone = "us-west-1c"
}

# Internet Gateway
resource "aws_internet_gateway" "my_igw" {
  vpc_id = aws_vpc.my_vpc.id
}

# ◆ Private Route Table
resource "aws_route_table" "my_rt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Associate Private Subnet with Private Route Table
resource "aws_route_table_association" "my_prta" {
  route_table_id = aws_route_table.my_rt.id
  subnet_id      = aws_subnet.pri_sub.id
}
```

Purpose:

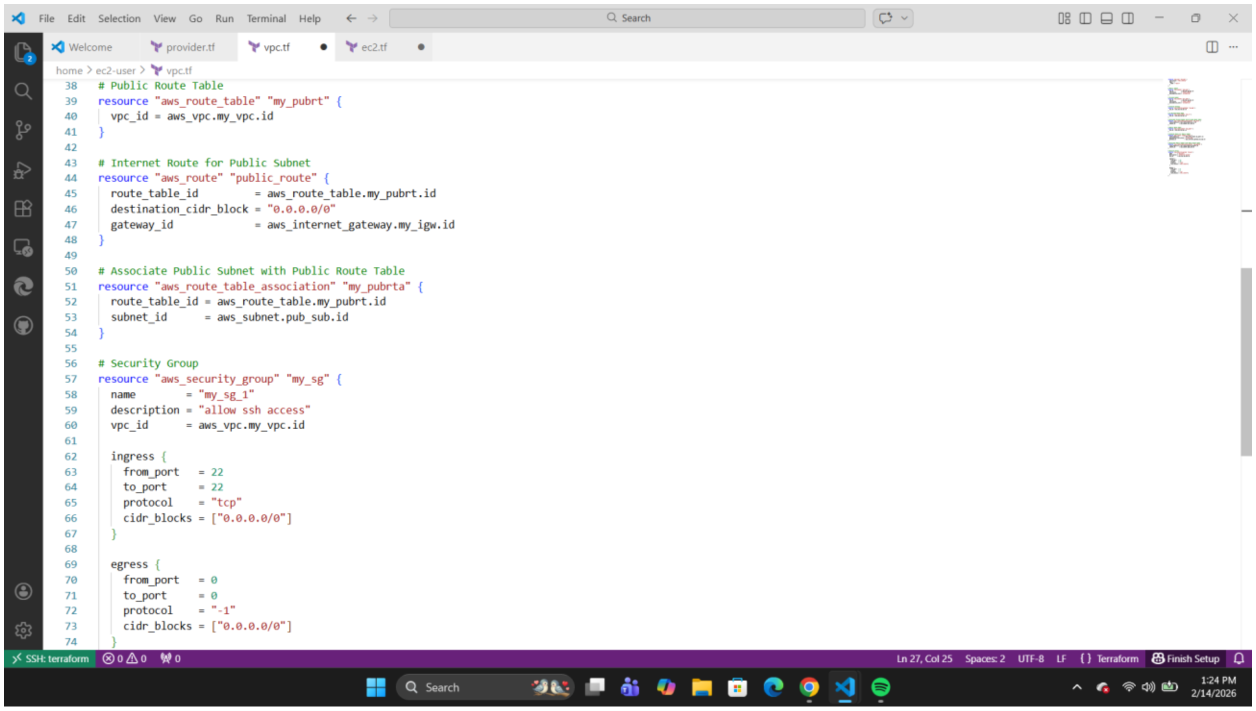
- Allows internet access to public subnet resources

---

10. Route Table Configuration (Public + Private)

10.1 Private Route Table

```
resource "aws_route_table" "my_rt" {
  vpc_id = aws_vpc.my_vpc.id
  tags = {
    Name = "private-rt"
  }
}
```



```
# Public Route Table
resource "aws_route_table" "my_pubrt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Internet Route for Public Subnet
resource "aws_route" "public_route" {
  route_table_id         = aws_route_table.my_pubrt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.my_igw.id
}

# Associate Public Subnet with Public Route Table
resource "aws_route_table_association" "my_pubrta" {
  route_table_id = aws_route_table.my_pubrt.id
  subnet_id      = aws_subnet.pub_sub.id
}

# Security Group
resource "aws_security_group" "my_sg" {
  name        = "my_sg_1"
  description = "allow ssh access"
  vpc_id      = aws_vpc.my_vpc.id

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

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

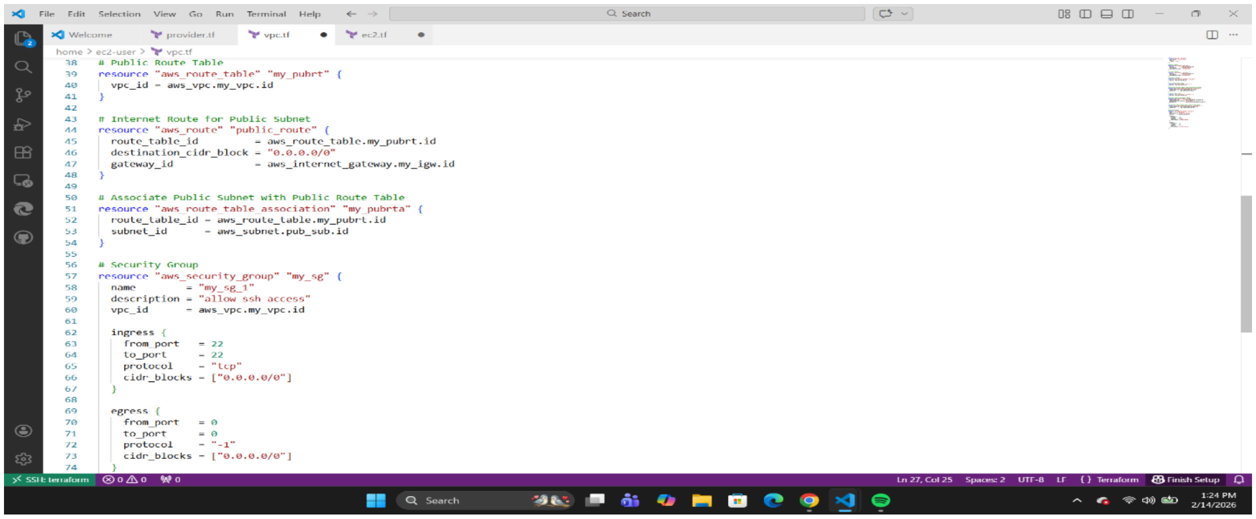
File Edit Selection View Go Run Terminal Help
Search
Welcome
provider.tf
vpc.tf
ec2.tf
home > ec2-user > vpc.tf
SSH: terraform
Ln 27, Col 25 Spaces: 2 UTF-8 LF Terraform Finish Setup
1:24 PM 2/14/2026

10.2 Public Route Table

```
resource "aws_route_table" "my_pubrt" {
  vpc_id = aws_vpc.my_vpc.id
  tags = {
    Name = "public-rt"
```

  }

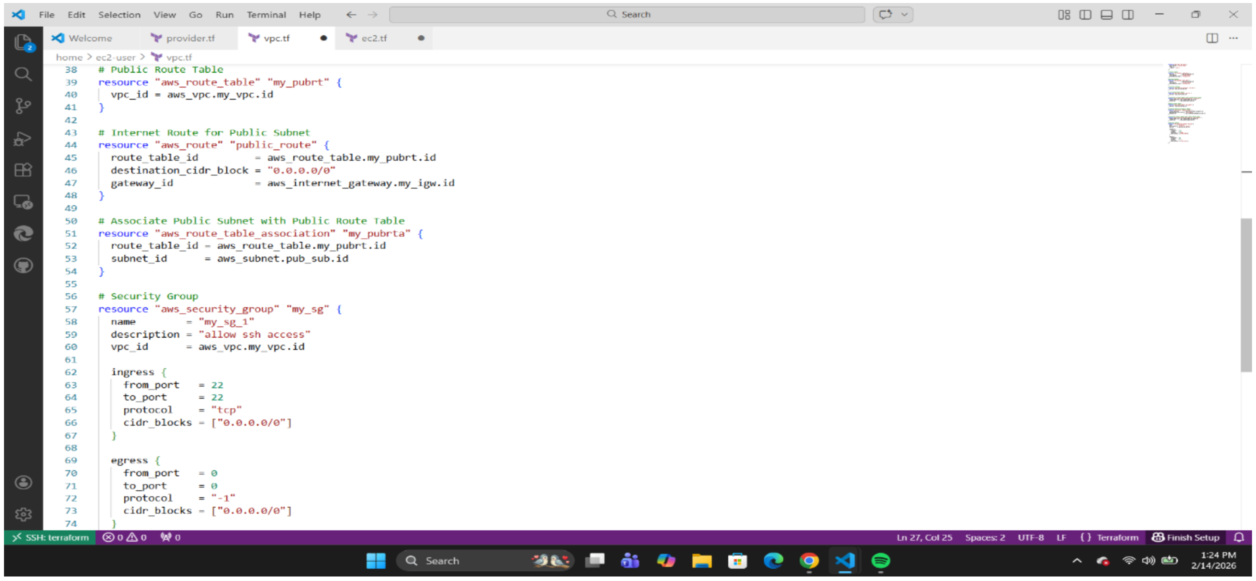
}



```
# Public Route Table
resource "aws_route_table" "my_pubrt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Internet Route for Public Subnet
resource "aws_route" "public_route" {
  route_table_id         = aws_route_table.my_pubrt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.my_igw.id
}

# Associate Public Subnet with Public Route Table
resource "aws_route_table_association" "my_pubrta" {
  route_table_id = aws_route_table.my_pubrt.id
  subnet_id      = aws_subnet.pub_sub.id
}

# Security Group
resource "aws_security_group" "my_sg" {
  name        = "my_sg_1"
  description = "allow ssh access"
  vpc_id      = aws_vpc.my_vpc.id

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

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

11. Public Internet Route

```
resource "aws_route" "public_route" {
  route_table_id         = aws_route_table.my_pubrt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.my_igw.id
}
```



```
# Public Route Table
resource "aws_route_table" "my_pubrt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Internet Route for Public Subnet
resource "aws_route" "public_route" {
  route_table_id         = aws_route_table.my_pubrt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.my_igw.id
}

# Associate Public Subnet with Public Route Table
resource "aws_route_table_association" "my_pubrta" {
  route_table_id = aws_route_table.my_pubrt.id
  subnet_id      = aws_subnet.pub_sub.id
}

# Security Group
resource "aws_security_group" "my_sg" {
  name        = "my_sg_1"
  description = "allow ssh access"
  vpc_id      = aws_vpc.my_vpc.id

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

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

Explanation:

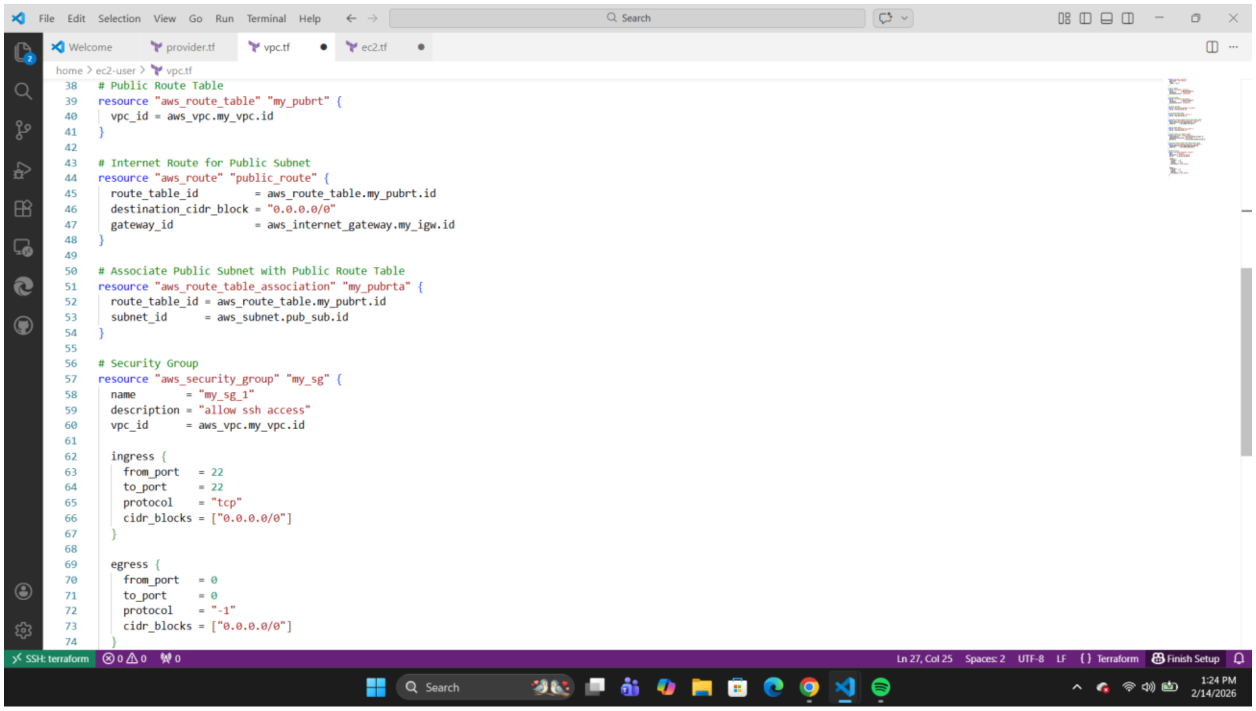
- 0.0.0.0/0 means internet access
- Attached to Internet Gateway

---

12. Route Table Association

12.1 Public Subnet Association

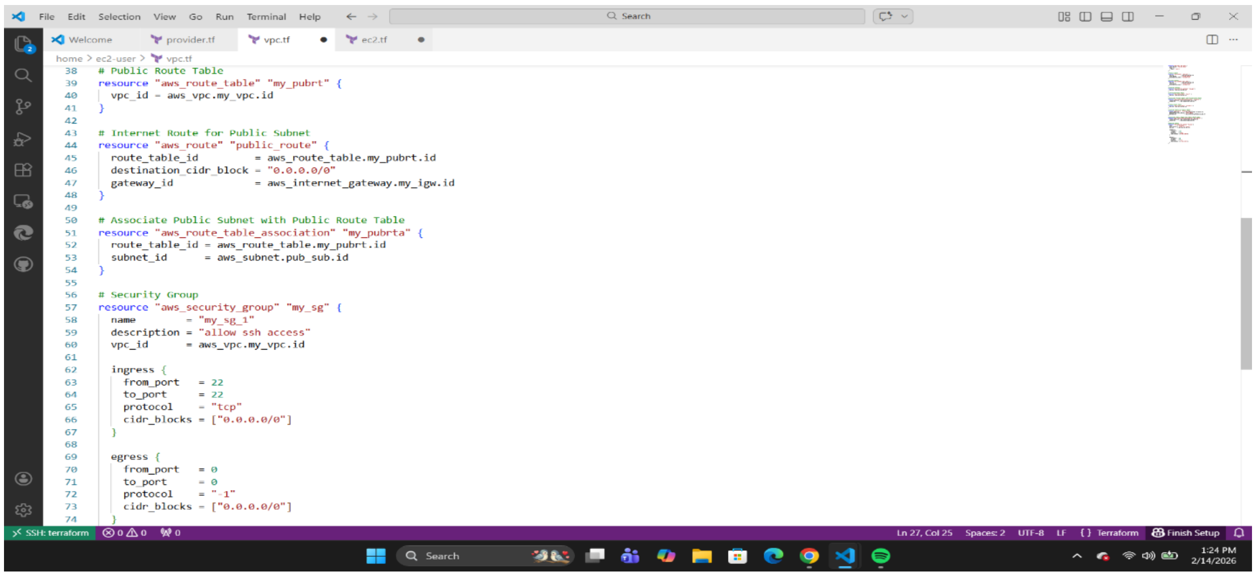
```
resource "aws_route_table_association" "my_pubrta" {
  route_table_id = aws_route_table.my_pubrt.id
  subnet_id      = aws_subnet.pub_sub.id
}
```



```
# Public Route Table
resource "aws_route_table" "my_pubrt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Internet Route for Public Subnet
resource "aws_route" "public_route" {
  route_table_id         = aws_route_table.my_pubrt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.my_igw.id
}

# Associate Public Subnet with Public Route Table
resource "aws_route_table_association" "my_pubrta" {
  route_table_id = aws_route_table.my_pubrt.id
  subnet_id      = aws_subnet.pub_sub.id
}

# Security Group
resource "aws_security_group" "my_sg" {
  name        = "my_sg_1"
  description = "allow ssh access"
  vpc_id      = aws_vpc.my_vpc.id

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

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

12.2 Private Subnet Association

```
resource "aws_route_table_association" "my_prta" {
  route_table_id = aws_route_table.my_rt.id
  subnet_id      = aws_subnet.pri_sub.id
}
```


File Edit Selection View Go Run Terminal Help
Welcome
provider.tf
vpc.tf
ec2.tf
home > ec2-user > vpc.tf
SSH: terraform
Ln 27, Col 25 Spaces: 2 UTF-8 LF Terraform Finish Setup
1:24 PM 2/14/2026

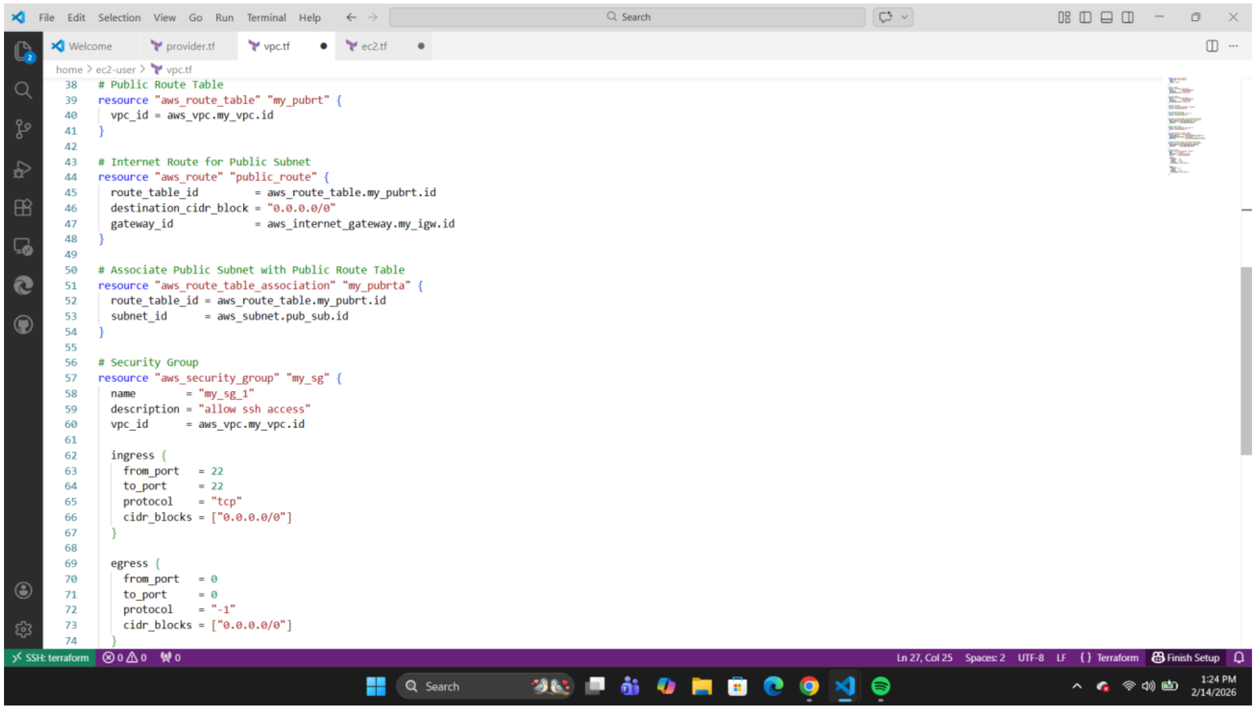
```
# Public Route Table
resource "aws_route_table" "my_pubrt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Internet Route for Public Subnet
resource "aws_route" "public_route" {
  route_table_id         = aws_route_table.my_pubrt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.my_igw.id
}

# Associate Public Subnet with Public Route Table
resource "aws_route_table_association" "my_pubrta" {
  route_table_id = aws_route_table.my_pubrt.id
  subnet_id      = aws_subnet.pub_sub.id
}

# Security Group
resource "aws_security_group" "my_sg" {
  name        = "my_sg_1"
  description = "allow ssh access"
  vpc_id      = aws_vpc.my_vpc.id

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

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

---

13. Security Group Configuration (Firewall)

```
resource "aws_security_group" "my_sg" {
  name        = "my_sg_1"
  description = "Allow SSH access"
  vpc_id      = aws_vpc.my_vpc.id
  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

}



```
# Public Route Table
resource "aws_route_table" "my_pubrt" {
  vpc_id = aws_vpc.my_vpc.id
}

# Internet Route for Public Subnet
resource "aws_route" "public_route" {
  route_table_id         = aws_route_table.my_pubrt.id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id             = aws_internet_gateway.my_igw.id
}

# Associate Public Subnet with Public Route Table
resource "aws_route_table_association" "my_pubrta" {
  route_table_id = aws_route_table.my_pubrt.id
  subnet_id      = aws_subnet.pub_sub.id
}

# Security Group
resource "aws_security_group" "my_sg" {
  name        = "my_sg_1"
  description = "allow ssh access"
  vpc_id      = aws_vpc.my_vpc.id

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

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

Explanation:

- Ingress: Allows SSH (Port 22)
- Egress: Allows all outgoing traffic

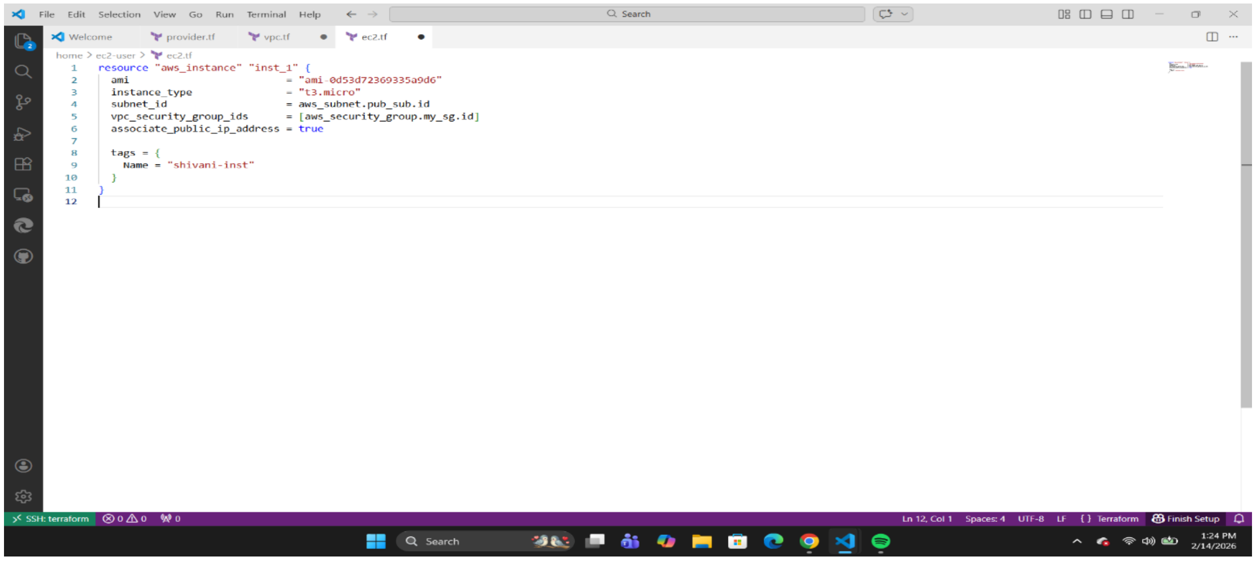
---

14. EC2 Instance Configuration (Final Step)

File: ec2.tf

```
resource "aws_instance" "inst_1" {
  ami                    = "ami-0d53d72369335a9d6"
  instance_type          = "t2.micro"
  subnet_id              = aws_subnet.pub_sub.id
  vpc_security_group_ids = [aws_security_group.my_sg.id]
  associate_public_ip_address = true
  tags = {
    Name = "terraform-ec2"
  }
```

}



```
resource "aws_instance" "inst_1" {
  ami                         = "ami-0d53d72369335a9d6"
  instance_type               = "t3.micro"
  subnet_id                   = aws_subnet.pub_sub.id
  vpc_security_group_ids      = [aws_security_group.my_sg.id]
  associate_public_ip_address = true

  tags = {
    Name = "shivani-inst"
  }
}
```

Important Notes:

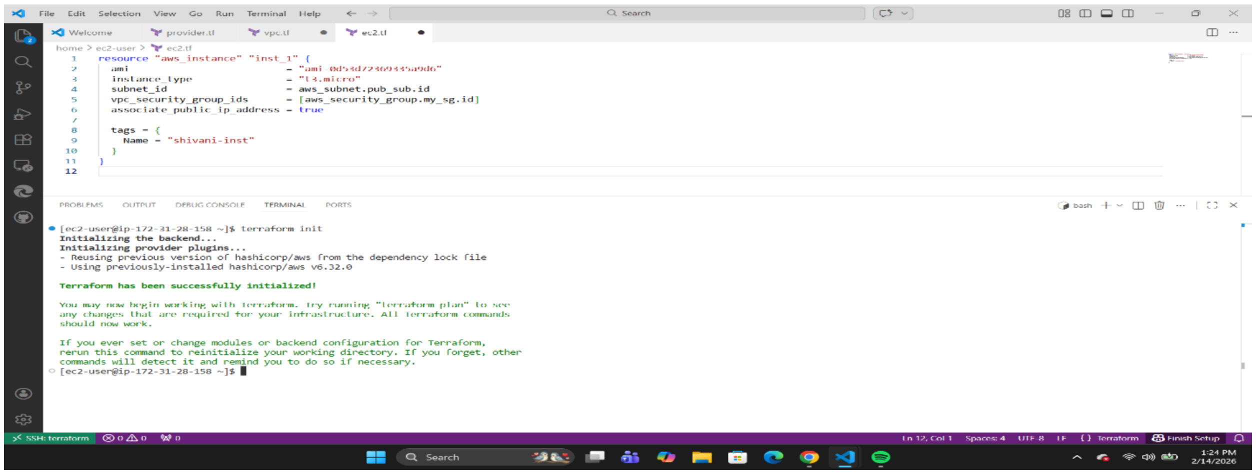
- t2.micro = Free Tier eligible
- Must use public subnet for public IP
- AMI must match region (us-west-1)

---

15. Complete Terraform Execution Commands (Step-by-Step)

Step 1: Initialize Terraform

terraform init



```
resource "aws_instance" "inst_1" {
  ami                         = "ami-0d53d72369335a9d6"
  instance_type               = "t3.micro"
  subnet_id                   = aws_subnet.pub_sub.id
  vpc_security_group_ids      = [aws_security_group.my_sg.id]
  associate_public_ip_address = true

  tags = {
    Name = "shivani-inst"
  }
}
```

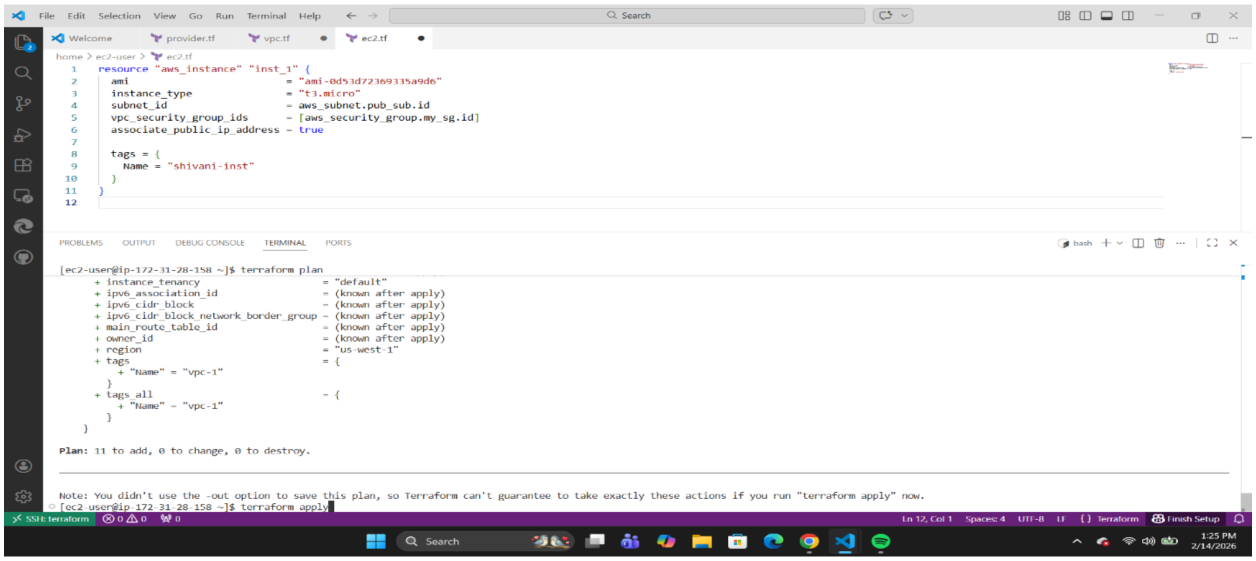
```
[ec2-user@ip-172-31-28-158 ~]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.32.0

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.
[ec2-user@ip-172-31-28-158 ~]$
```

Step 2: Validate Code

terraform validate

Step 3: Plan Infrastructure

terraform plan



```
resource "aws_instance" "inst_1" {
  ami                         = "ami-0d53d72369335a9d6"
  instance_type               = "t3.micro"
  subnet_id                   = aws_subnet.pub_sub.id
  vpc_security_group_ids      = [aws_security_group.my_sg.id]
  associate_public_ip_address = true

  tags = {
    Name = "shivani-inst"
  }
}
```

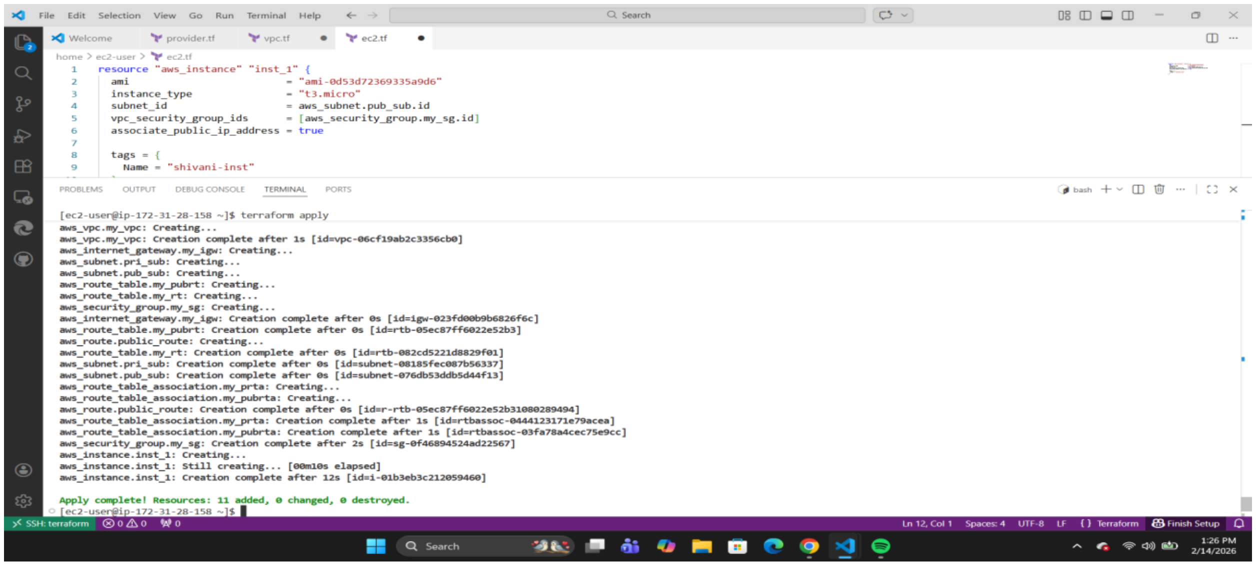
```
[ec2-user@ip-172-31-28-158 ~]$ terraform plan
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "us-west-1"
      + tags                                 = {
          + "Name" = "vpc-1"
        }
      + tags_all                             = {
          + "Name" = "vpc-1"
        }
    }

Plan: 11 to add, 0 to change, 0 to destroy.

─────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
[ec2-user@ip-172-31-28-158 ~]$ terraform apply
```

Step 4: Apply Configuration

terraform apply

Type:

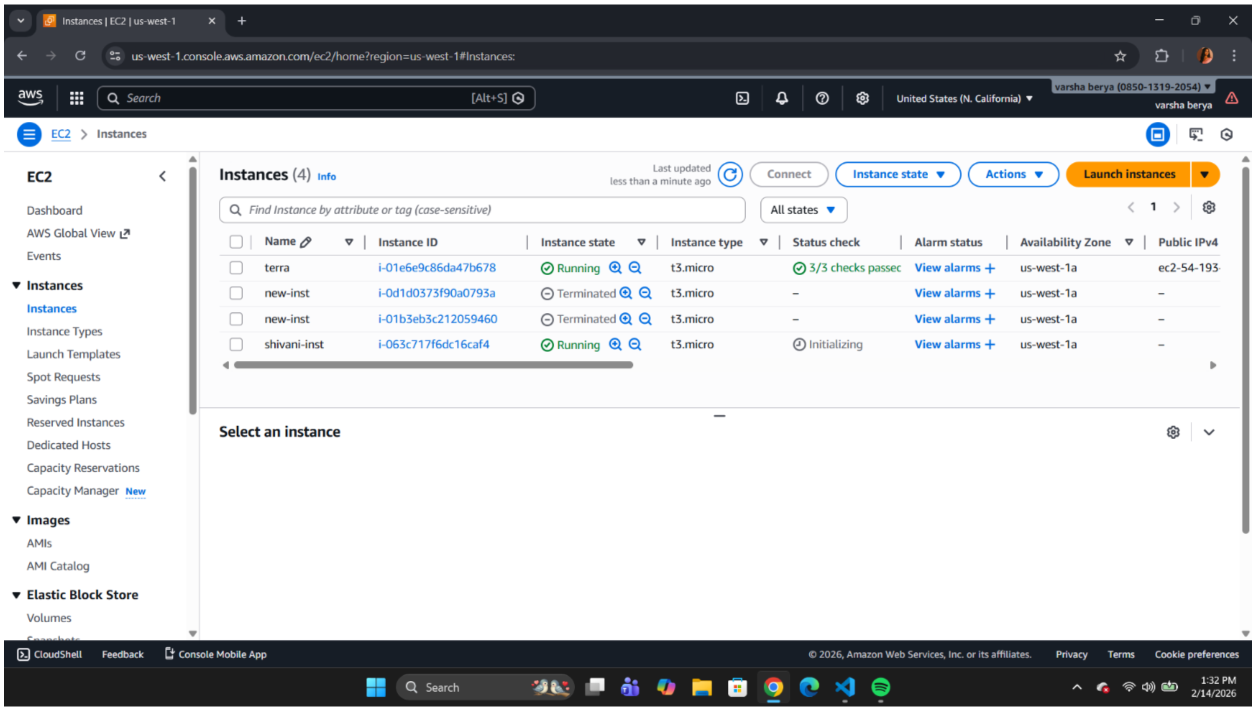
Yes



```
resource "aws_instance" "inst_1" {
  ami                         = "ami-0d53d72369335a9d6"
  instance_type               = "t3.micro"
  subnet_id                   = aws_subnet.pub_sub.id
  vpc_security_group_ids      = [aws_security_group.my_sg.id]
  associate_public_ip_address = true

  tags = {
    Name = "shivani-inst"
```

```
[ec2-user@ip-172-31-28-158 ~]$ terraform apply
aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Creation complete after 1s [id=vpc-06cf19ab2c3356cb0]
aws_internet_gateway.my_igw: Creating...
aws_subnet.pri_sub: Creating...
aws_subnet.pub_sub: Creating...
aws_route_table.my_pubrt: Creating...
aws_route_table.my_rt: Creating...
aws_security_group.my_sg: Creating...
aws_internet_gateway.my_igw: Creation complete after 0s [id=igw-023fd00b9b6826f6c]
aws_route_table.my_pubrt: Creation complete after 0s [id=rtb-05ec87ff6022e52b3]
aws_route.public_route: Creating...
aws_route_table.my_rt: Creation complete after 0s [id=rtb-082cd5221d8829f01]
aws_subnet.pri_sub: Creation complete after 0s [id=subnet-08185fec087b56337]
aws_subnet.pub_sub: Creation complete after 0s [id=subnet-076db53ddb5d44f13]
aws_route_table_association.my_prta: Creating...
aws_route_table_association.my_pubrta: Creating...
aws_route.public_route: Creation complete after 0s [id=r-rtb-05ec87ff6022e52b31080289494]
aws_route_table_association.my_prta: Creation complete after 1s [id=rtbassoc-0444123171e79acea]
aws_route_table_association.my_pubrta: Creation complete after 1s [id=rtbassoc-03fa78a4cec75e9cc]
aws_security_group.my_sg: Creation complete after 2s [id=sg-0f46894524ad22567]
aws_instance.inst_1: Creating...
aws_instance.inst_1: Still creating... [00m10s elapsed]
aws_instance.inst_1: Creation complete after 12s [id=i-01b3eb3c212059460]

Apply complete! Resources: 11 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-28-158 ~]$
```

Step 5: Check Created Resources



Instances | EC2 | us-west-1

us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#Instances:

EC2 > Instances

United States (N. California) — varsha berya (0850-1319-2054)

Instances (4) Info

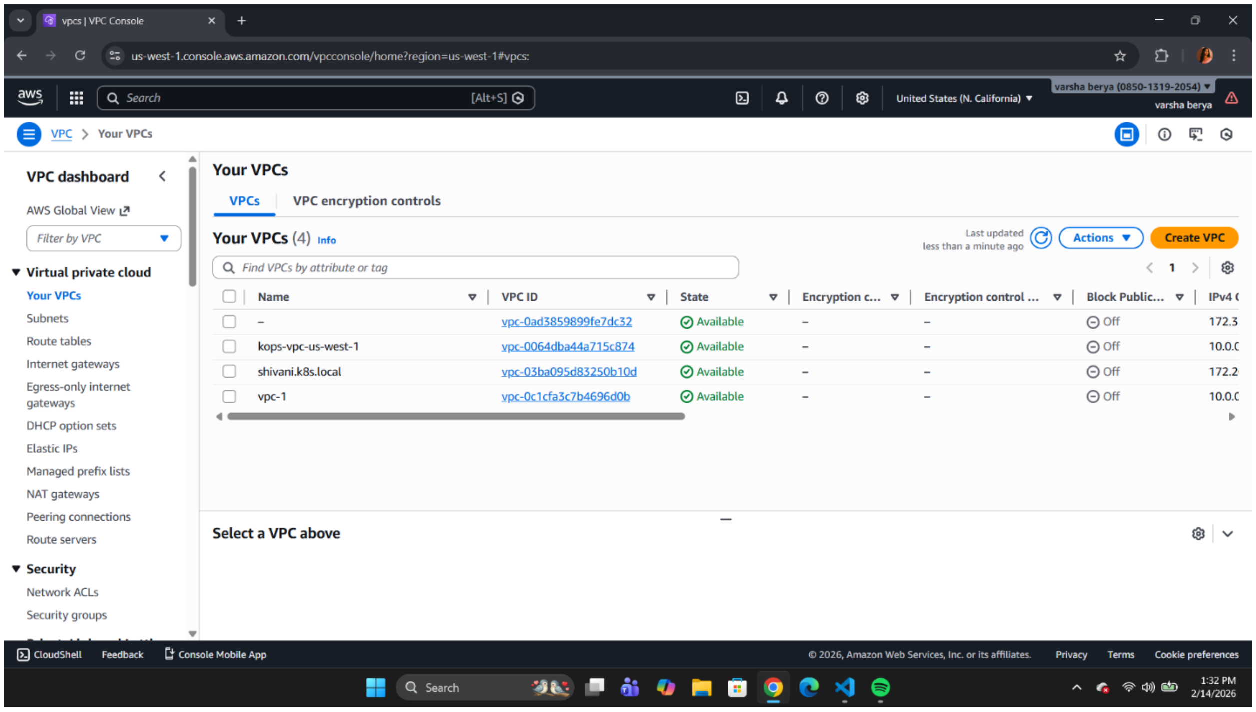
Last updated less than a minute ago

Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) — All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
terra	i-01e6e9c86da47b678	Running	t3.micro	3/3 checks passec	View alarms +	us-west-1a	ec2-54-193-
new-inst	i-0d1d0373f90a0793a	Terminated	t3.micro	–	View alarms +	us-west-1a	–
new-inst	i-01b3eb3c212059460	Terminated	t3.micro	–	View alarms +	us-west-1a	–
shivani-inst	i-063c717f6dc16caf4	Running	t3.micro	Initializing	View alarms +	us-west-1a	–

Select an instance



vpcs | VPC Console

us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#vpcs:

VPC > Your VPCs

United States (N. California) — varsha berya (0850-1319-2054)

Your VPCs

VPCs | VPC encryption controls

Your VPCs (4) Info

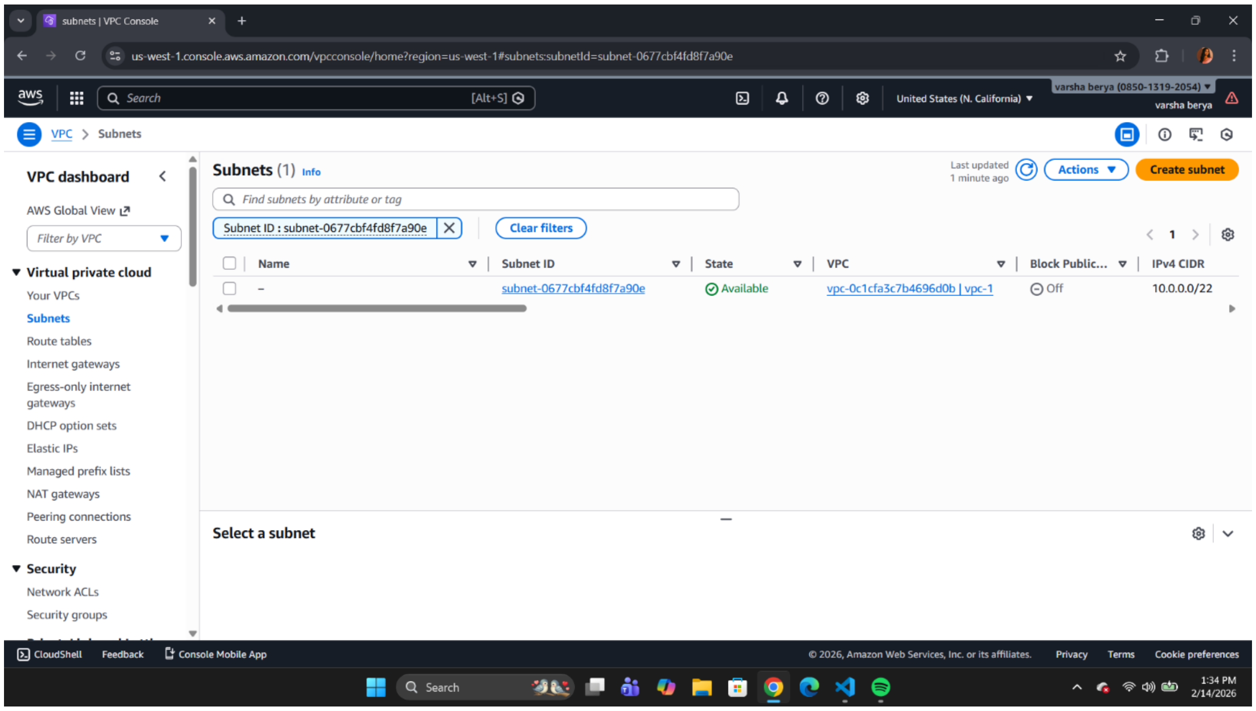
Last updated less than a minute ago

Actions | Create VPC

Find VPCs by attribute or tag

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 (
–	vpc-0ad3859899fe7dc32	Available	–	–	Off	172.3
kops-vpc-us-west-1	vpc-0064dba44a715c874	Available	–	–	Off	10.0.0
shivani.k8s.local	vpc-03ba095d83250b10d	Available	–	–	Off	172.2
vpc-1	vpc-0c1cfa3c7b4696d0b	Available	–	–	Off	10.0.0

Select a VPC above



subnets | VPC Console

us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#subnets:subnetId=subnet-0677cbf4fd8f7a90e

VPC > Subnets

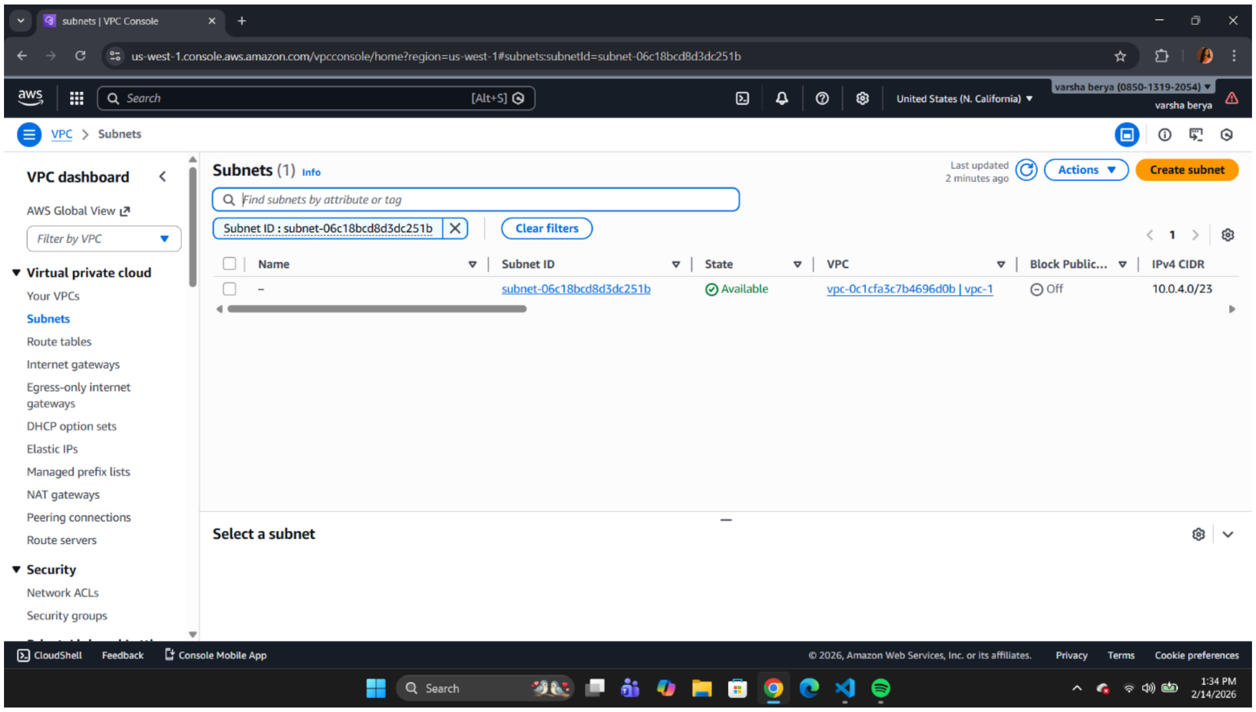
Subnets (1) Info

Last updated 1 minute ago

Subnet ID : subnet-0677cbf4fd8f7a90e

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
–	subnet-0677cbf4fd8f7a90e	Available	vpc-0c1cfa3c7b4696d0b | vpc-1	Off	10.0.0.0/22

Select a subnet



subnets | VPC Console

us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#subnets:subnetId=subnet-06c18bcd8d3dc251b

VPC > Subnets

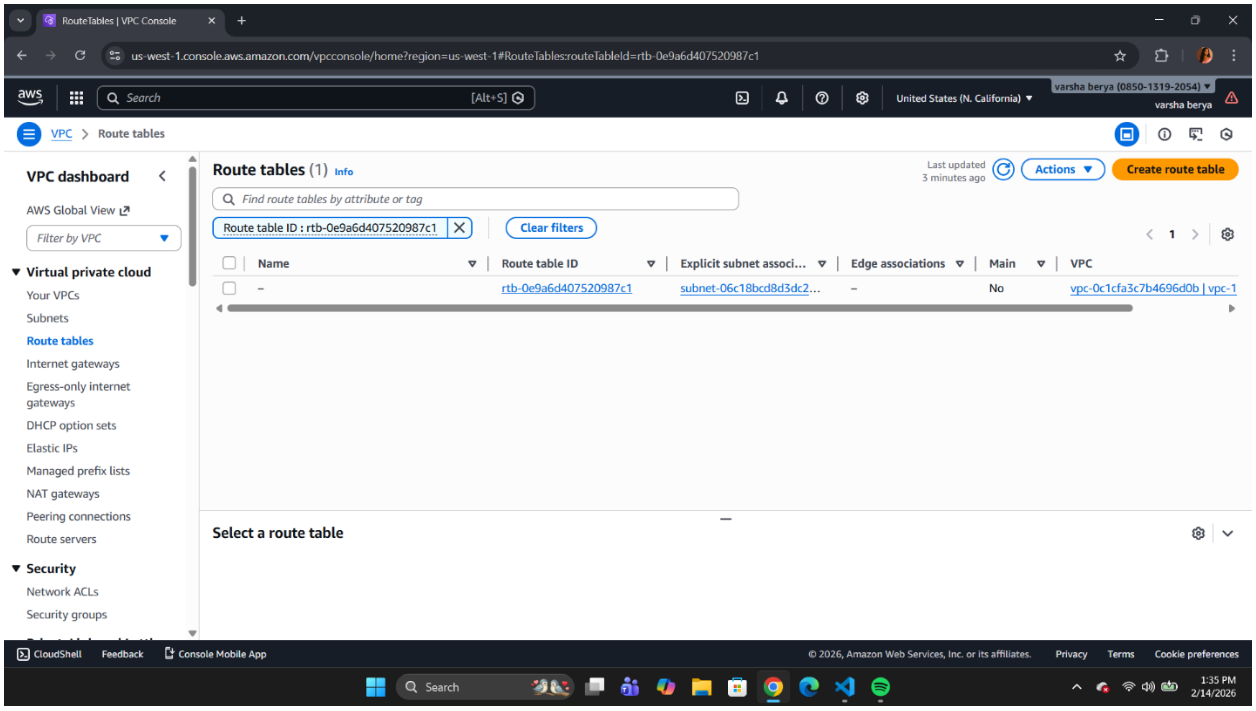
Subnets (1) Info

Last updated 2 minutes ago

Subnet ID : subnet-06c18bcd8d3dc251b

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
–	subnet-06c18bcd8d3dc251b	Available	vpc-0c1cfa3c7b4696d0b | vpc-1	Off	10.0.4.0/23

Select a subnet


RouteTables | VPC Console

us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#RouteTables:routeTableId=rtb-0e9a6d407520987c1

United States (N. California) · varsha berya (0850-1319-2054) · varsha berya

VPC > Route tables

VPC dashboard
AWS Global View
Filter by VPC
Virtual private cloud: Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers
Security: Network ACLs, Security groups

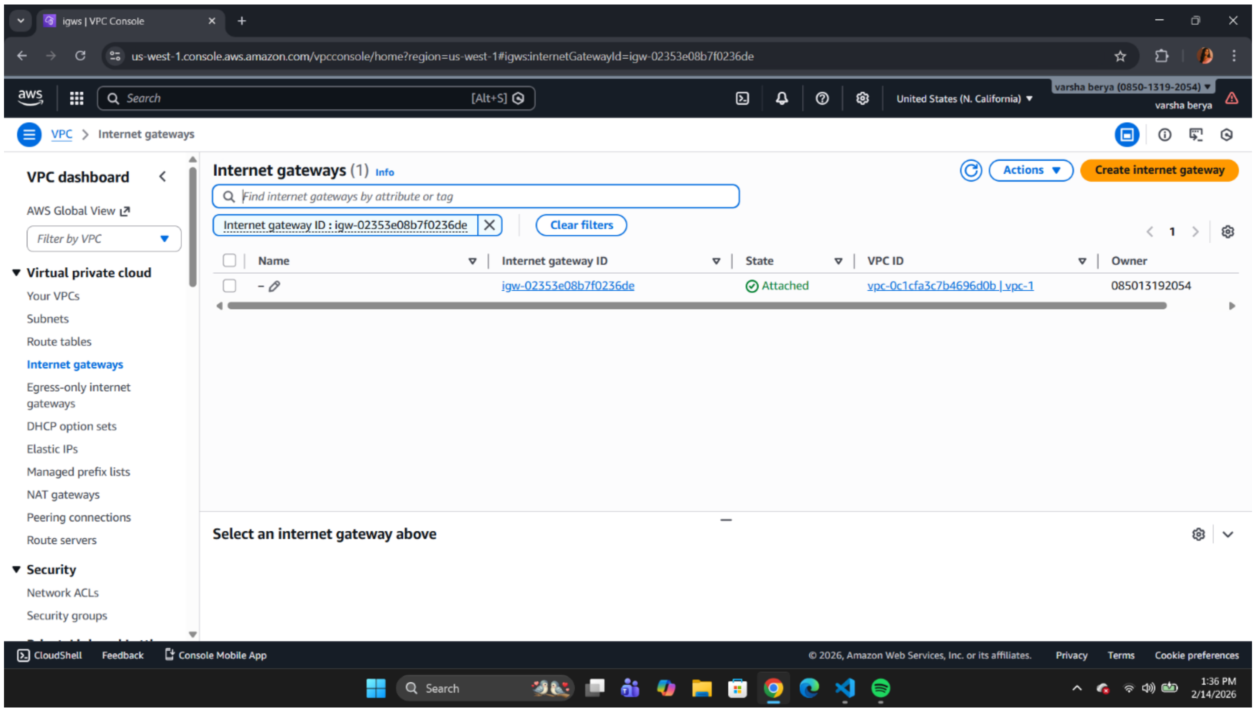
Route tables (1) Info
Last updated 3 minutes ago · Actions · Create route table
Find route tables by attribute or tag
Route table ID : rtb-0e9a6d407520987c1 · Clear filters

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
–	rtb-0e9a6d407520987c1	subnet-06c18bcd8d3dc2...	–	No	vpc-0c1cfa3c7b4696d0b | vpc-1

Select a route table

CloudShell · Feedback · Console Mobile App · © 2026, Amazon Web Services, Inc. or its affiliates. · Privacy · Terms · Cookie preferences

1:35 PM 2/14/2026


igws | VPC Console

us-west-1.console.aws.amazon.com/vpcconsole/home?region=us-west-1#igws:internetGatewayId=igw-02353e08b7f0236de

United States (N. California) · varsha berya (0850-1319-2054) · varsha berya

VPC > Internet gateways

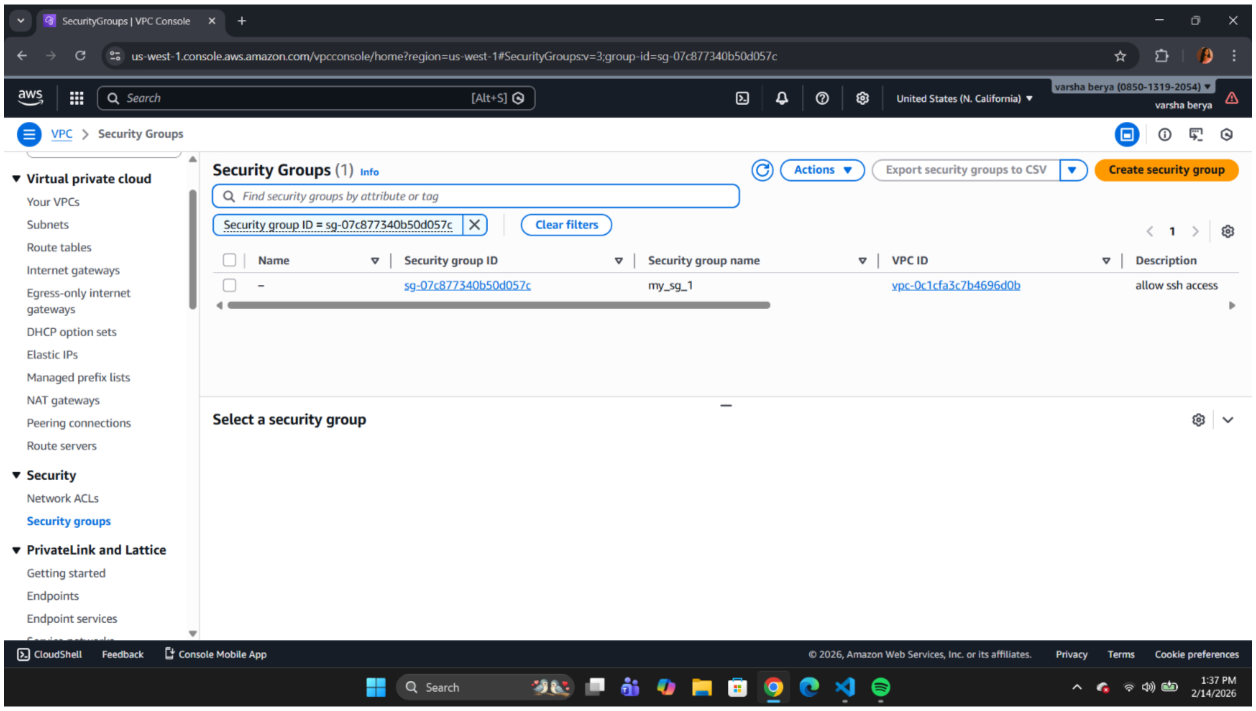
Internet gateways (1) Info
Actions · Create internet gateway
Find internet gateways by attribute or tag
Internet gateway ID : igw-02353e08b7f0236de · Clear filters

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-02353e08b7f0236de	Attached	vpc-0c1cfa3c7b4696d0b | vpc-1	085013192054

Select an internet gateway above

CloudShell · Feedback · Console Mobile App · © 2026, Amazon Web Services, Inc. or its affiliates. · Privacy · Terms · Cookie preferences

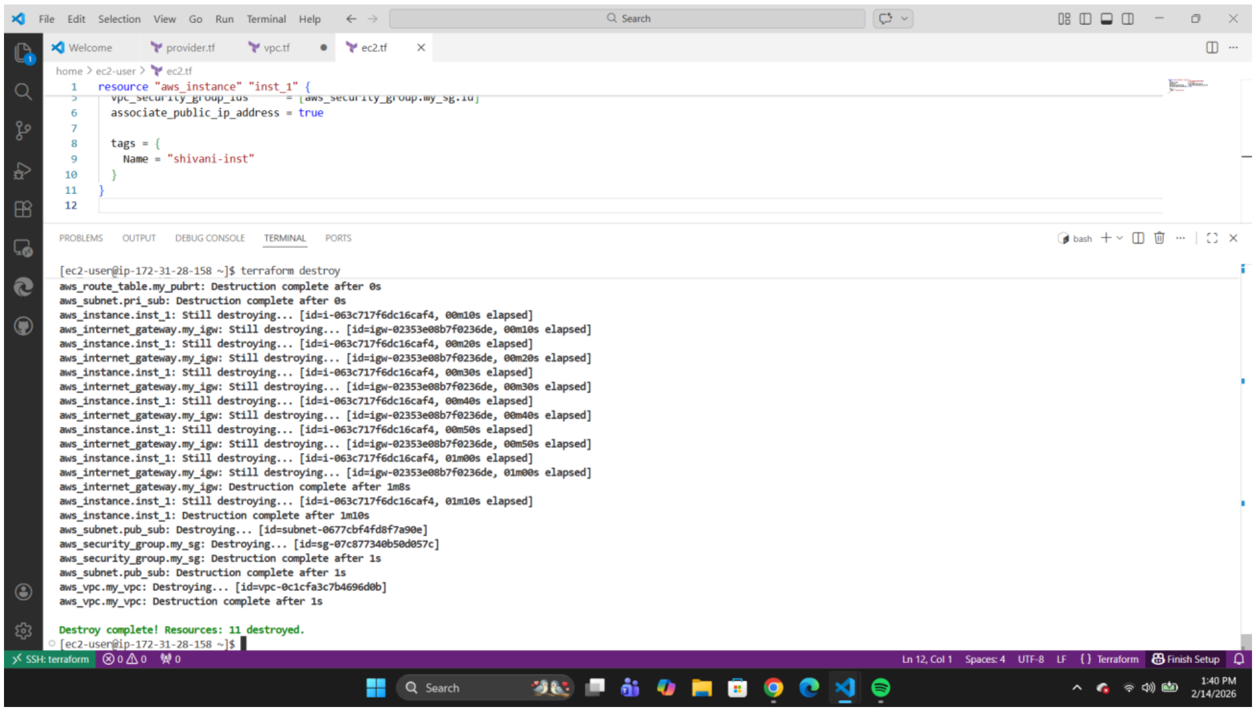
1:36 PM 2/14/2026


VPC > Security Groups
Security Groups (1) Info
Actions
Export security groups to CSV
Create security group
Find security groups by attribute or tag
Security group ID = sg-07c877340b50d057c
Clear filters
Select a security group

Name	Security group ID	Security group name	VPC ID	Description
–	sg-07c877340b50d057c	my_sg_1	vpc-0c1cfa3c7b4696d0b	allow ssh access

15. Destroy All Resources (Optional Cleanup)

terraform destroy



```
resource "aws_instance" "inst_1" {
  associate_public_ip_address = true

  tags = {
    Name = "shivani-inst"
  }
}
```

```
[ec2-user@ip-172-31-28-158 ~]$ terraform destroy
aws_route_table.my_pubrt: Destruction complete after 0s
aws_subnet.pri_sub: Destruction complete after 0s
aws_instance.inst_1: Still destroying... [id=i-063c717f6dc16caf4, 00m10s elapsed]
aws_internet_gateway.my_igw: Still destroying... [id=igw-02353e08b7f0236de, 00m10s elapsed]
aws_instance.inst_1: Still destroying... [id=i-063c717f6dc16caf4, 00m20s elapsed]
aws_internet_gateway.my_igw: Still destroying... [id=igw-02353e08b7f0236de, 00m20s elapsed]
aws_instance.inst_1: Still destroying... [id=i-063c717f6dc16caf4, 00m30s elapsed]
aws_internet_gateway.my_igw: Still destroying... [id=igw-02353e08b7f0236de, 00m30s elapsed]
aws_instance.inst_1: Still destroying... [id=i-063c717f6dc16caf4, 00m40s elapsed]
aws_internet_gateway.my_igw: Still destroying... [id=igw-02353e08b7f0236de, 00m40s elapsed]
aws_instance.inst_1: Still destroying... [id=i-063c717f6dc16caf4, 00m50s elapsed]
aws_internet_gateway.my_igw: Still destroying... [id=igw-02353e08b7f0236de, 00m50s elapsed]
aws_instance.inst_1: Still destroying... [id=i-063c717f6dc16caf4, 01m00s elapsed]
aws_internet_gateway.my_igw: Still destroying... [id=igw-02353e08b7f0236de, 01m00s elapsed]
aws_internet_gateway.my_igw: Destruction complete after 1m8s
aws_instance.inst_1: Still destroying... [id=i-063c717f6dc16caf4, 01m10s elapsed]
aws_instance.inst_1: Destruction complete after 1m10s
aws_subnet.pub_sub: Destroying... [id=subnet-0677cbf4fd8f7a90e]
aws_security_group.my_sg: Destroying... [id=sg-07c877340b50d057c]
aws_security_group.my_sg: Destruction complete after 1s
aws_subnet.pub_sub: Destruction complete after 1s
aws_vpc.my_vpc: Destroying... [id=vpc-0c1cfa3c7b4696d0b]
aws_vpc.my_vpc: Destruction complete after 1s

Destroy complete! Resources: 11 destroyed.
[ec2-user@ip-172-31-28-158 ~]$
```

Purpose:

- Avoid AWS billing
- Clean environment after project

---

16. Best Practices for Academic & DevOps Projects

- Use Free Tier resources
- Keep code modular (separate tf files)
- Add tags to all resources
- Use correct region and AZ
- Always run terraform validate before apply
- Store code in GitHub for version control

---

17. Final Conclusion

This project demonstrates a complete real-world AWS infrastructure deployment using Terraform. It includes networking, security, routing, and compute services in an automated and scalable way. By fixing errors like undeclared resources, region mismatch, availability zone issues, and free-tier limitations, the infrastructure becomes stable, reproducible, and suitable for DevOps learning, cloud labs project.