# 🧠 **Internship Task Documentation**

### Task 2: Terraform to Create EC2 Infrastructure

**Name:** Ujjawal Rawat  
**Date:** 4-10-2025  
**Tool Used:** AWS ,Terraform & Vscode with AI

## 🎯 ****Objective****

The goal of this task is to **automate the creation of an AWS EC2 instance** using Terraform.  
Once created, the instance should:

* Allow access on **port 22 (SSH)** and **port 80 (HTTP)**.
* **Automatically install Docker** using user\_data.

## 🏗️ ****Architecture Overview****

When Terraform is executed:

1. It connects to AWS via your access keys.
2. Creates:
   * A **Security Group** (for ports 22 and 80).
   * An **EC2 Instance** (Ubuntu).
3. Executes a **user\_data script** to install Docker automatically.

## ⚙️ ****Steps to Implement****

### **Step 1: Install Terraform**

# For Ubuntu

sudo apt update

sudo apt install -y wget unzip

# Download Terraform

wget https://releases.hashicorp.com/terraform/1.9.0/terraform\_1.9.0\_linux\_amd64.zip

# Extract and move binary

unzip terraform\_1.9.0\_linux\_amd64.zip

sudo mv terraform /usr/local/bin/

# Verify installation

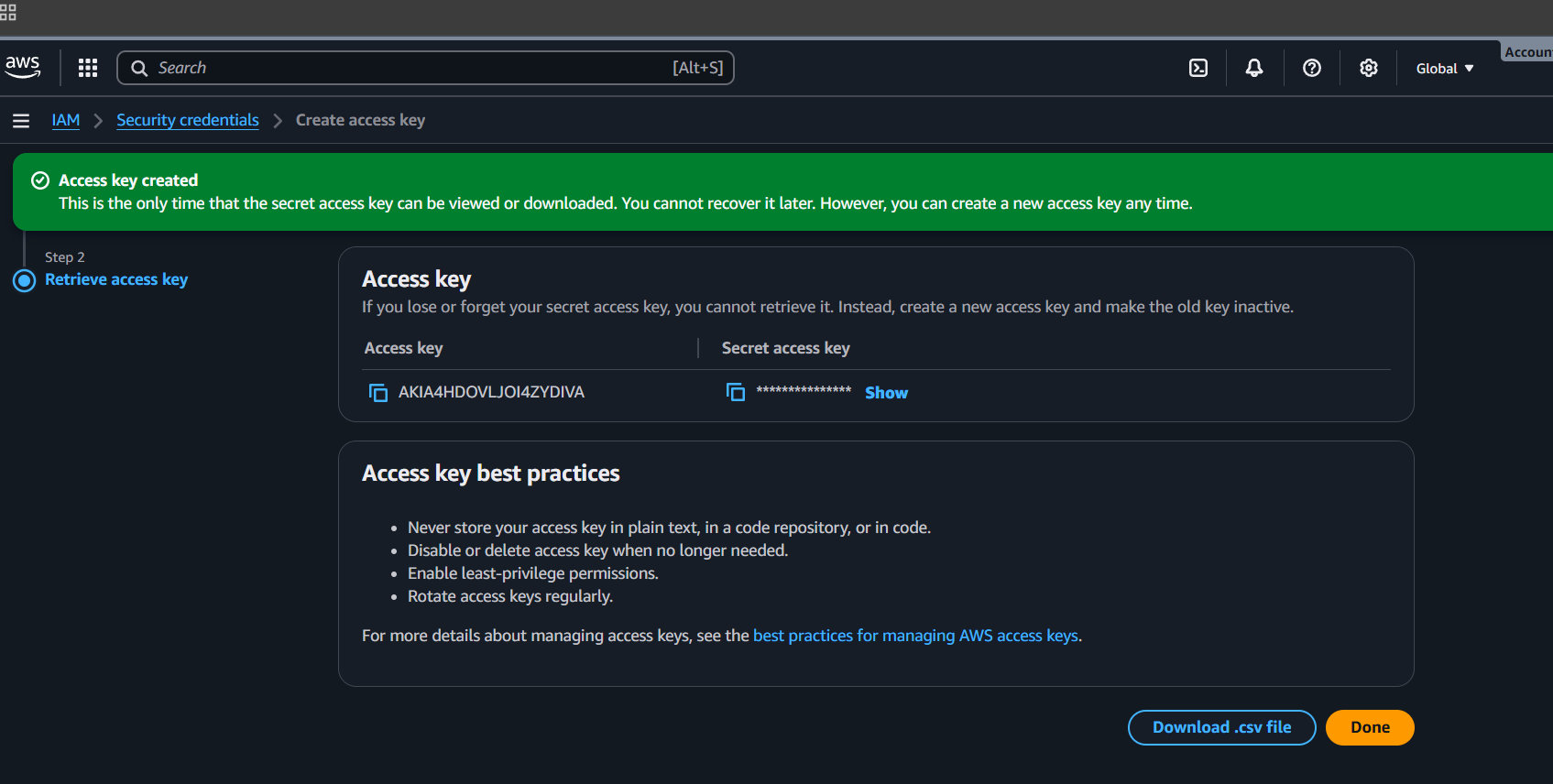
terraform -version

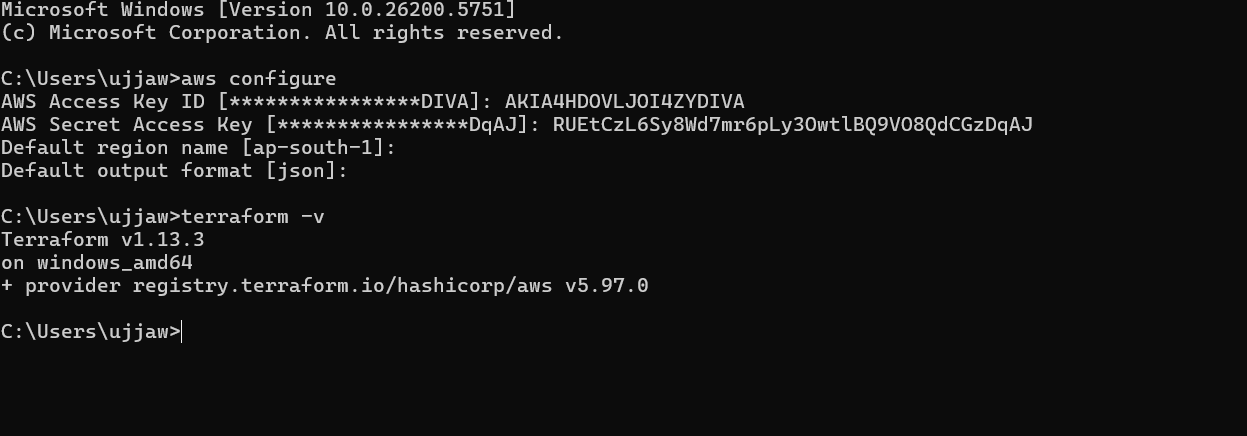
### **Step 2: Configure AWS CLI**

aws configure

Enter your:

* AWS Access Key
* AWS Secret Key
* Region (e.g., ap-south-1)
* Output format (e.g., json)





**CREATE THE main.tf FILE**

**task main.tf**

Step 1: Define the AWS Provider

provider "aws" {

region = "ap-south-1" # Mumbai region

}

# Step 2: Create a Security Group (Allow SSH and HTTP)

resource "aws\_security\_group" "web\_sg" {

name = "web-sg"

description = "Allow SSH (22) and HTTP (80) access"

ingress {

description = "Allow SSH"

from\_port = 22

to\_port = 22

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

ingress {

description = "Allow HTTP"

from\_port = 80

to\_port = 80

protocol = "tcp"

cidr\_blocks = ["0.0.0.0/0"]

}

egress {

description = "Allow all outbound traffic"

from\_port = 0

to\_port = 0

protocol = "-1"

cidr\_blocks = ["0.0.0.0/0"]

}

tags = {

Name = "WebSecurityGroup"

}

}

# Step 3: Launch EC2 Instance and Install Docker Automatically

resource "aws\_instance" "web" {

ami = "ami-02d26659fd82cf299" # Ubuntu 22.04 LTS (ap-south-1)

instance\_type = "t2.micro"

key\_name = "your-keypair-name" # Replace with your own AWS key pair name

vpc\_security\_group\_ids = [aws\_security\_group.web\_sg.id]

# user\_data installs Docker at instance startup

user\_data = <<-EOF

#!/bin/bash

apt update -y

apt install -y docker.io

systemctl start docker

systemctl enable docker

docker --version > /home/ubuntu/docker\_version.txt

EOF

tags = {

Name = "Terraform-EC2-Docker"

}

}

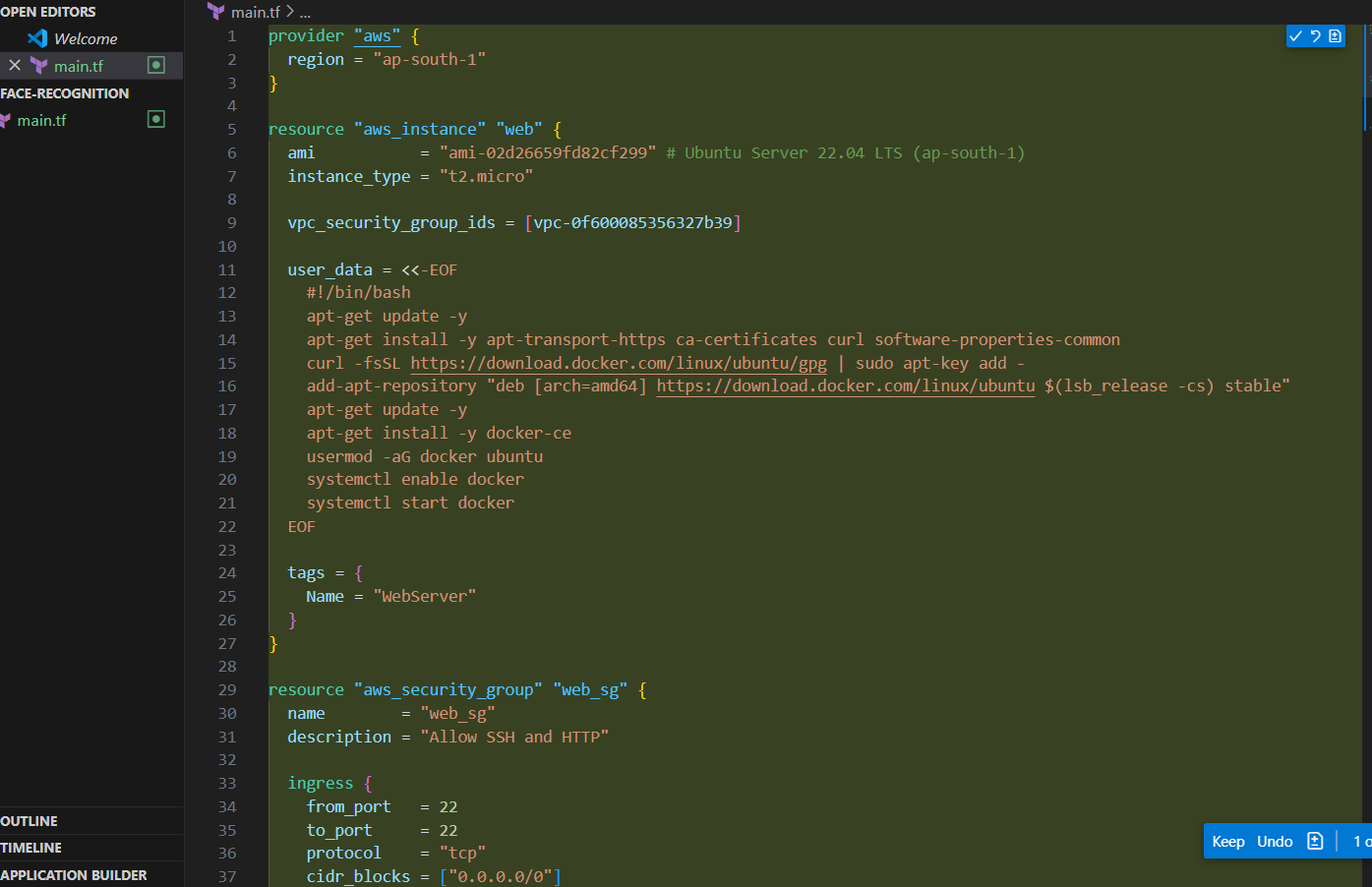
# Step 4: Output the EC2 Public IP

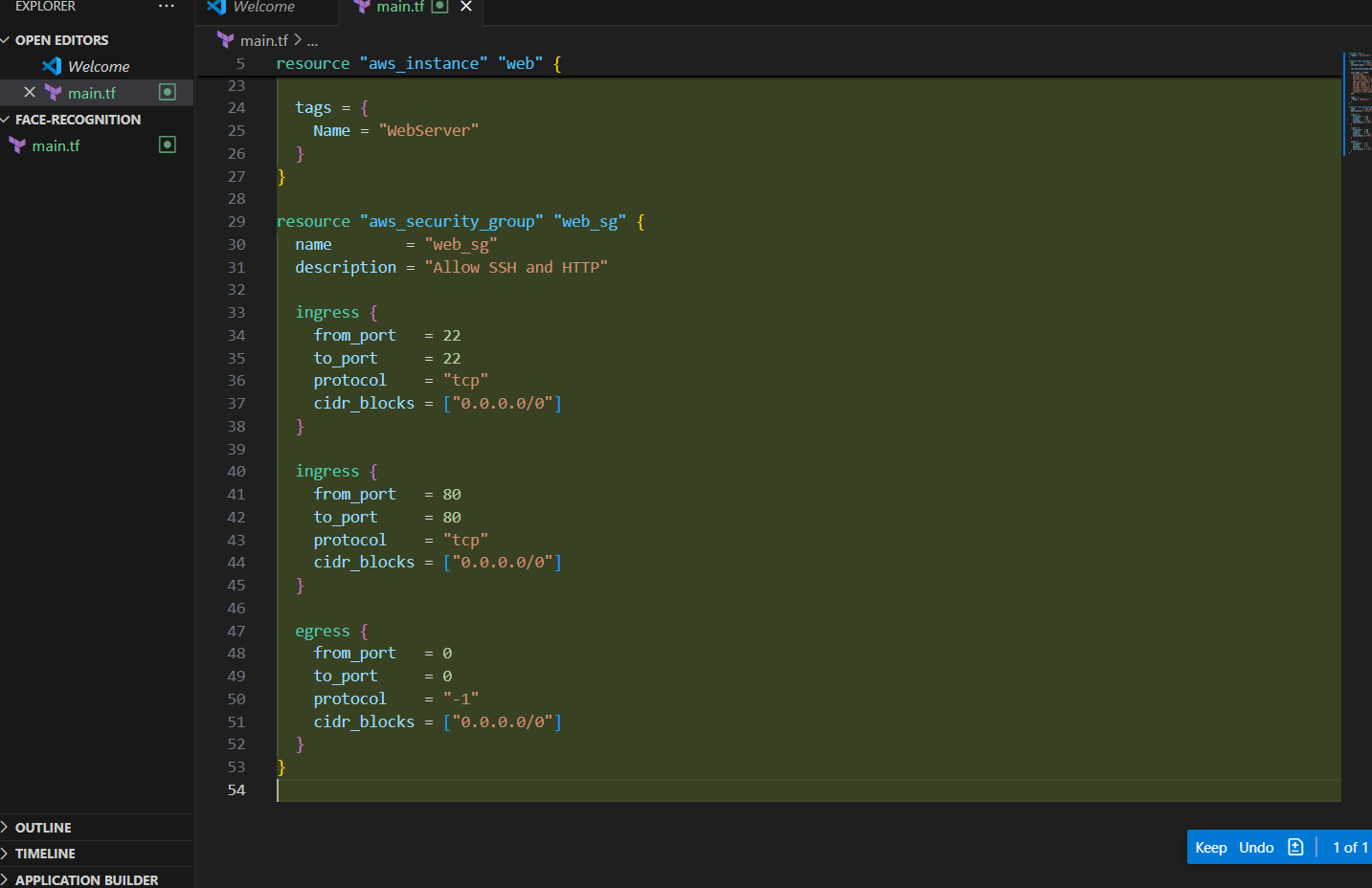
output "ec2\_public\_ip" {

description = "Public IP of EC2 instance"

value = aws\_instance.web.public\_ip

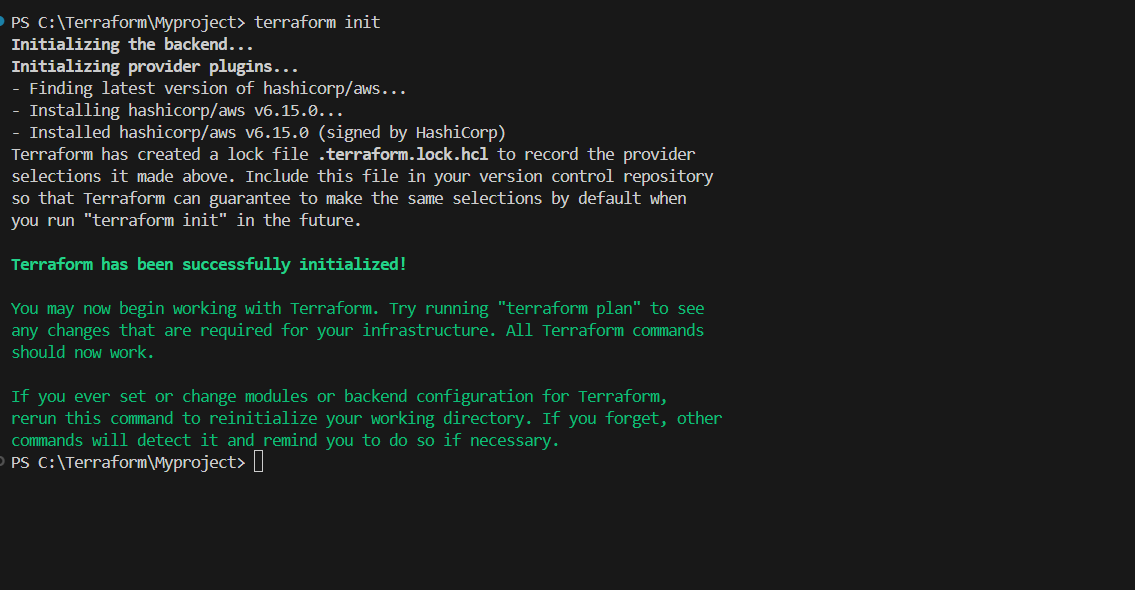
}





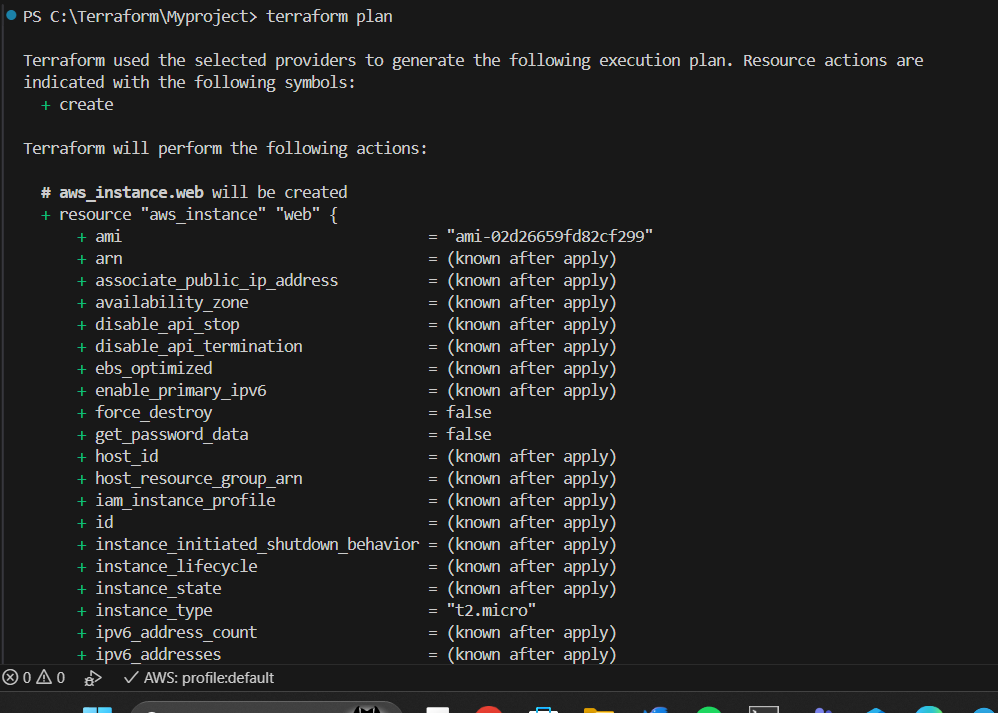
## 🧾 ****How to Run****

**Initialize Terraform**

terraform init

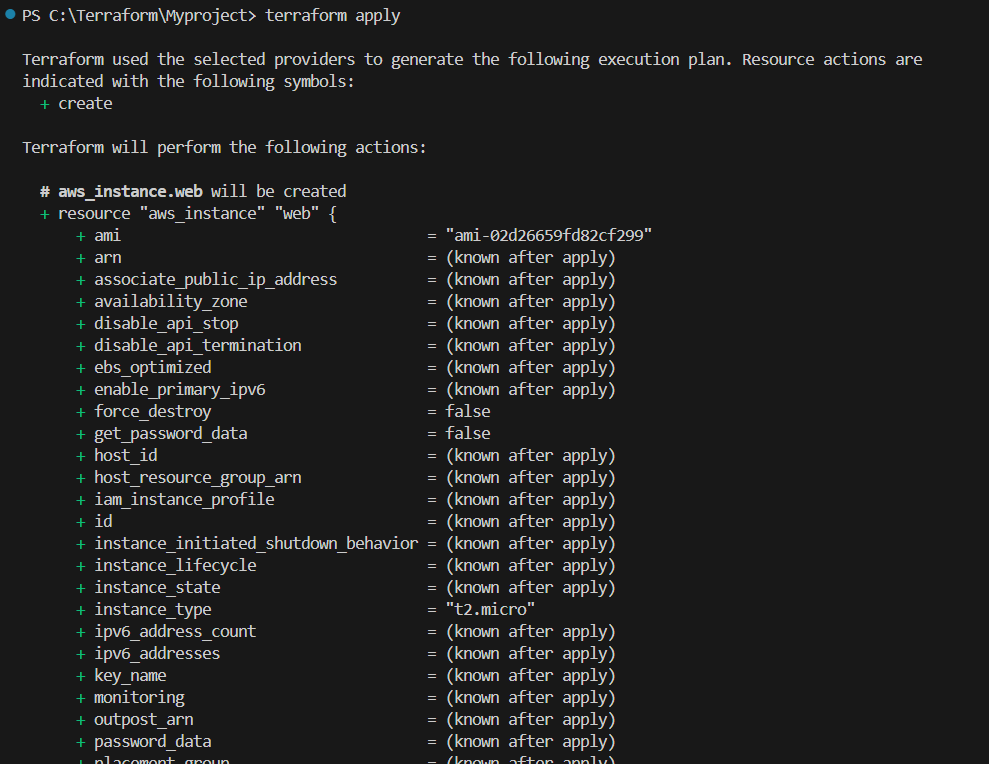
**Plan Resources**

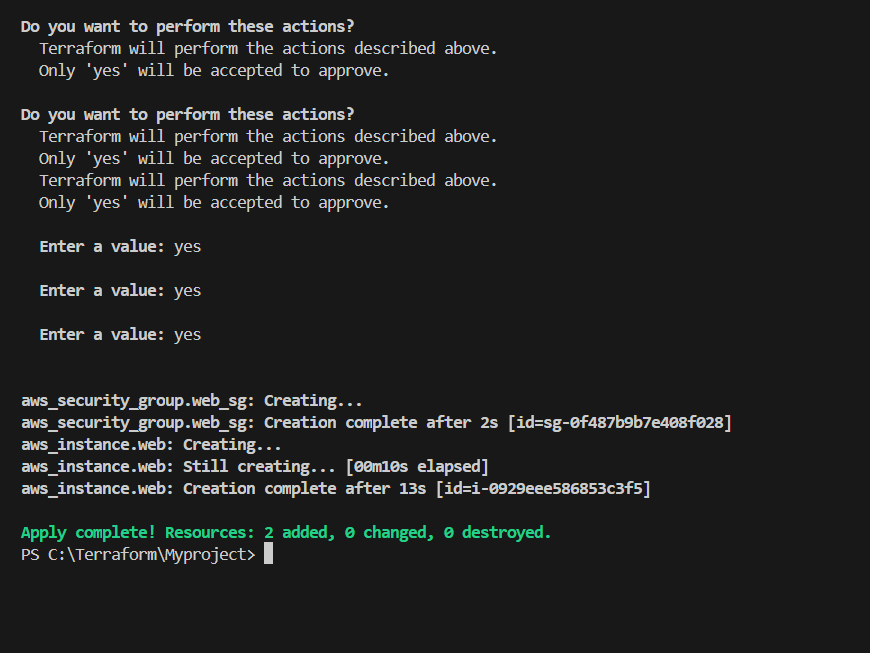
terraform plan



**Apply Configuration**

terraform apply





## ✅ ****After Apply****

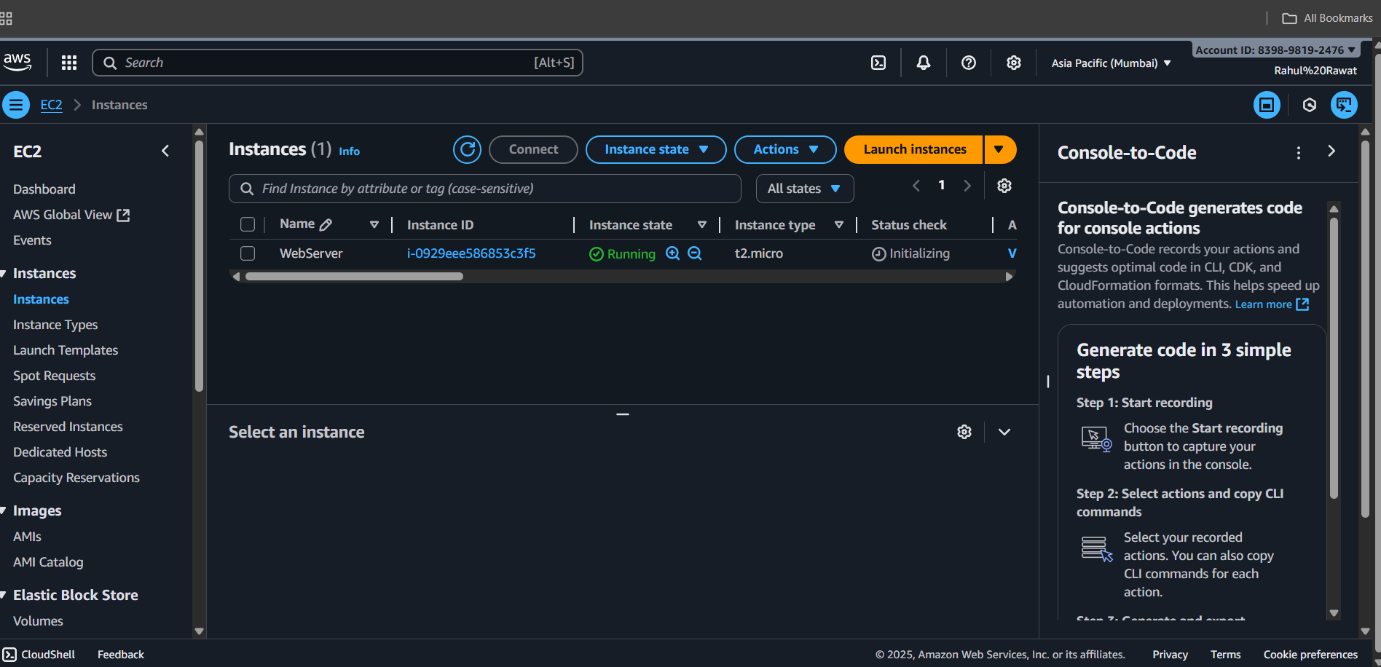
Terraform will display your **EC2 Public IP** like this:

Outputs:

ec2\_public\_ip = "3.108.xxx.xxx"

Use that IP to SSH:

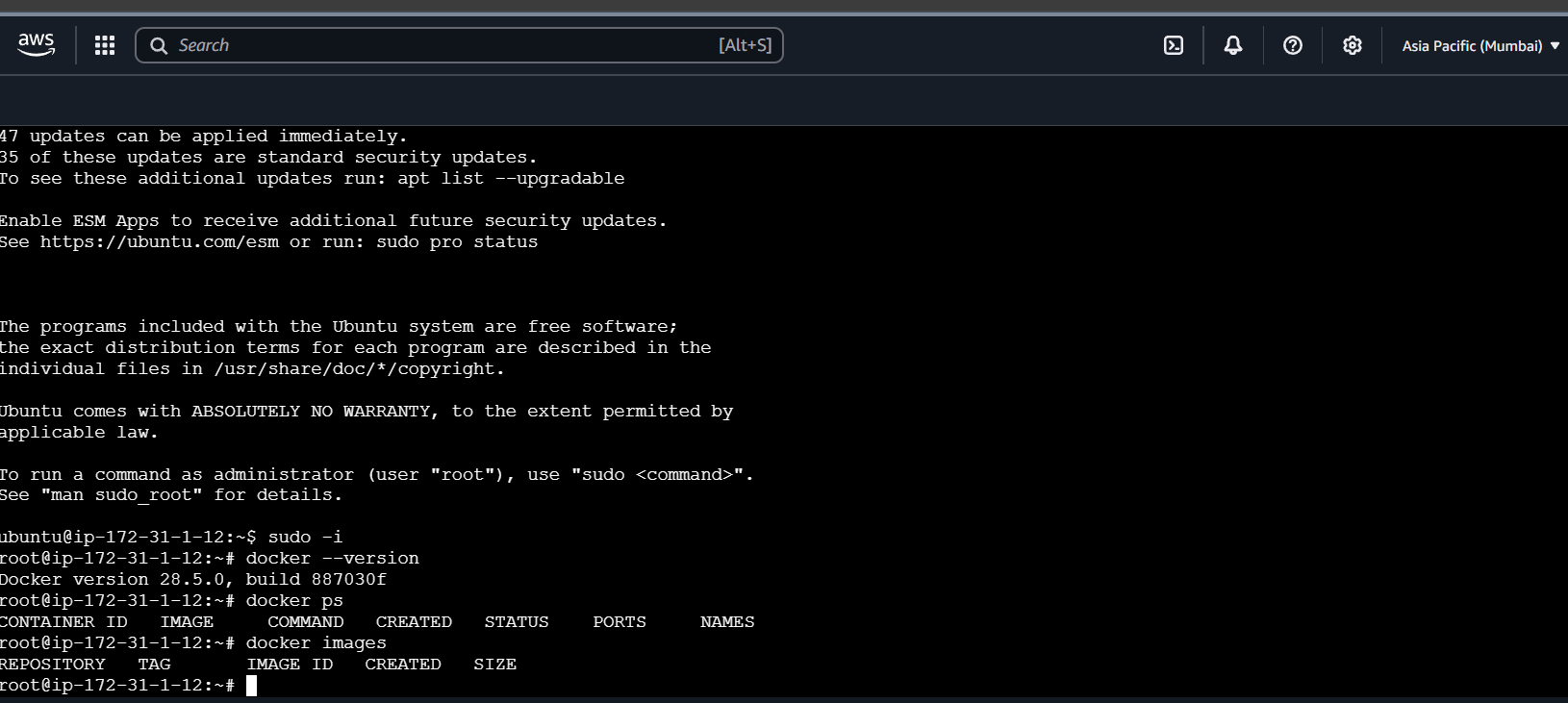
ssh -i "your-key.pem" ubuntu@<EC2\_PUBLIC\_IP>



## 🐳 ****Verify Docker Installation****

Inside your EC2:

docker --version



## 💡 ****Summary (for your PPT)****

| **Step** | **Action** | **Description** |
| --- | --- | --- |
| 1️⃣ | Provider | AWS Region = ap-south-1 |
| 2️⃣ | Security Group | Opens ports 22 & 80 |
| 3️⃣ | EC2 Instance | Ubuntu 22.04 with Docker installed |
| 4️⃣ | Output | Displays EC2 Public IP after deployment |