**SHIKHA**

**ASSIGNMENT 1**

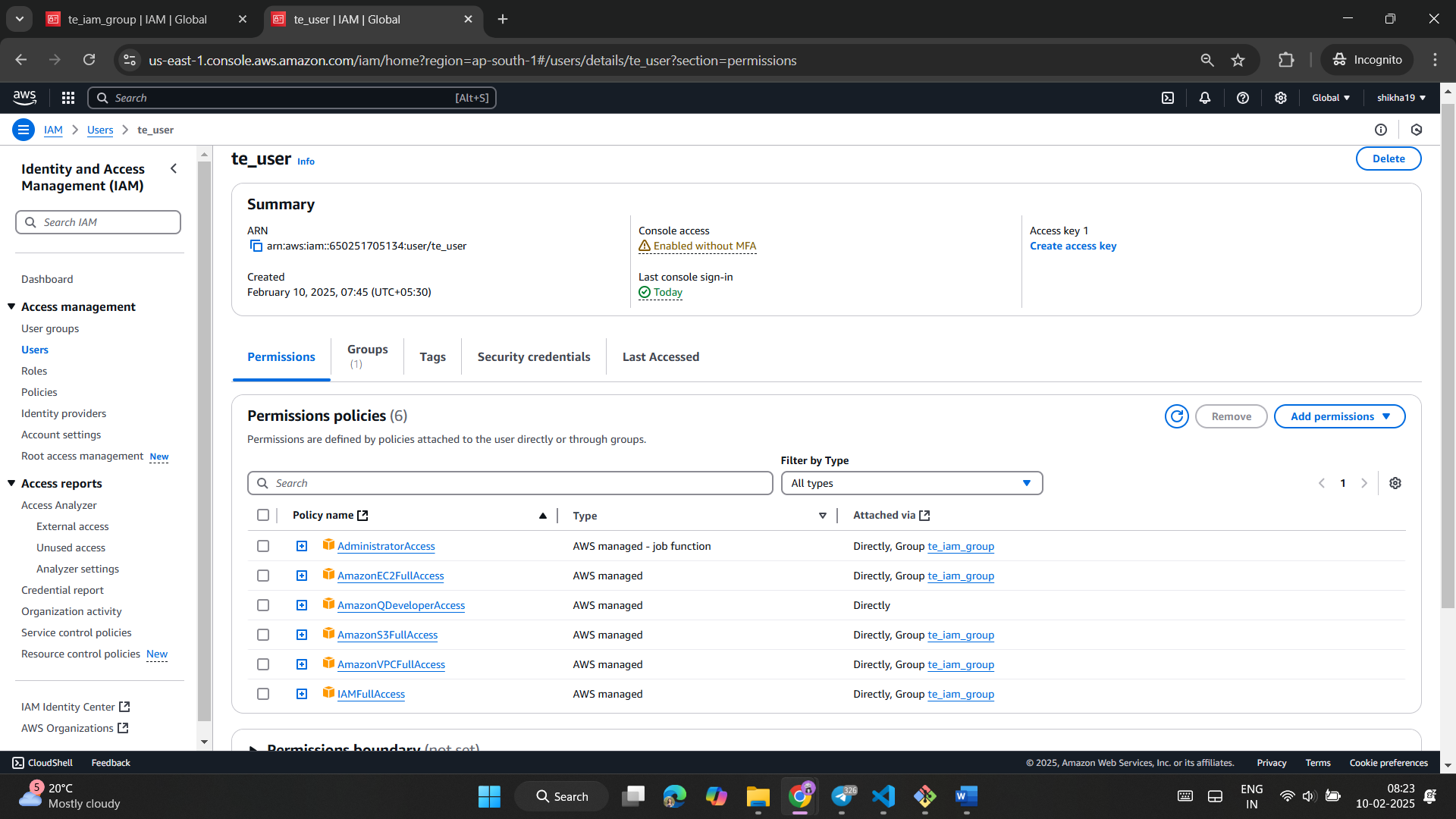
IAM

Step 1: IAM (Identity and Access Management)

1.1 Create an IAM User with Programmatic Access

1. Go to AWS Management Console → Navigate to IAM.
2. Policy Attach- ec2fullacess , s3fullacess, vpcfullacess,uamfullacess

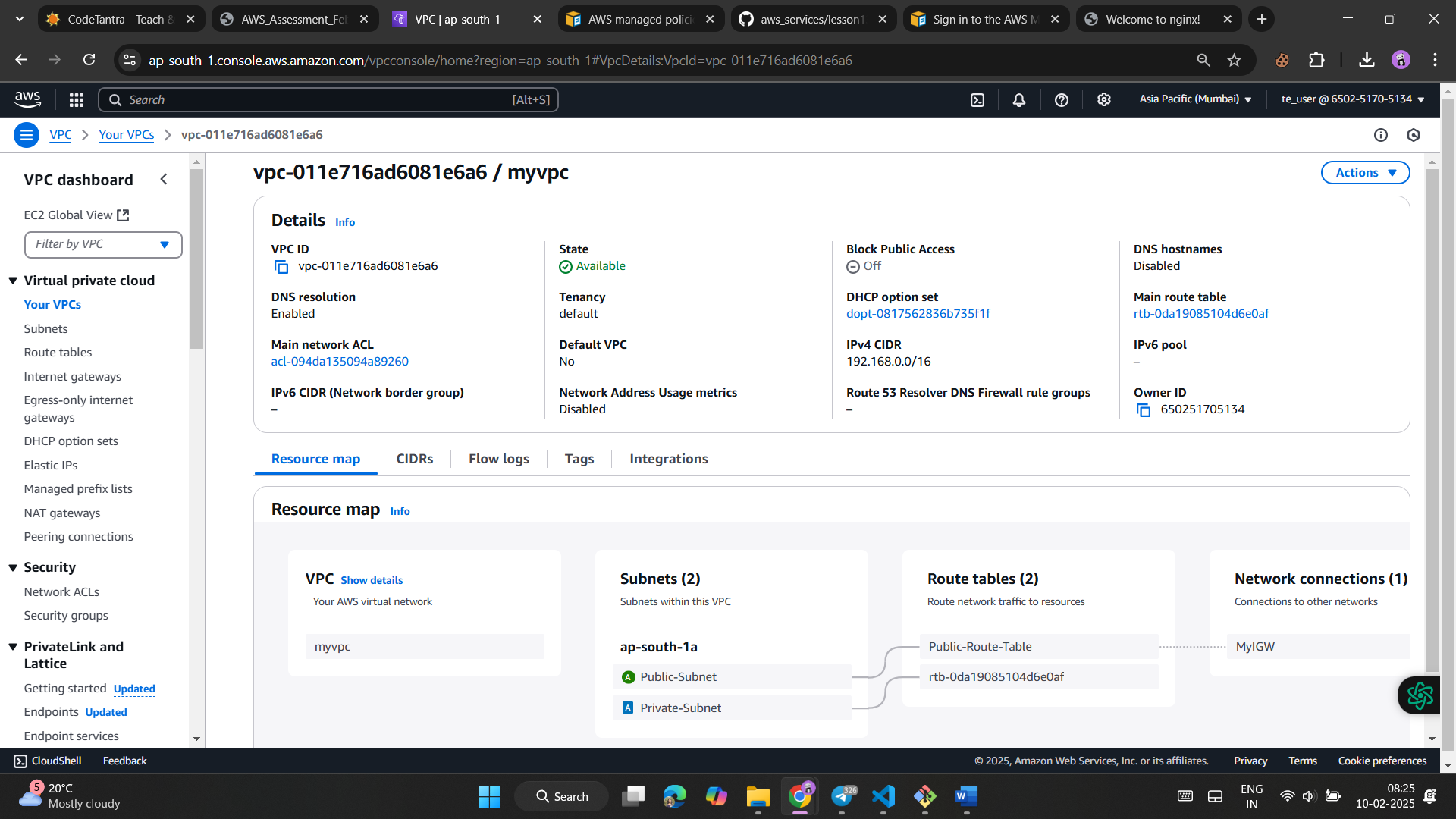
Create a IAM Group.

A screenshot of a computer

AI-generated content may be incorrect.

VPC

* Step 2: VPC and Subnets
* 2.1 Create a Custom VPC
* Enter CIDR Block: 192.168.0.0/16
* 2.2 Create Public and Private Subnets
  + Public Subnet:
    - Name: Public-Subnet
    - CIDR: 192.168.1.0/24
  + Private Subnet:
    - Name: Private-Subnet
    - CIDR: 192.168.2.0/24

A screenshot of a computer

AI-generated content may be incorrect.

Public subnet

A computer screen with a white background

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Public route table

**Click on Edit Routes, then Add Route:**

**Destination: 0.0.0.0/0**

**Target: Select Internet Gateway (MyIGW)**

A screenshot of a computer

AI-generated content may be incorrect.

INTERNET GATEWAY

Create and Attach an Internet Gateway

A screenshot of a computer

AI-generated content may be incorrect.

Ec2

A screenshot of a computer

AI-generated content may be incorrect.

NGINX

A screenshot of a computer

AI-generated content may be incorrect.

**COMMANDS**

Install & Configure Nginx

CHMOD COMMAND

ssh -i your-key.pem ec2-user@your-public-ip

sudo yum install nginx -y

sudo systemctl start nginx

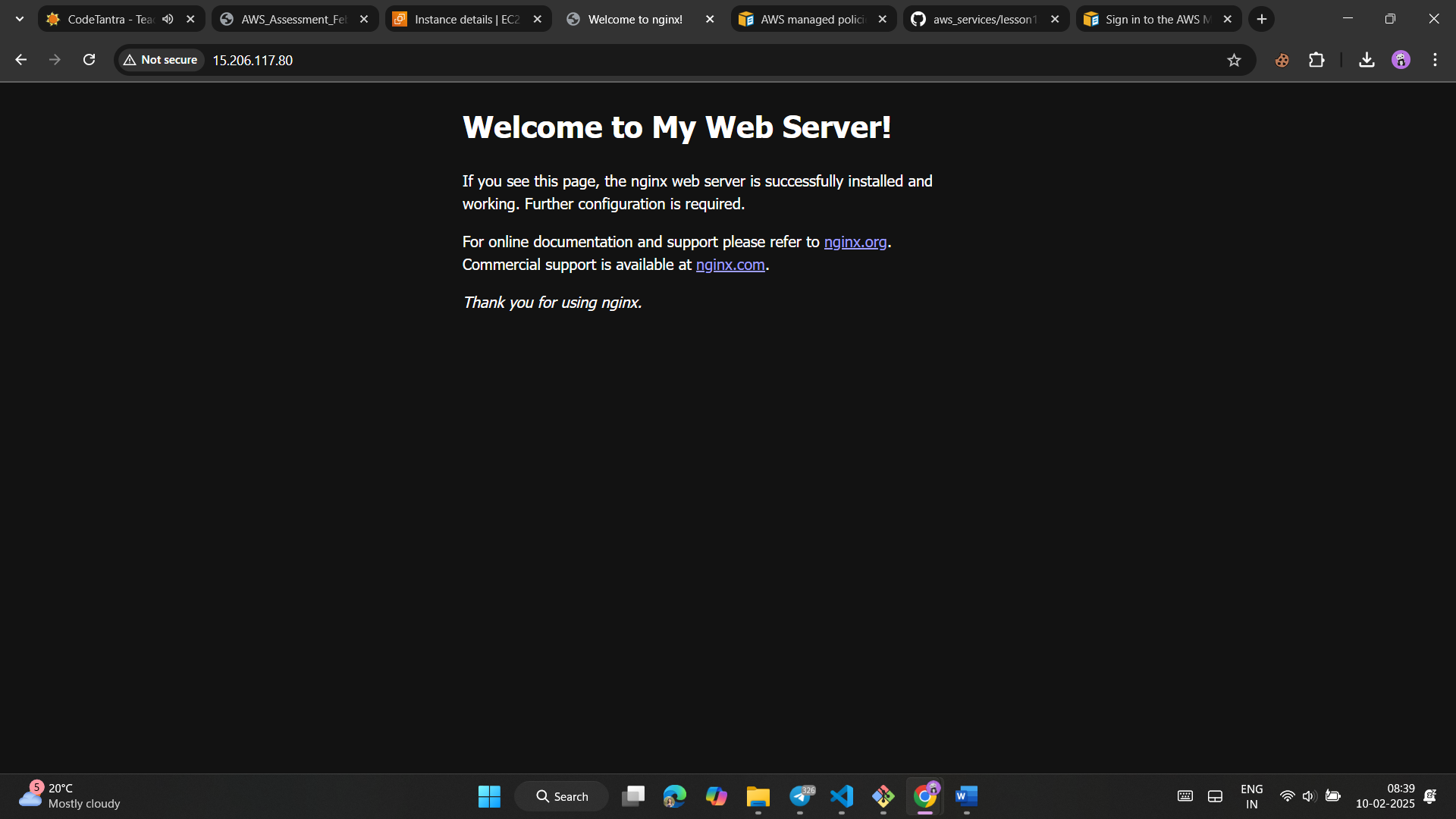
sudo systemctl enable nginx

ssh -i your-key.pem ec2-user@your-public-ip

<http://15.206.117.80>

cd /usr/share/nginx/html

sudo nano index.html

<!DOCTYPE html>  
<html>  
<head>  
<title>My Nginx Web Server</title>  
</head>  
<body>  
<h1>Welcome to My Web Server!</h1>  
</body>  
</html>

**What I Learn**

1. IAM (Identity and Access Management): Managing Access Control

* IAM enables the creation of users and groups, assigning fine-grained permissions to control access to AWS resources.

2. VPC (Virtual Private Cloud) and Subnets: Enhancing Security

* VPC provides an isolated giving full control over IP addressing, subnets, route tables, and gateways.
* I learned to create a custom VPC with public and private subnets:.
* By configuring route tables and attaching an Internet Gateway to the VPC, I allowed Internet access only for resources in the public subnet

3. EC2 (Elastic Compute Cloud): I learned how to launch an EC2 instance in the public and use SSH to access it.

4. Nginx: Lightweight Web Hosting

* I installed and configured Nginx on an EC2 instance, hosting the default welcome page and testing the setup using the instance's public IP address.

Overall Summary

This assignment demonstrated the integration of various AWS services to hosting a web server. By leveraging IAM, VPC, EC2, and Nginx, I gained hands-on experience in managing access.