**Implementing a Web Application Firewall (WAF) to Filter Web Traffic**

**NAME: Ajaz Noorain Ahamed**

**CONTACT:** [**noorainahamed13@gmail.com**](mailto:noorainahamed13@gmail.com)

**REPORT:** Implementing a Web Application Firewall (WAF) to Filter Web Traffic

**1. Introduction**

A **Web Application Firewall (WAF)** is a security solution designed to protect web applications from common threats such as **SQL injection, cross-site scripting (XSS), and Distributed Denial-of-Service (DDoS) attacks**. WAF acts as a **filtering layer** between users and the web application, analyzing and blocking malicious requests before they reach the backend servers.

In this project, we will implement **AWS WAF** (Web Application Firewall) to secure a web application hosted on AWS. AWS WAF integrates with **Amazon CloudFront, Application Load Balancer (ALB), and API Gateway** to protect against malicious traffic.

## **2. Project Objectives**

**The goal of this project is to:**

* **Deploy AWS WAF to monitor and filter web traffic.**
* **Protect against common security threats such as SQL Injection, XSS, and Bot Attacks.**
* **Integrate AWS WAF with Application Load Balancer (ALB) to safeguard EC2 instances.**
* **Configure custom rules to block unwanted traffic.**
* **Set up monitoring and logging using AWS CloudWatch for analysis.**

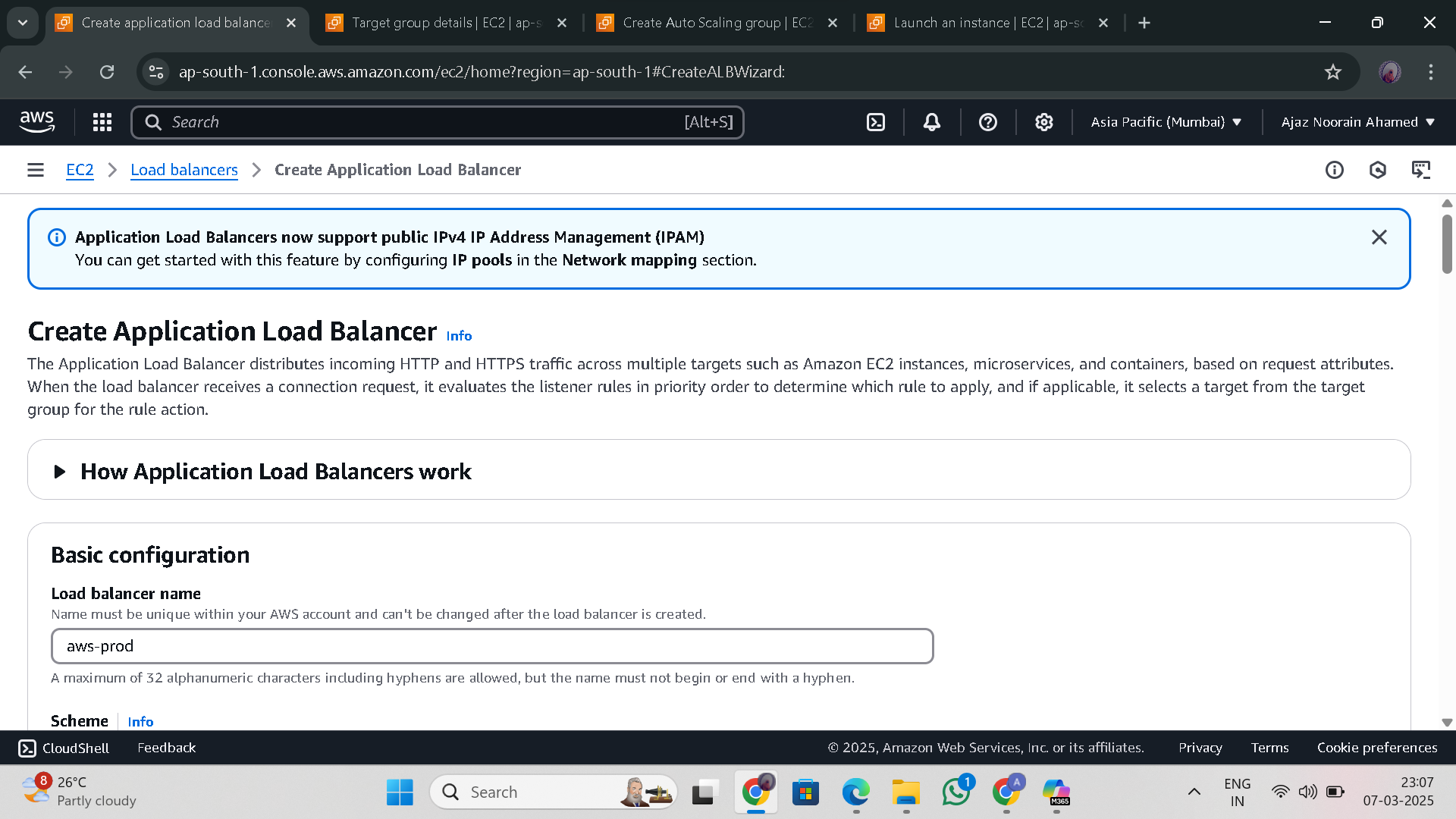
## **3. Architecture**

### **AWS Services Used:**

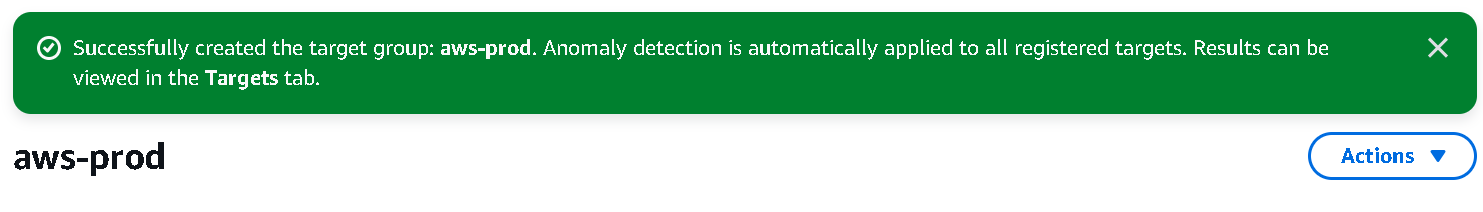
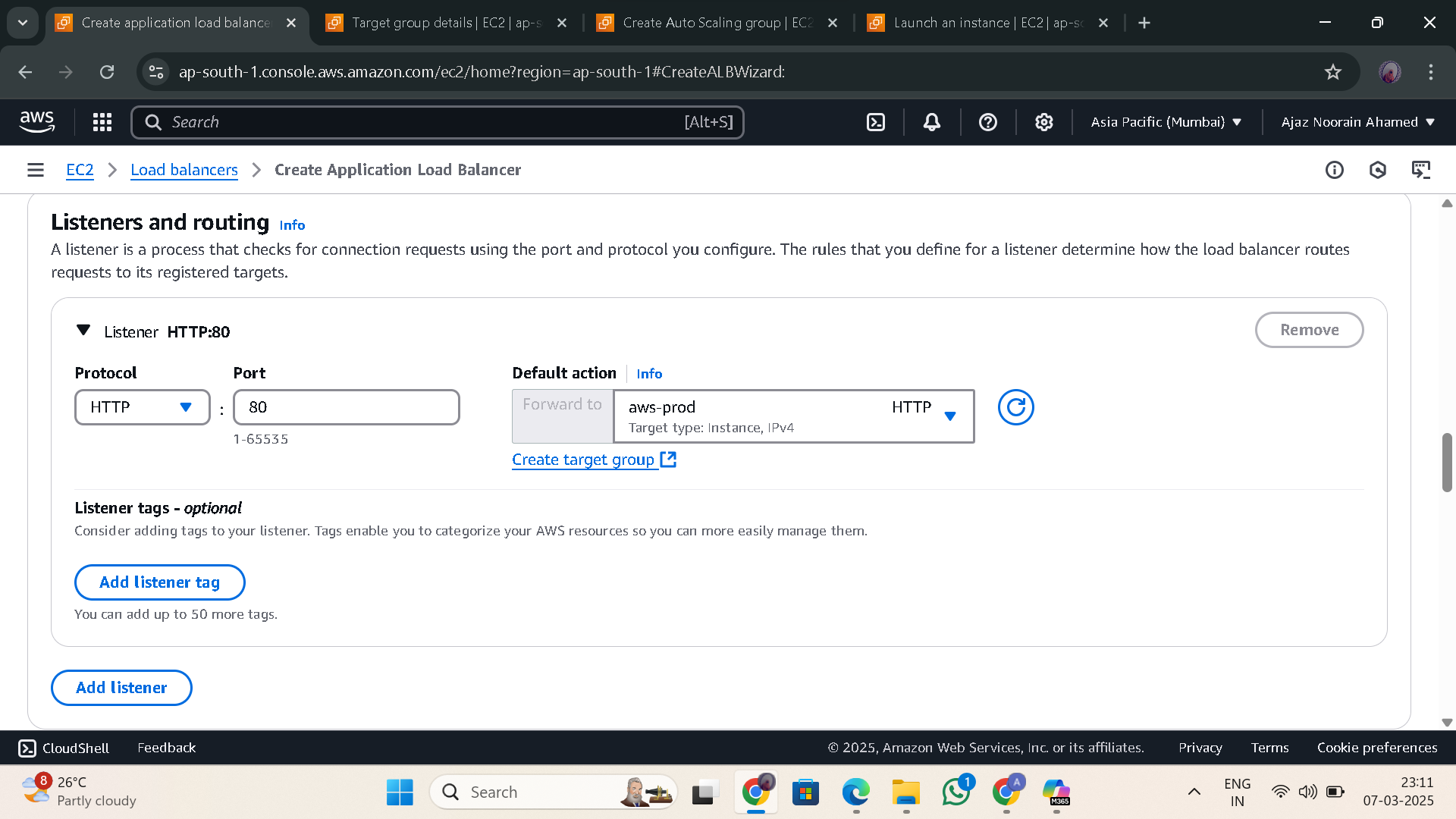
* **AWS WAF – Protects web applications by filtering malicious traffic.**
* **Application Load Balancer (ALB) – Distributes traffic across EC2 instances.**
* **EC2 Instances – Hosts the web application.**
* **Amazon CloudWatch – Monitors WAF logs and performance.**
* **Amazon S3 – Stores WAF logs (optional).**
* **AWS Lambda (Optional) – Automates rule updates based on logs.**

## **4. Implementation Steps**

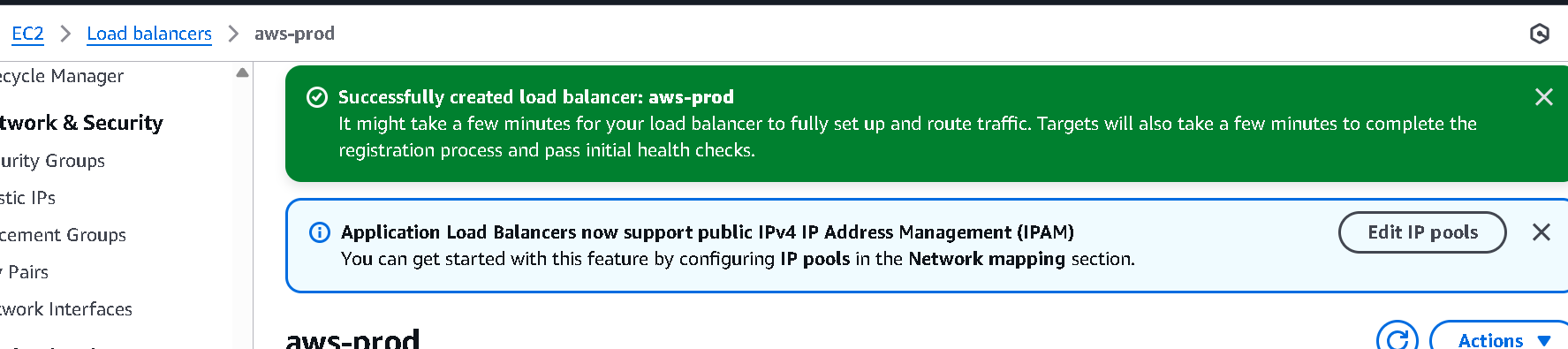
### **Step 1: Set Up an Application Load Balancer (ALB)**



1. **Go to AWS Console → EC2 → Load Balancers → Create Load Balancer.**
2. **Select "Application Load Balancer".**
3. **Configure:** 
   1. **Name: (any name)**
   2. **Scheme: Internet-facing**
   3. **Listeners: HTTP (80) or HTTPS (443) (Recommended)**
   4. **Availability Zones: Select at least two AZs.**
4. **Create a Target Group and register your EC2 instances.**



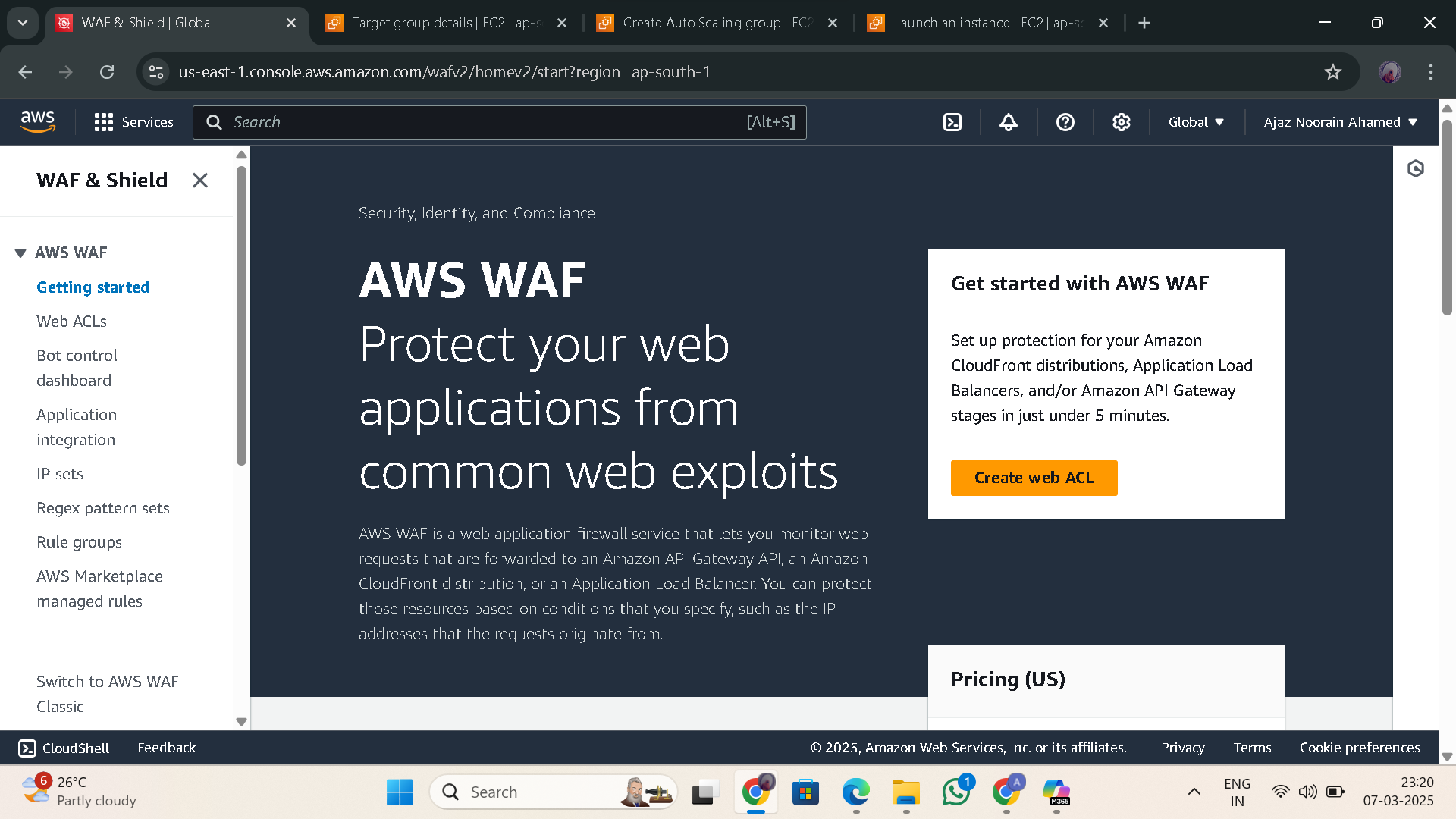
1. **Launch the ALB (application load balancer)**



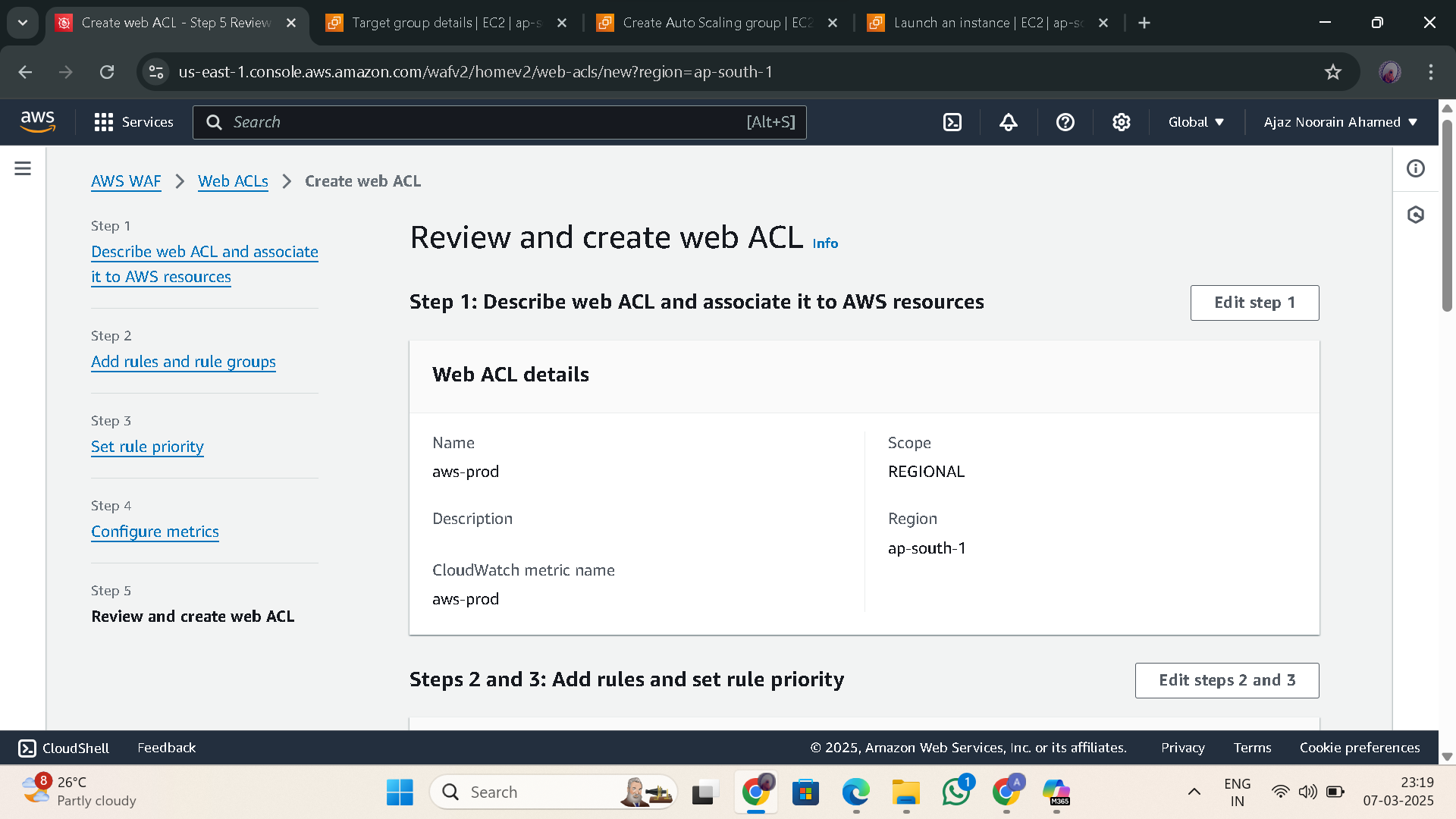
### **Step 2: Deploy a Web Application (Optional)**

1. **Launch EC2 instances** and install a web server (e.g., Apache, Nginx).
2. **Host a sample web application** (e.g., WordPress, custom web app).
3. **Ensure the security group allows ALB traffic**.

### **Step 3: Create AWS WAF Web ACL**



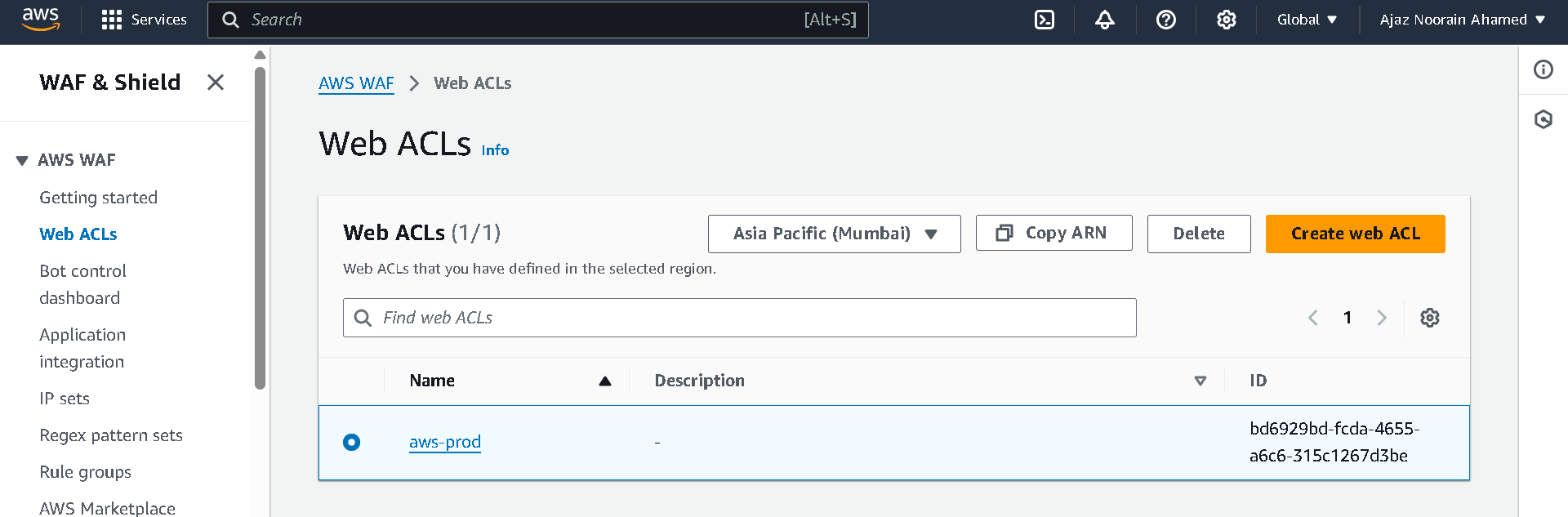
1. **Go to AWS WAF & Shield** in AWS Console.
2. **Create a new Web ACL**:
   1. Name: (aws-prod)
   2. Scope: **Regional (For ALB, API Gateway, or CloudFront)**
3. Resource: Select the **ALB**



1. **Add Rule Groups**:

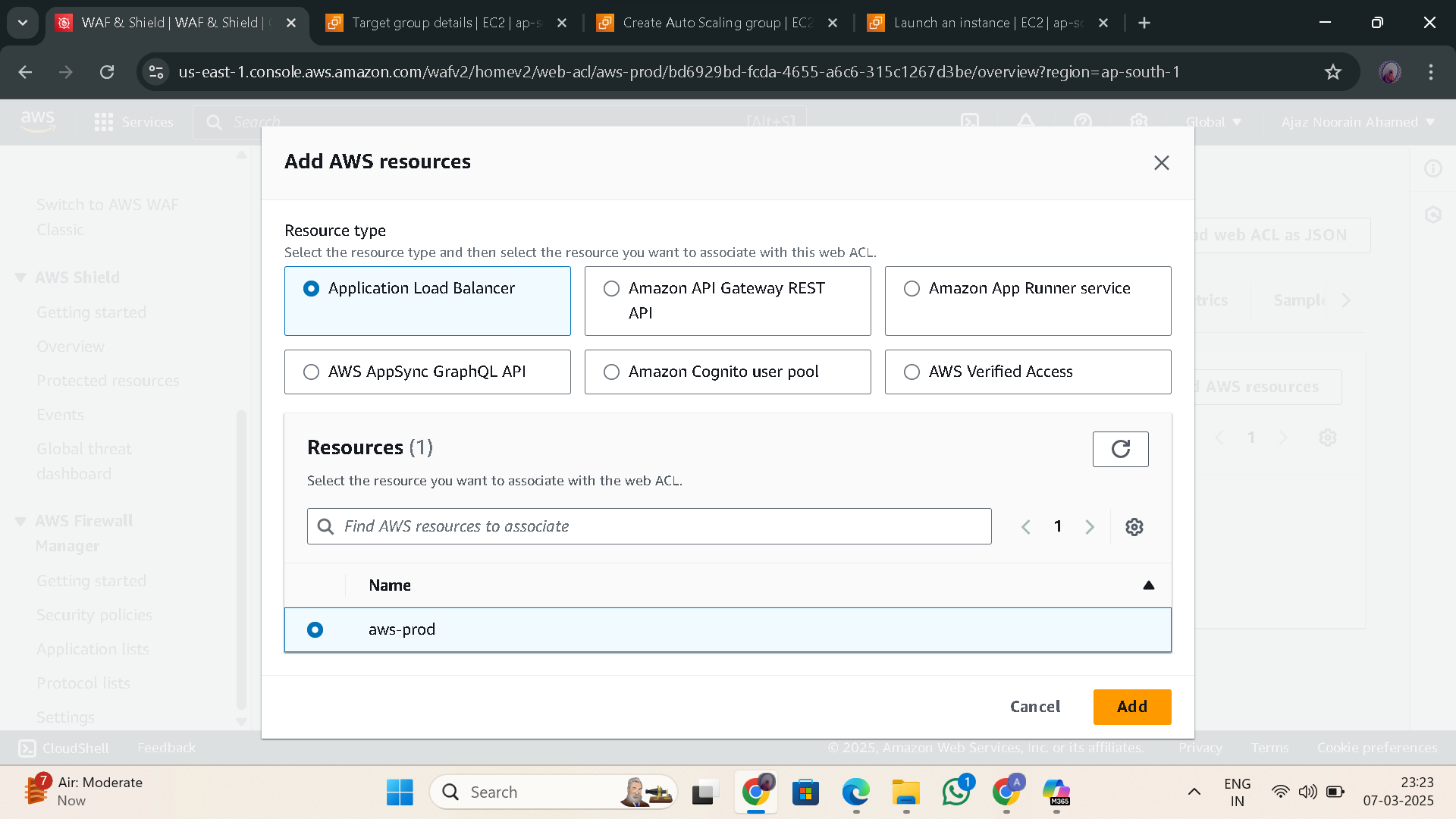
**AWS Managed Rules** (Predefined security rules by AWS)

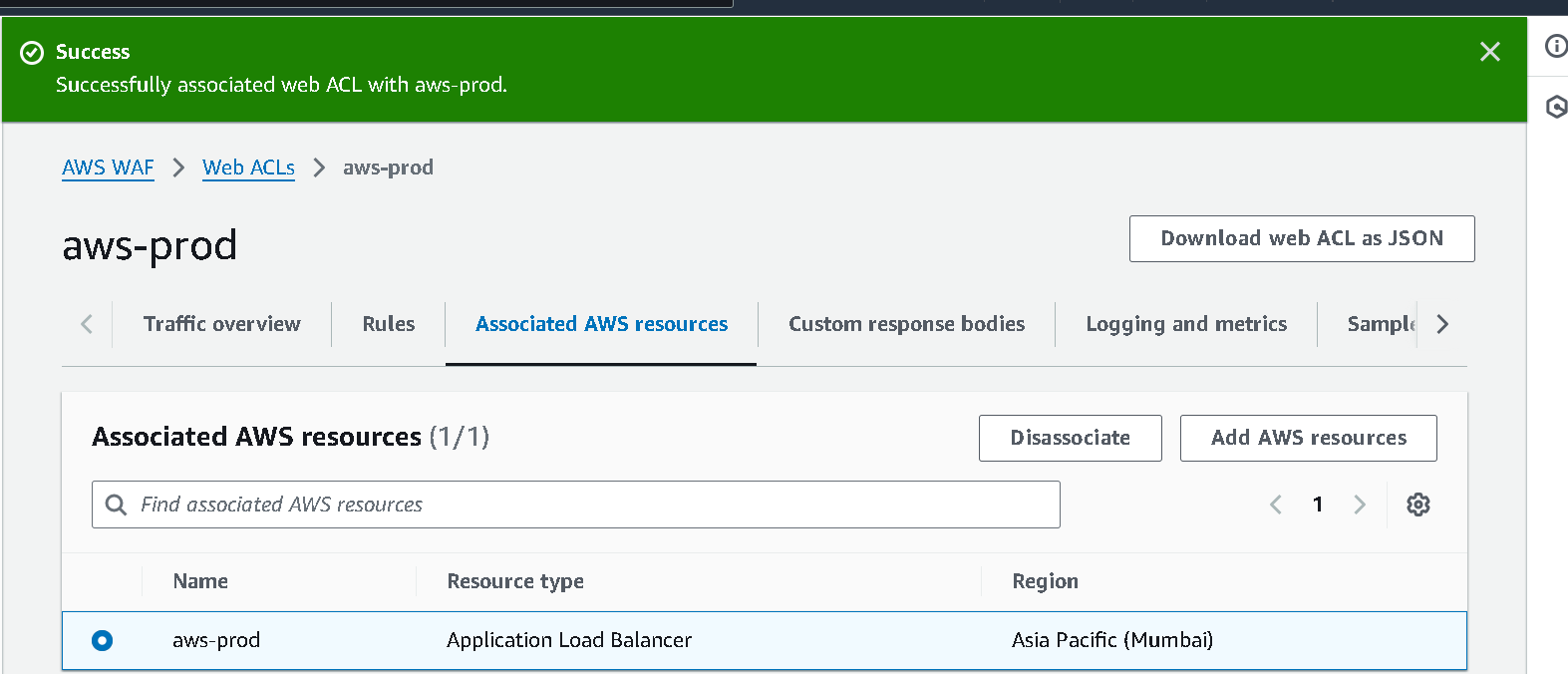
1. **Set Default Action** → Allow traffic that does not match any blocking rules.
2. **Deploy the Web ACL**.



### **Step 4: Associate WAF with ALB**

* In the **WAF Console**, select the **Web ACL (aws-prod)**.
* Click **Resources → Add ALB (aws-prod)**.
* Save and deploy.





### **Step 5: Test the WAF Configuration**

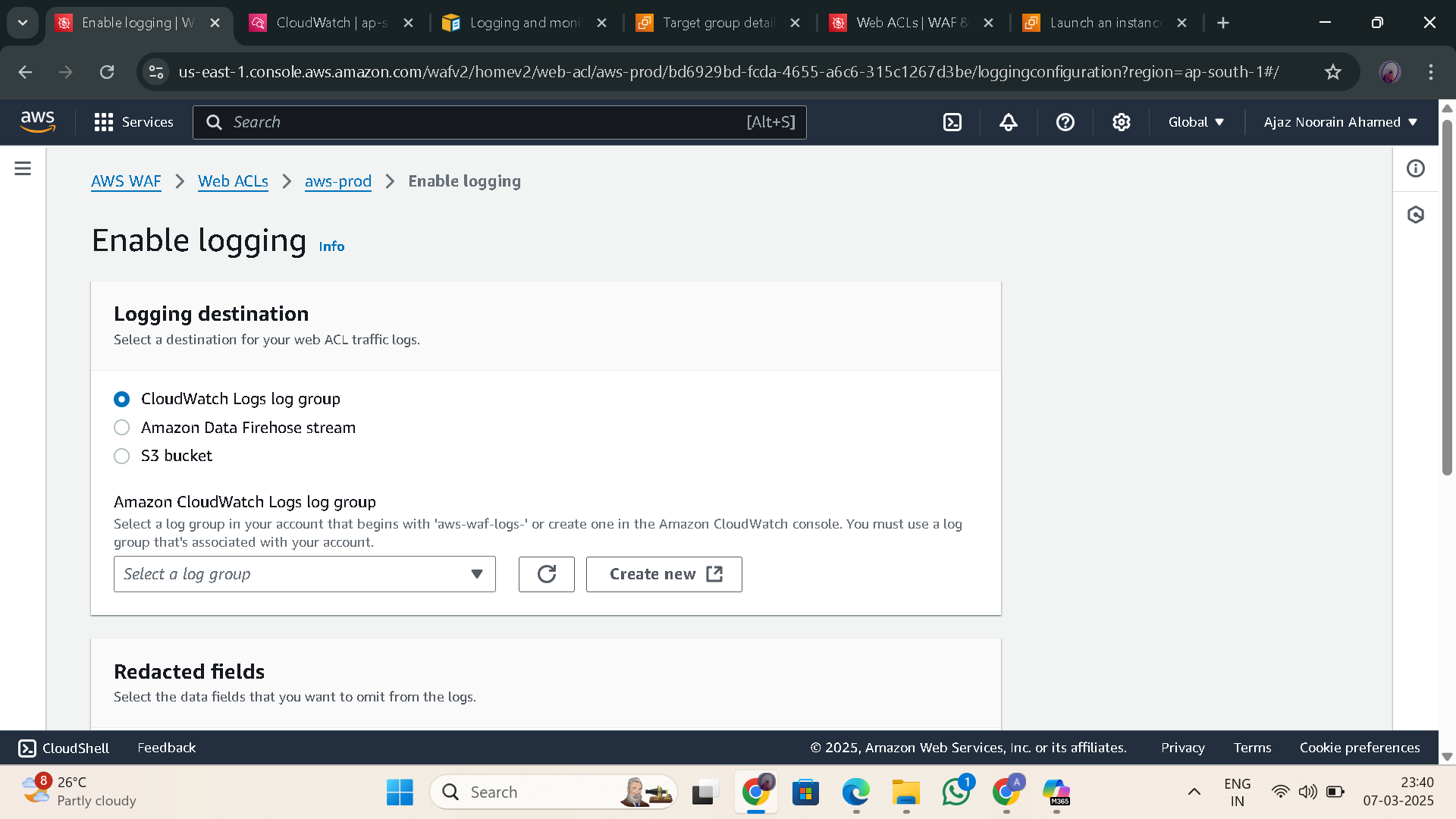
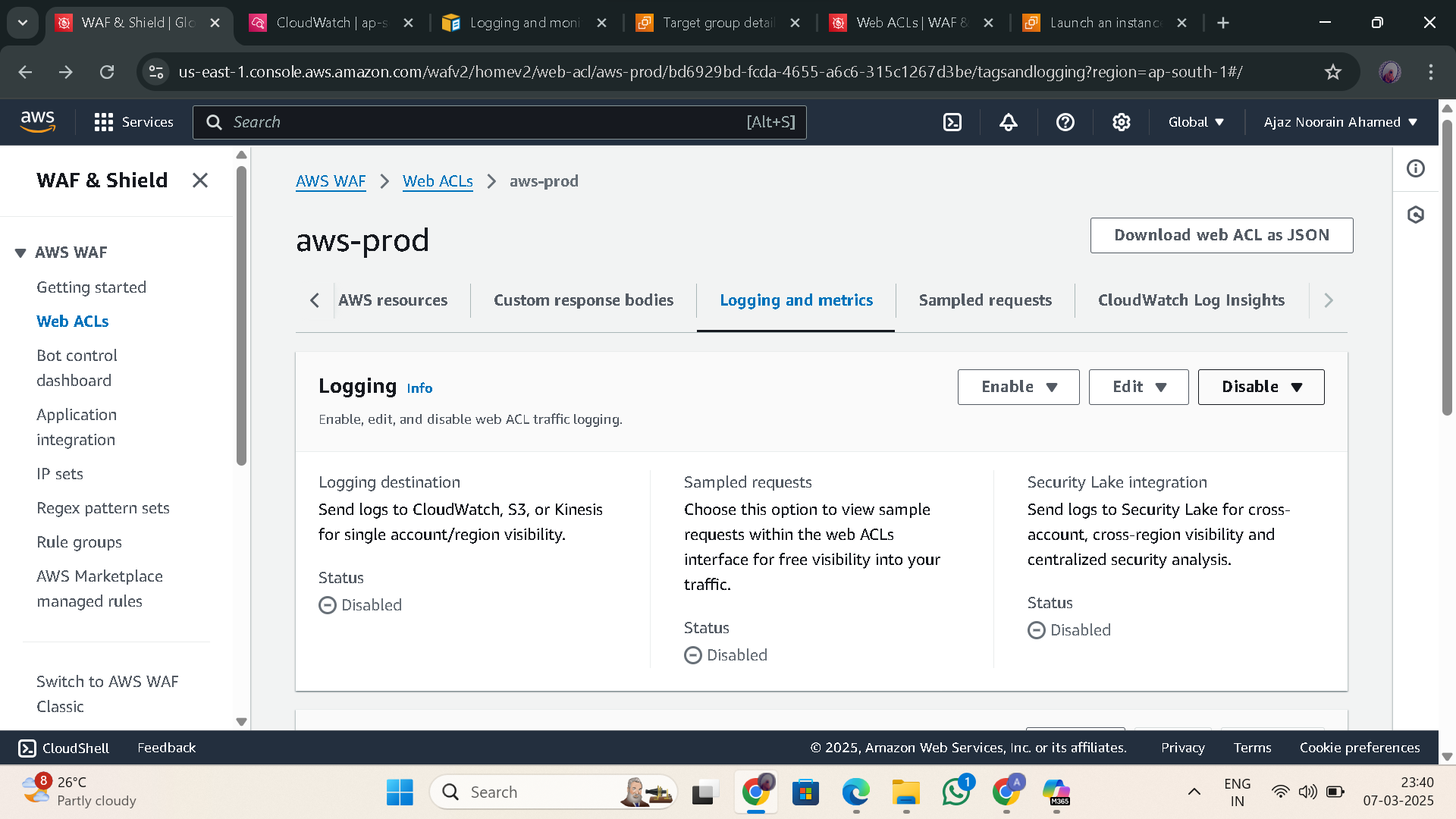
* **Use tools like Postman or c\_URL to send malicious requests (e.g., SQL injection attack).**
* **Example:**

**curl -X GET "**[**http://your-ALB-domain.com/?id=1**](http://your-ALB-domain.com/?id=1)**' OR '1'='1"**

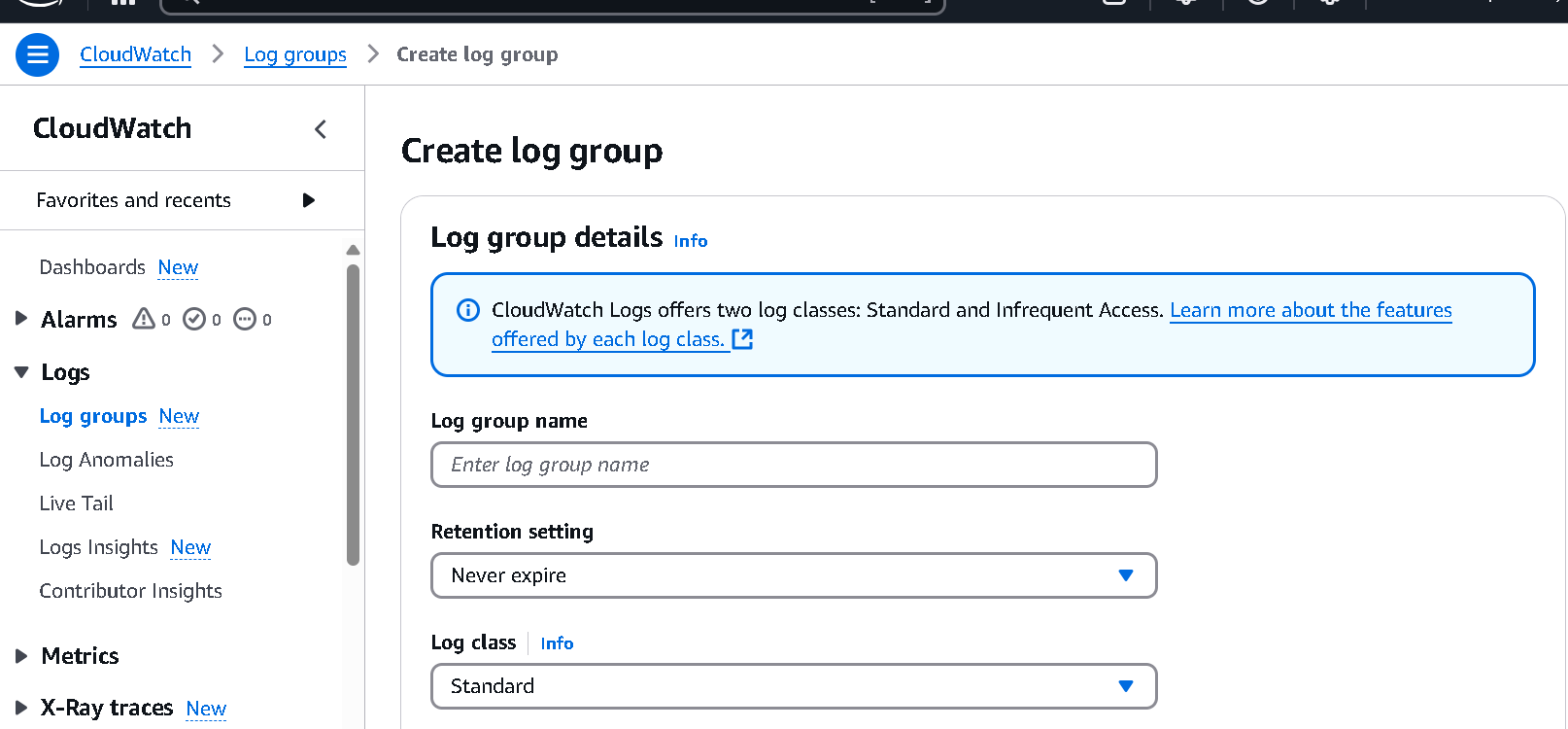
* **Verify in AWS WAF Logs (CloudWatch) that the request is blocked.**

### **Step 6: Monitor and Tune AWS WAF**

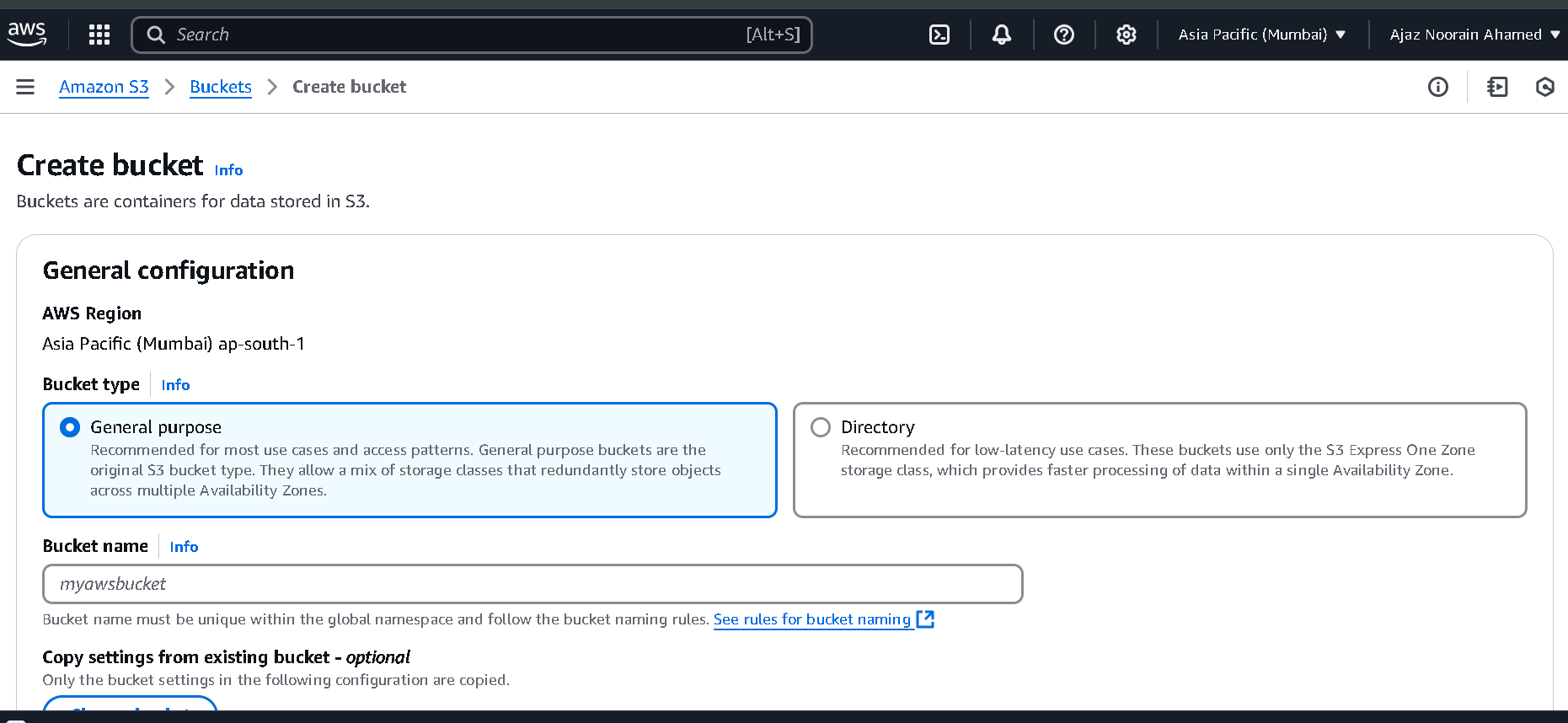
* **Enable Logging**:
  + Go to AWS WAF → Logging and Metrics.
  + Enable logging to **Amazon CloudWatch** or **S3**.



* Create and Select your **cloud watch or S3**



**OR**



* **Analyze Logs**:
  + Identify false positives and fine-tune rules.
* **Enable AWS Shield Standard** (Free DDoS Protection).

