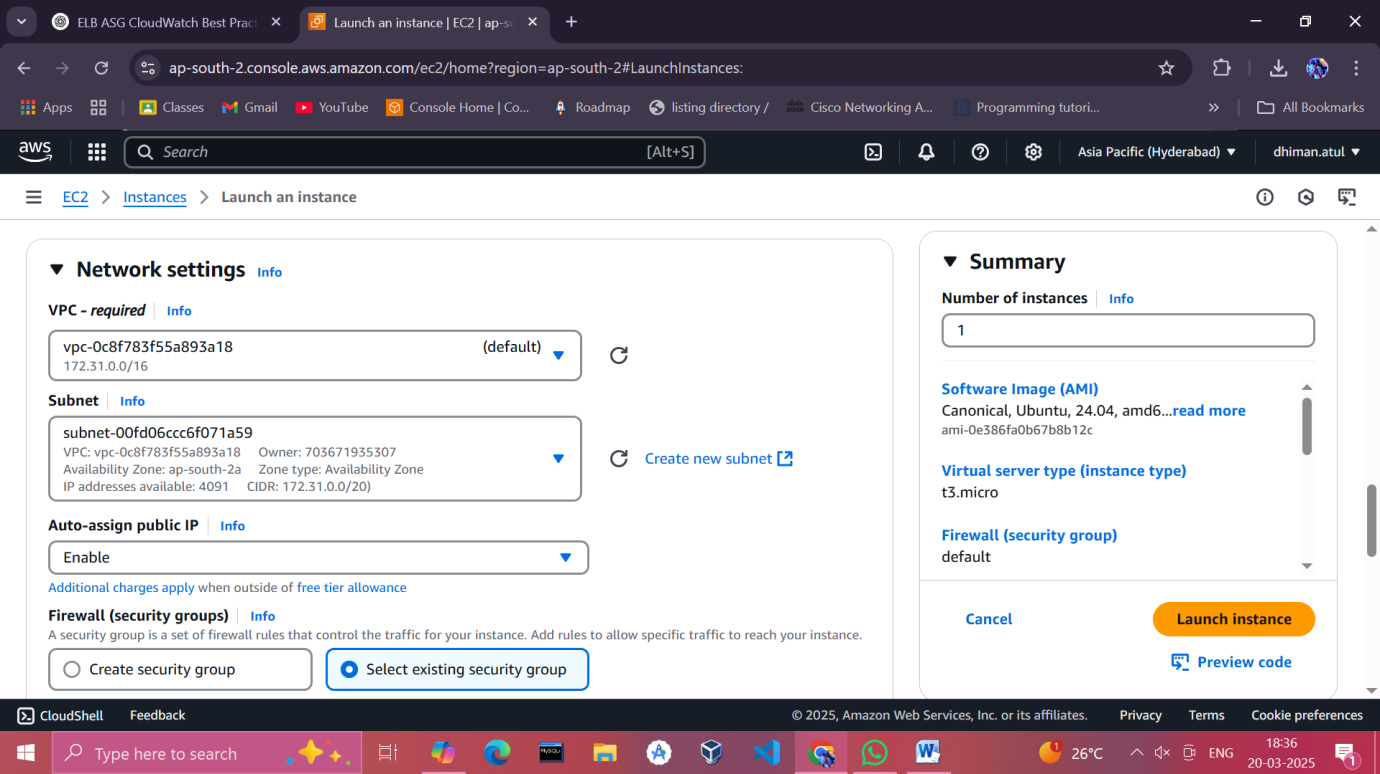
**Create a web application with an Elastic Load Balancer, an Auto Scaling group across two Availability Zones, and an Amazon CloudWatch metric with an alarm. Verify Auto Scaling by removing instances and testing load changes.**

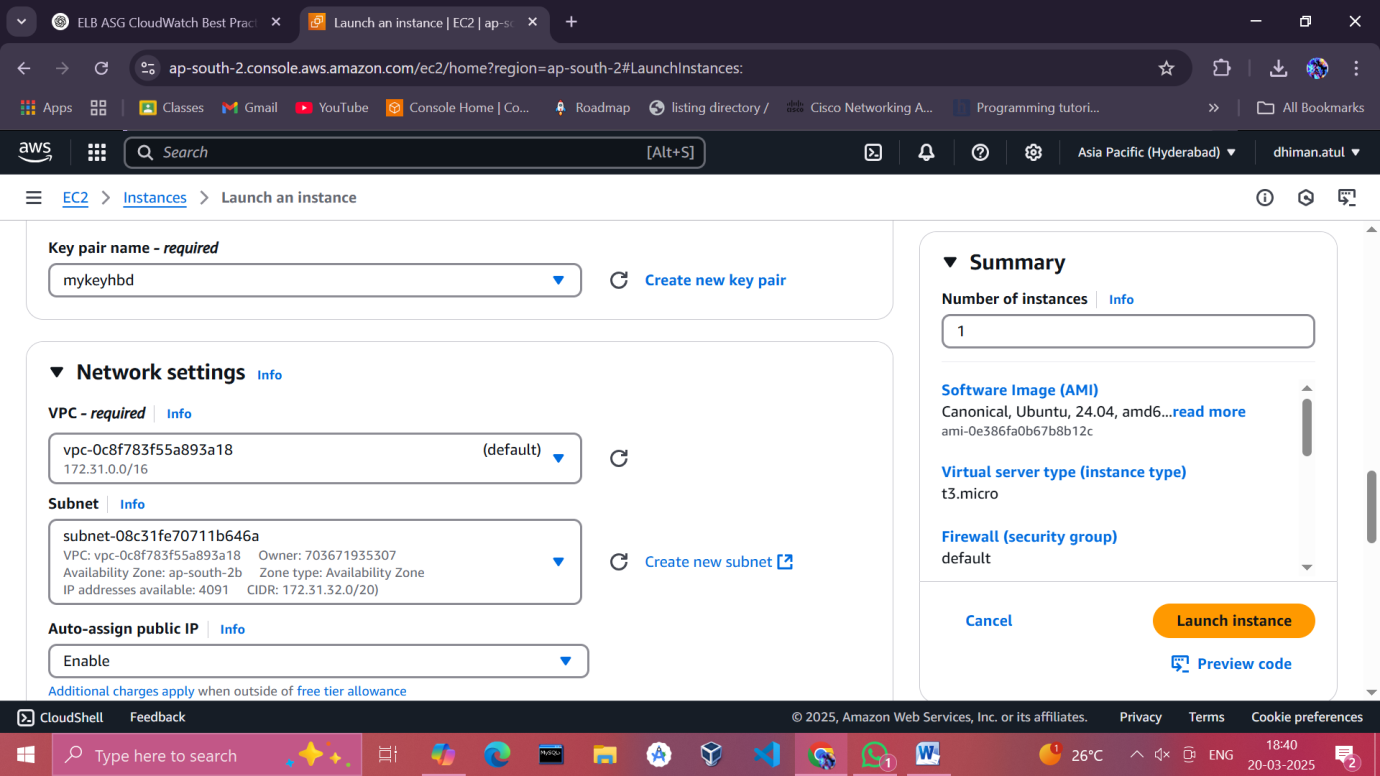
**1. Deploy Web Application on EC2 Instances**

1. **Launch EC2 Instances**:
   * Access the **AWS EC2 Dashboard** and click on **Launch Instance**.
   * Select an Ubuntu AMI (e.g., **Ubuntu Server 20.04 LTS**).
   * Choose an **instance type** (e.g., **t2.micro** for free-tier eligibility).
   * In the **network settings**, choose your VPC and select two subnets in different Availability Zones.

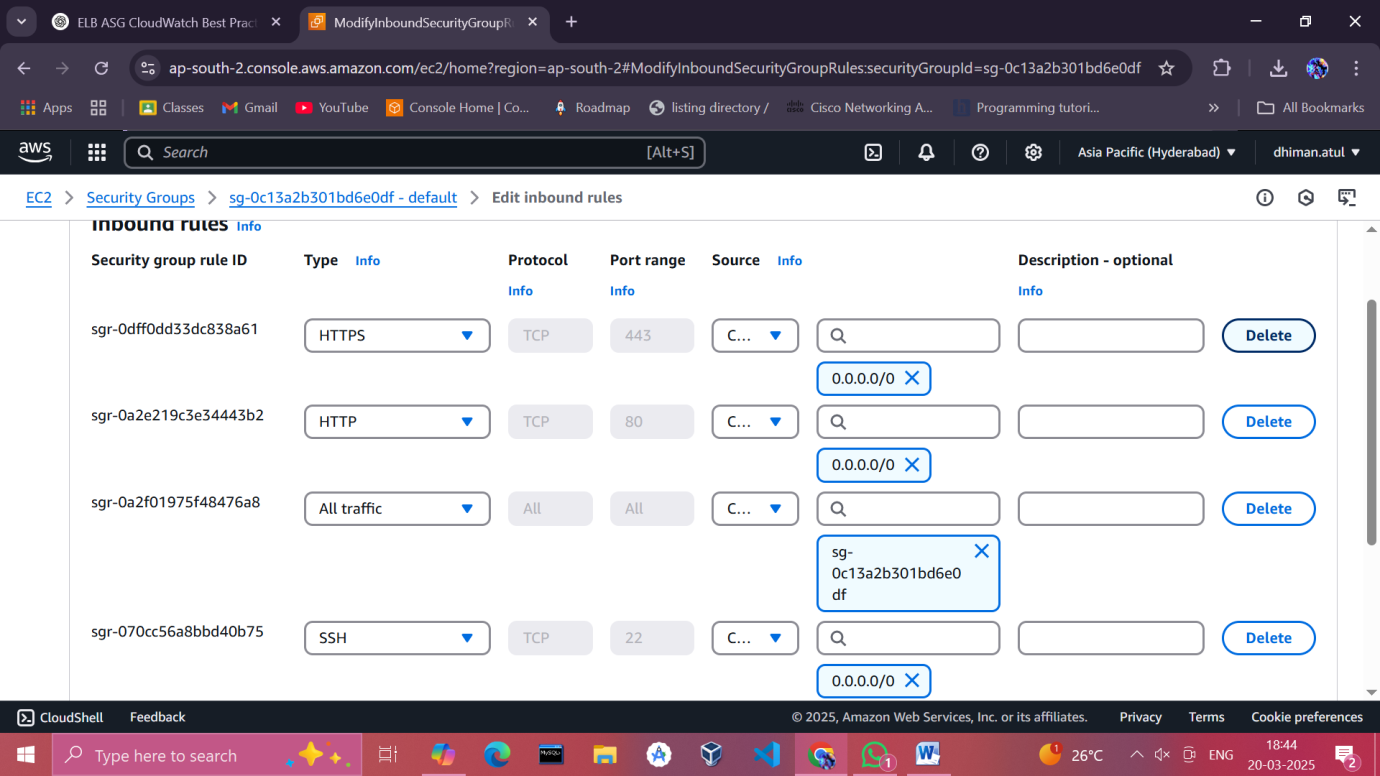
Zone a:



Zone b:

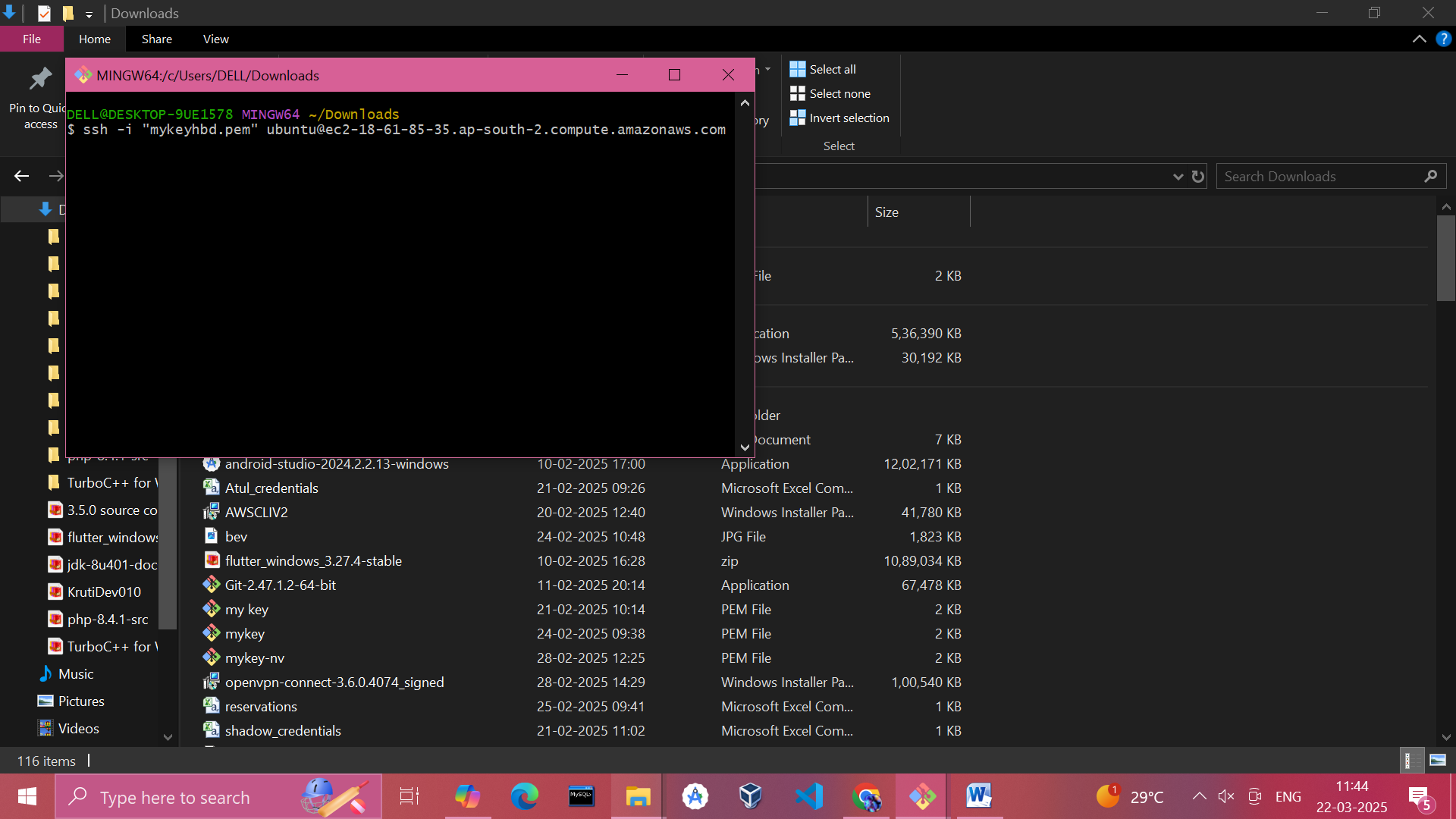


* + Enable **Auto-assign Public IP** for accessibility.
  + Configure a **security group**:
    - Allow **port 22 (SSH)** for remote access.
    - Allow **port 80 (HTTP)** for web traffic.

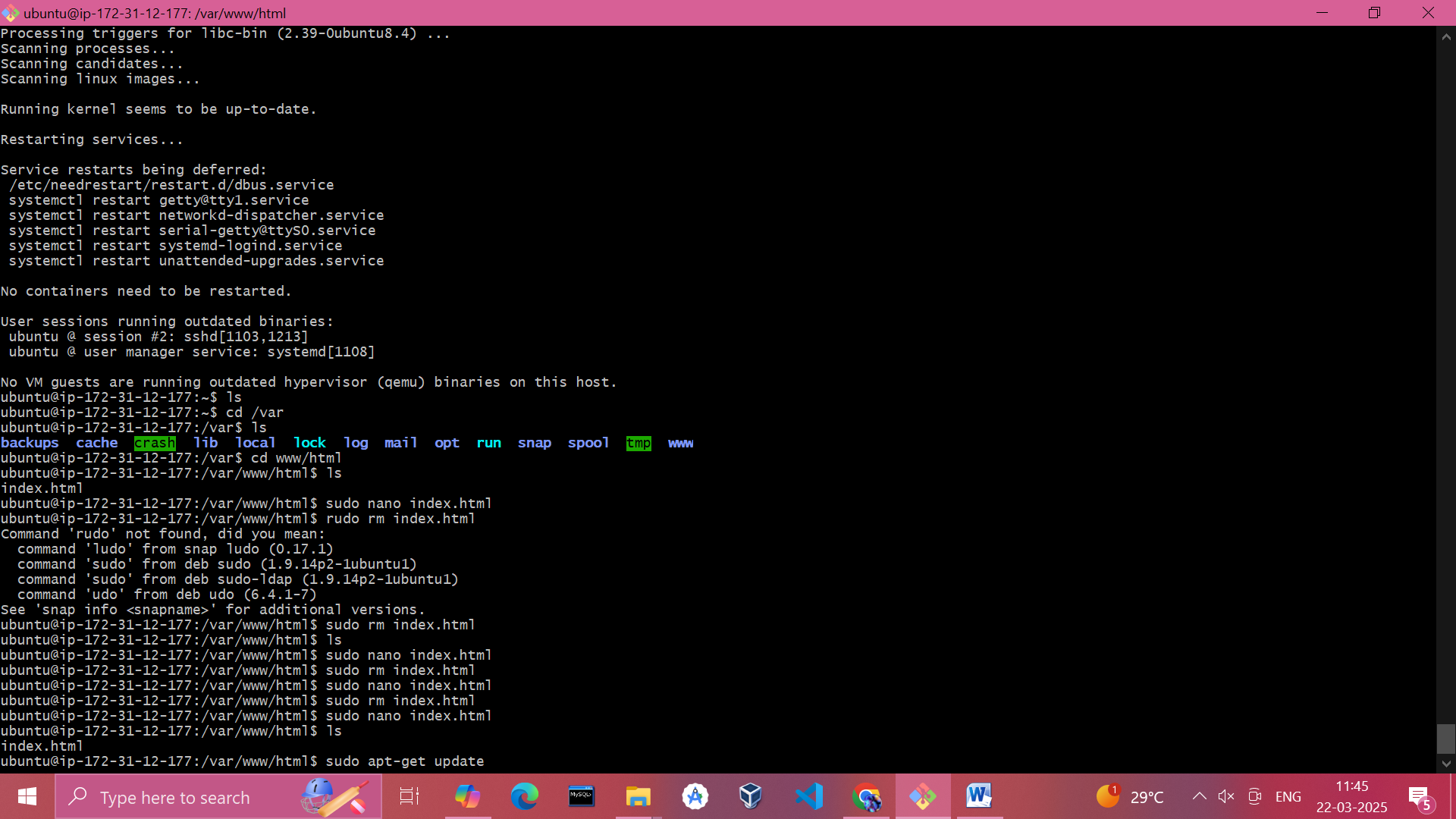


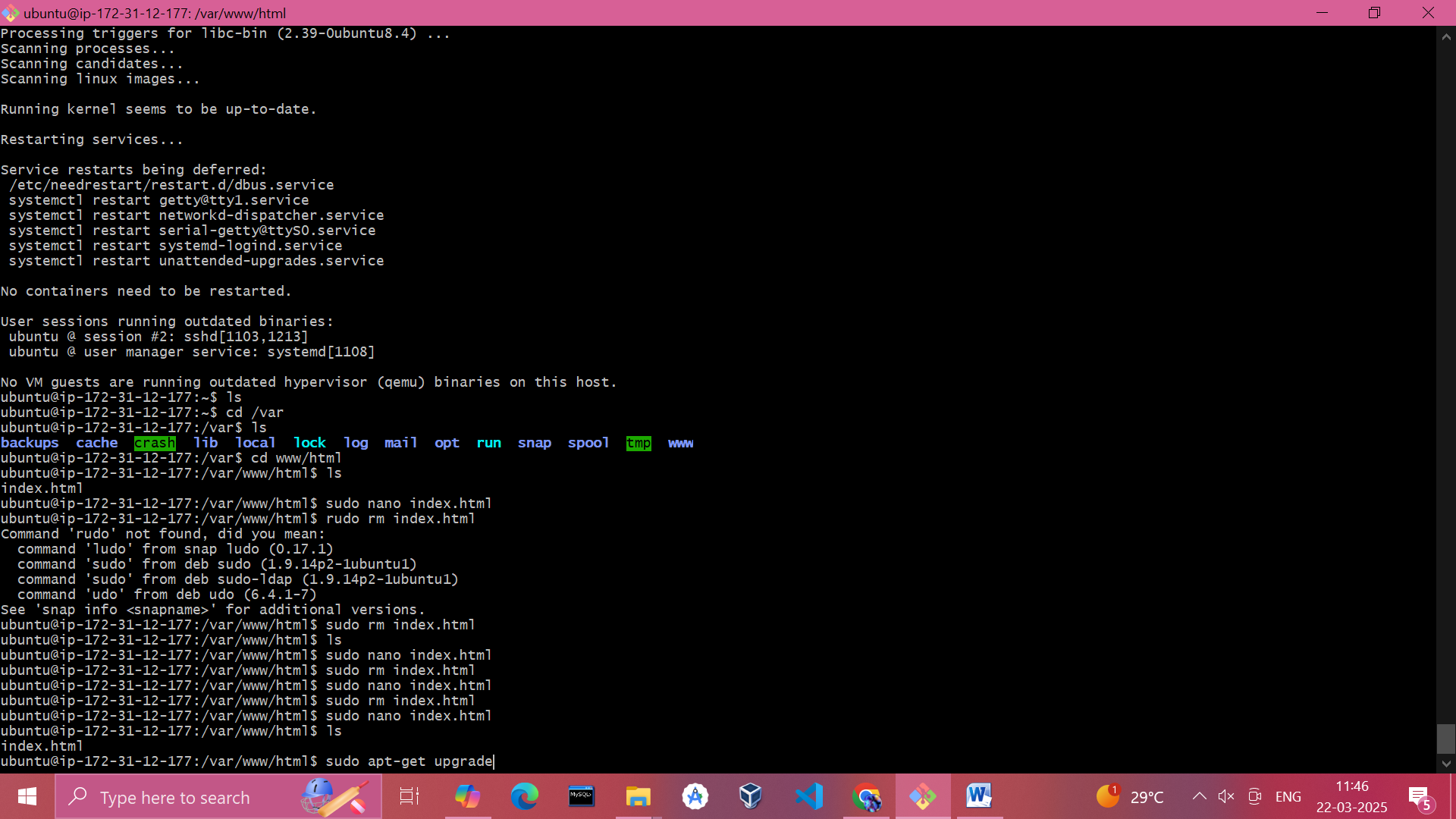
* + Review and launch, creating or using an existing key pair.

1. **Set Up the Web Server**:
   * Connect to each instance via SSH:
   * ssh -i your-key.pem ubuntu@<instance-public-IP>

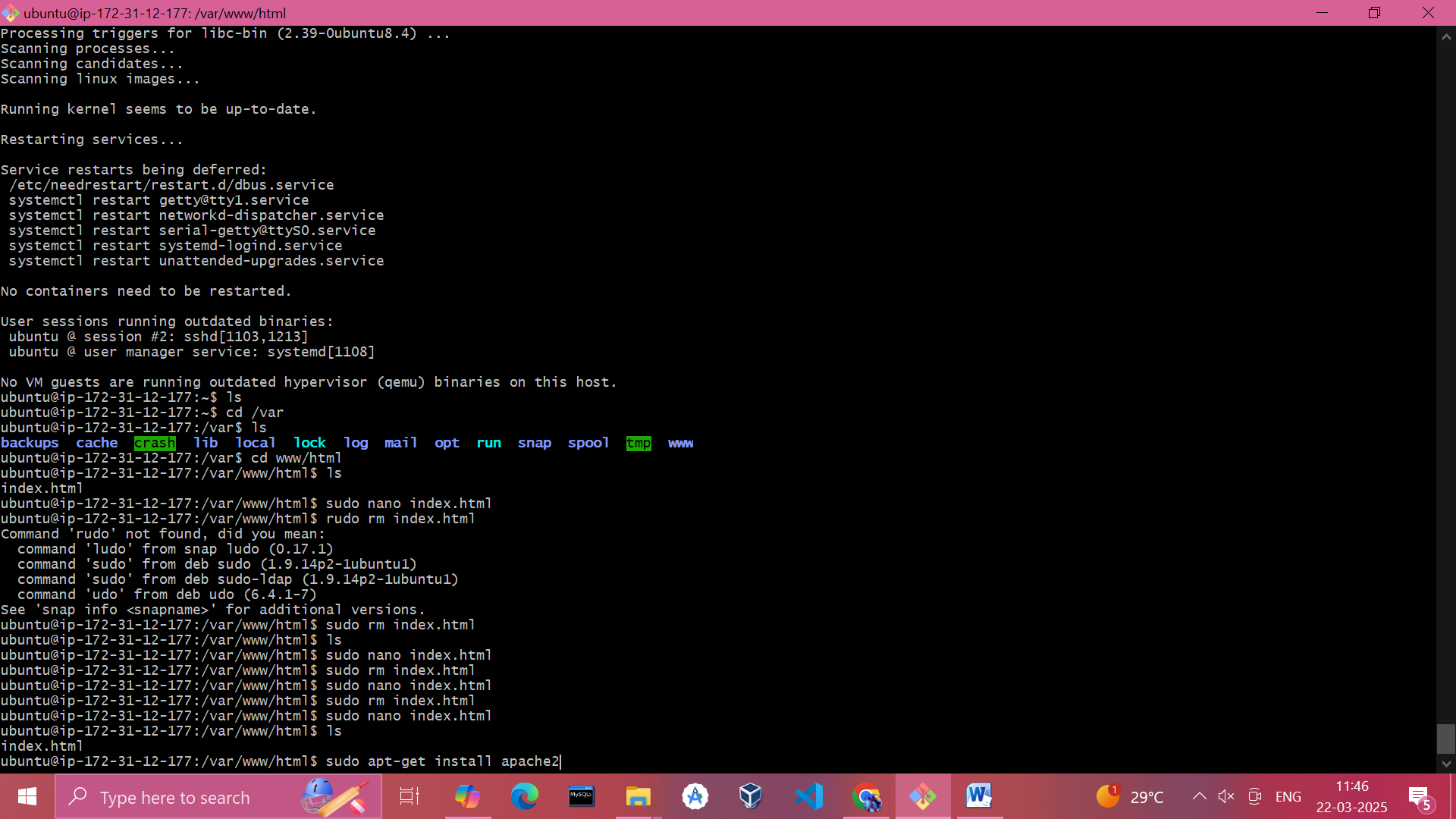


* + Update system packages:
  + sudo apt update && sudo apt upgrade –y

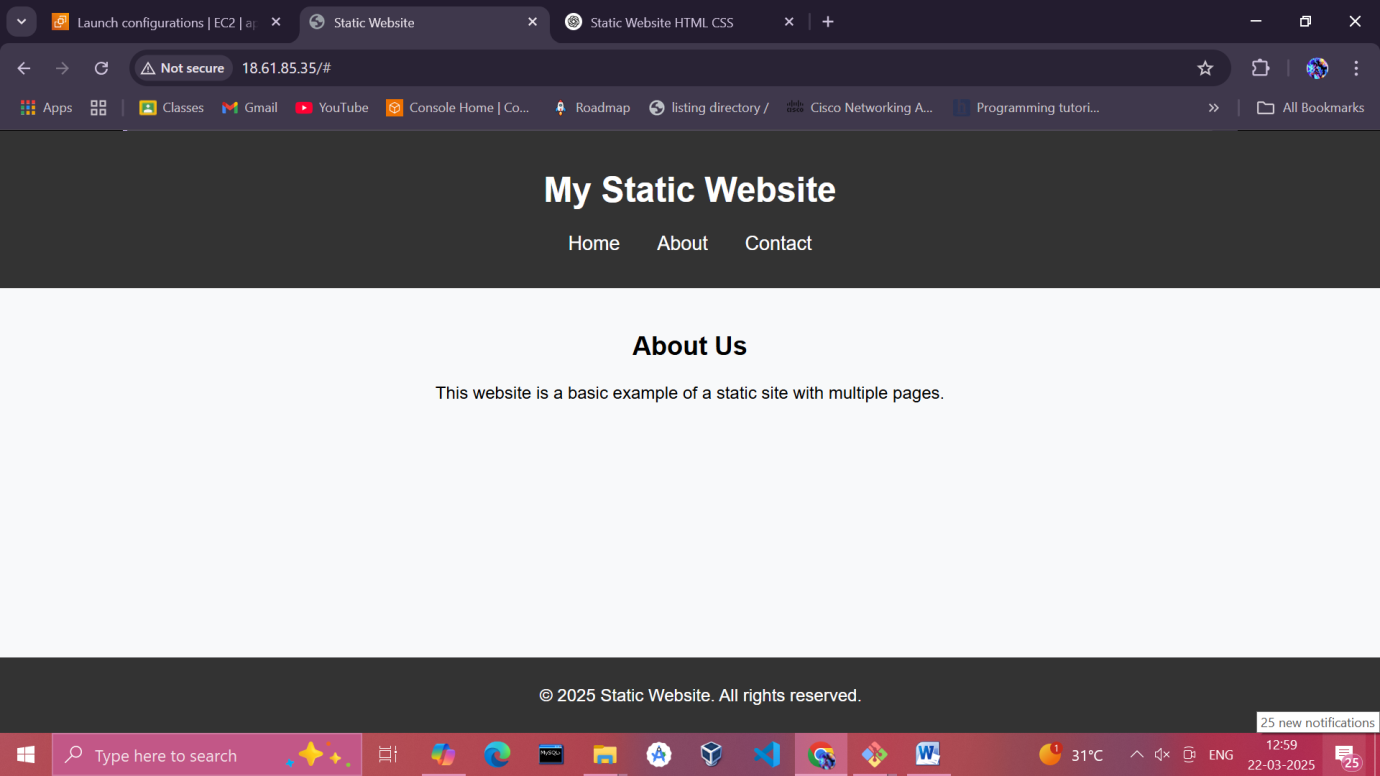




* + Install a web server (e.g., Apache or Nginx):
    - Apache:
    - sudo apt install apache2 -y

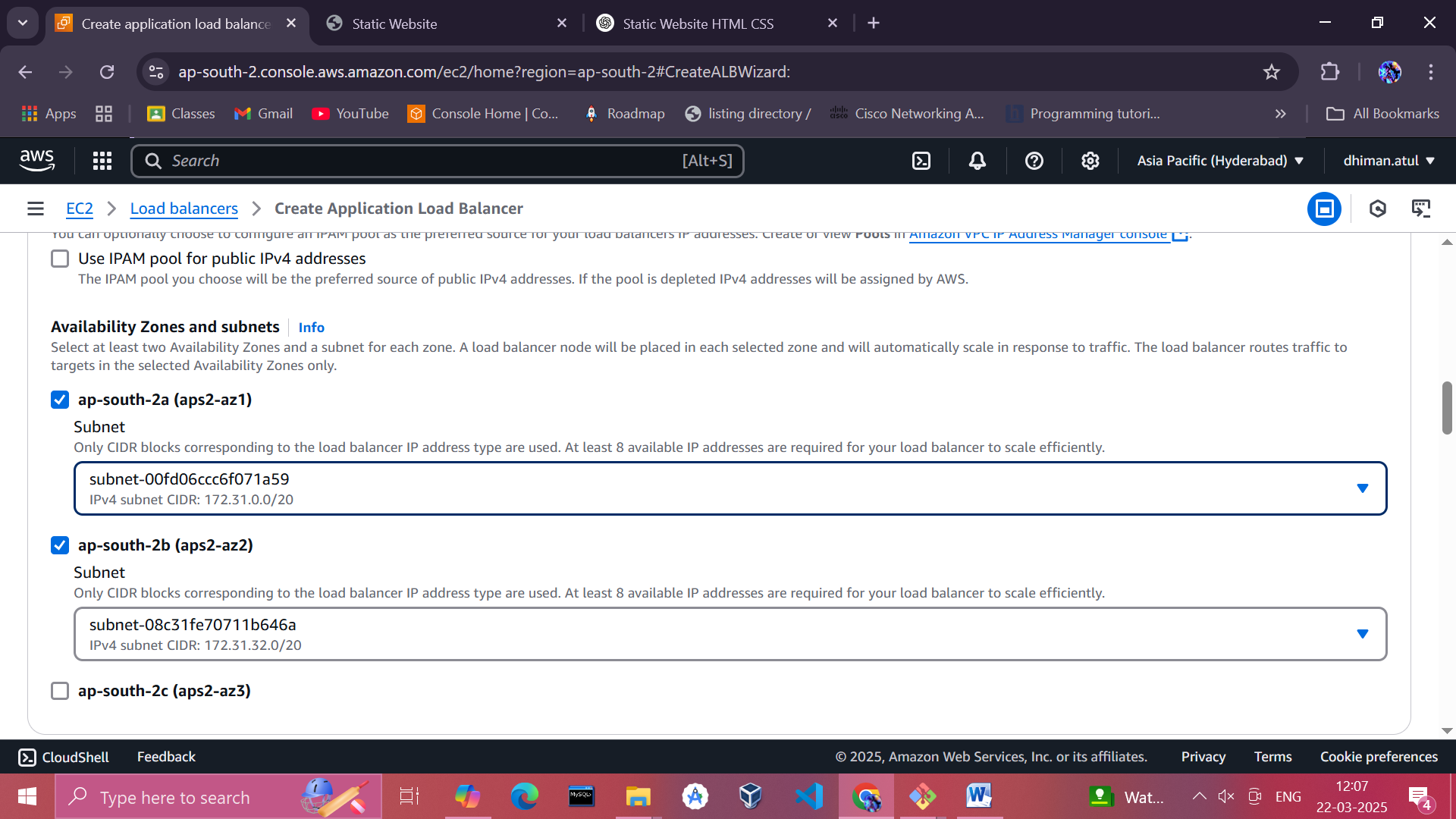


* + Deploy your web application files to the appropriate directory:
    - Apache: /var/www/html
  + Adjust permissions:
  + sudo chown -R www-data:www-data /var/www/html
  + sudo chmod -R 755 /var/www/html

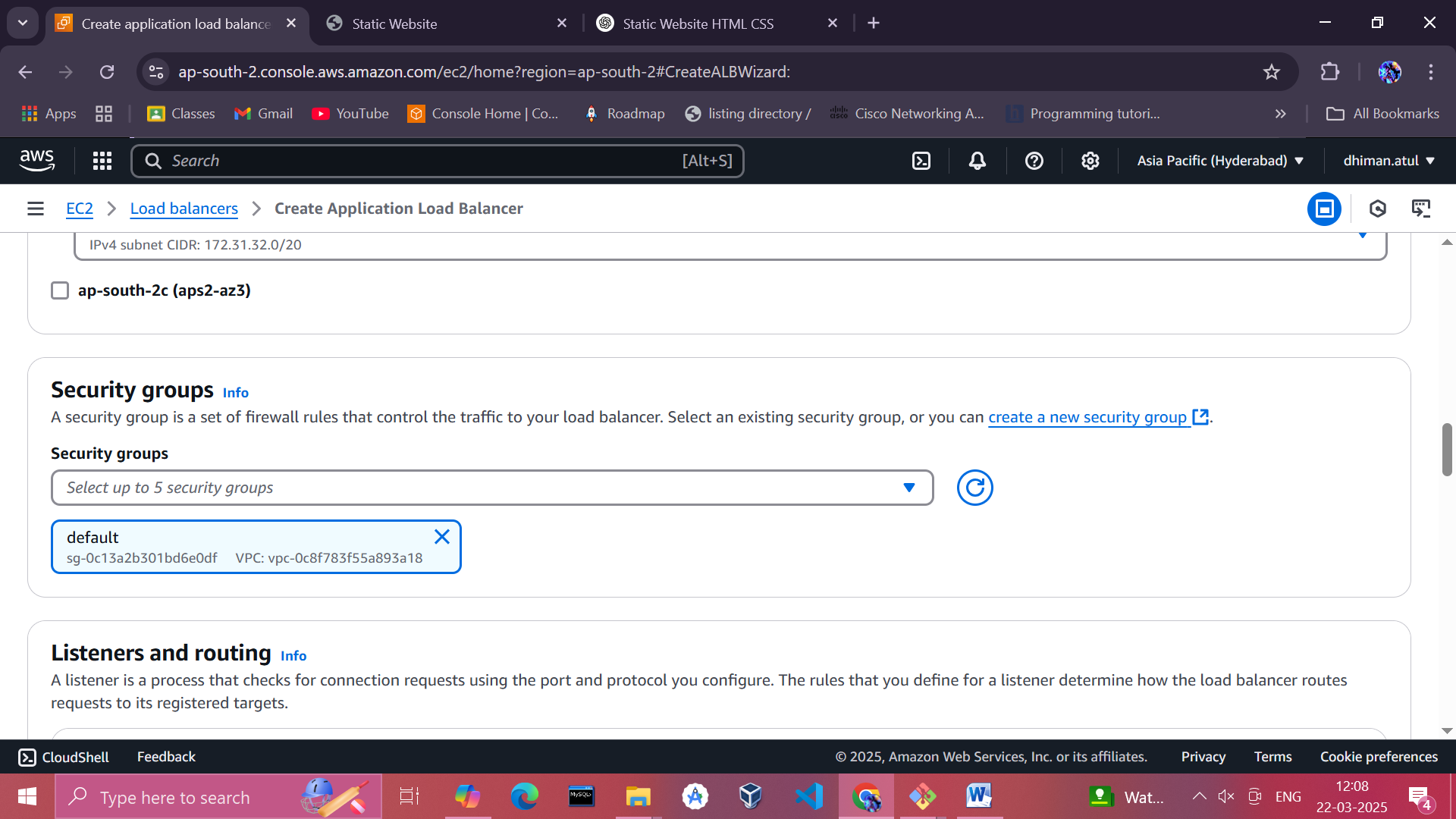


**2. Set Up Elastic Load Balancer (ELB)**

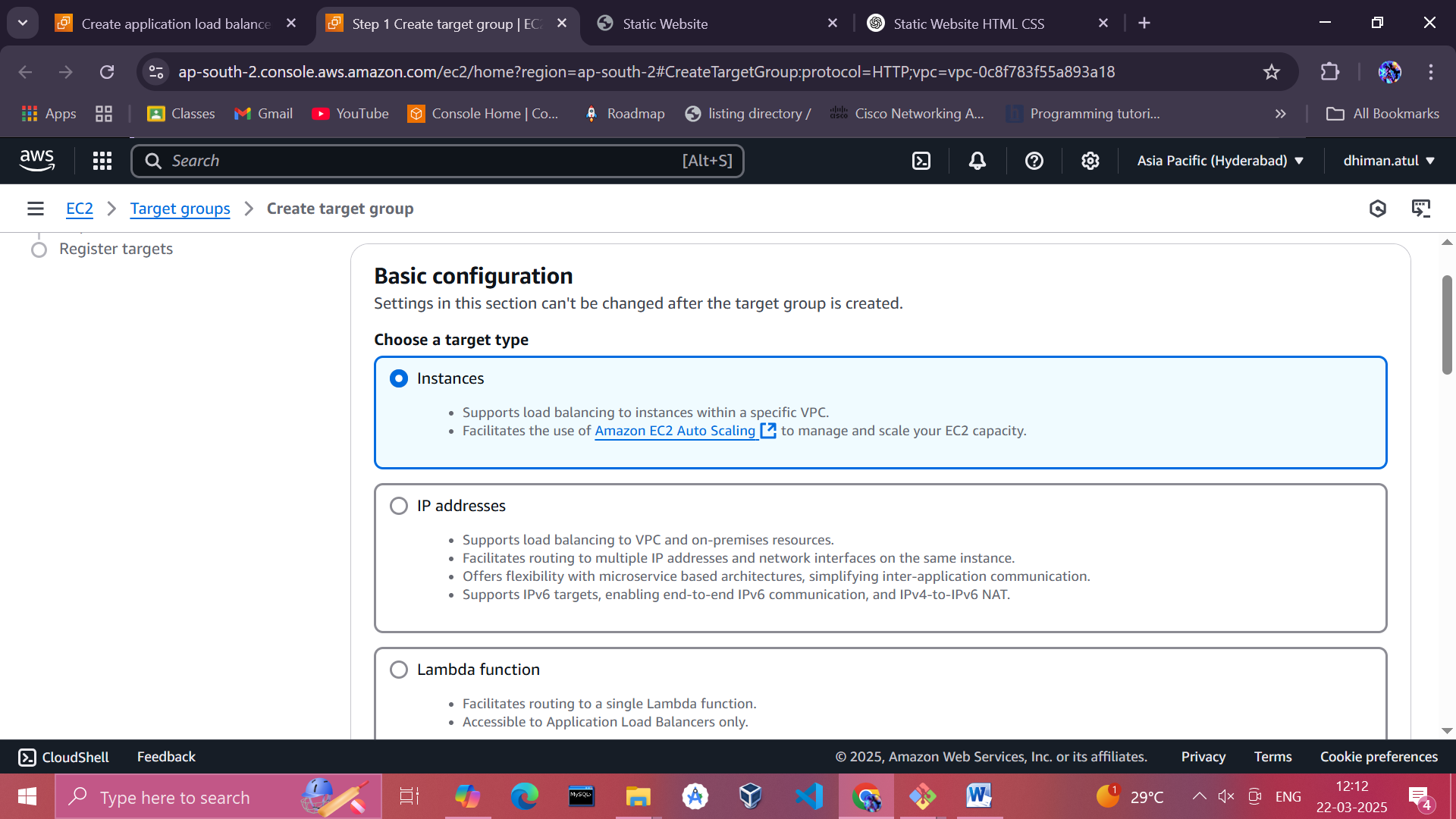
1. **Navigate to the Load Balancer Section**:
   * Go to the **EC2 Dashboard** > **Load Balancers** > **Create Load Balancer**.
2. **Configure Load Balancer**:
   * Select **Application Load Balancer**.
   * Set **Name** (e.g., WebApp-ALB).
   * Choose **Internet-facing** as the type.
   * Select two subnets in different Availability Zones.



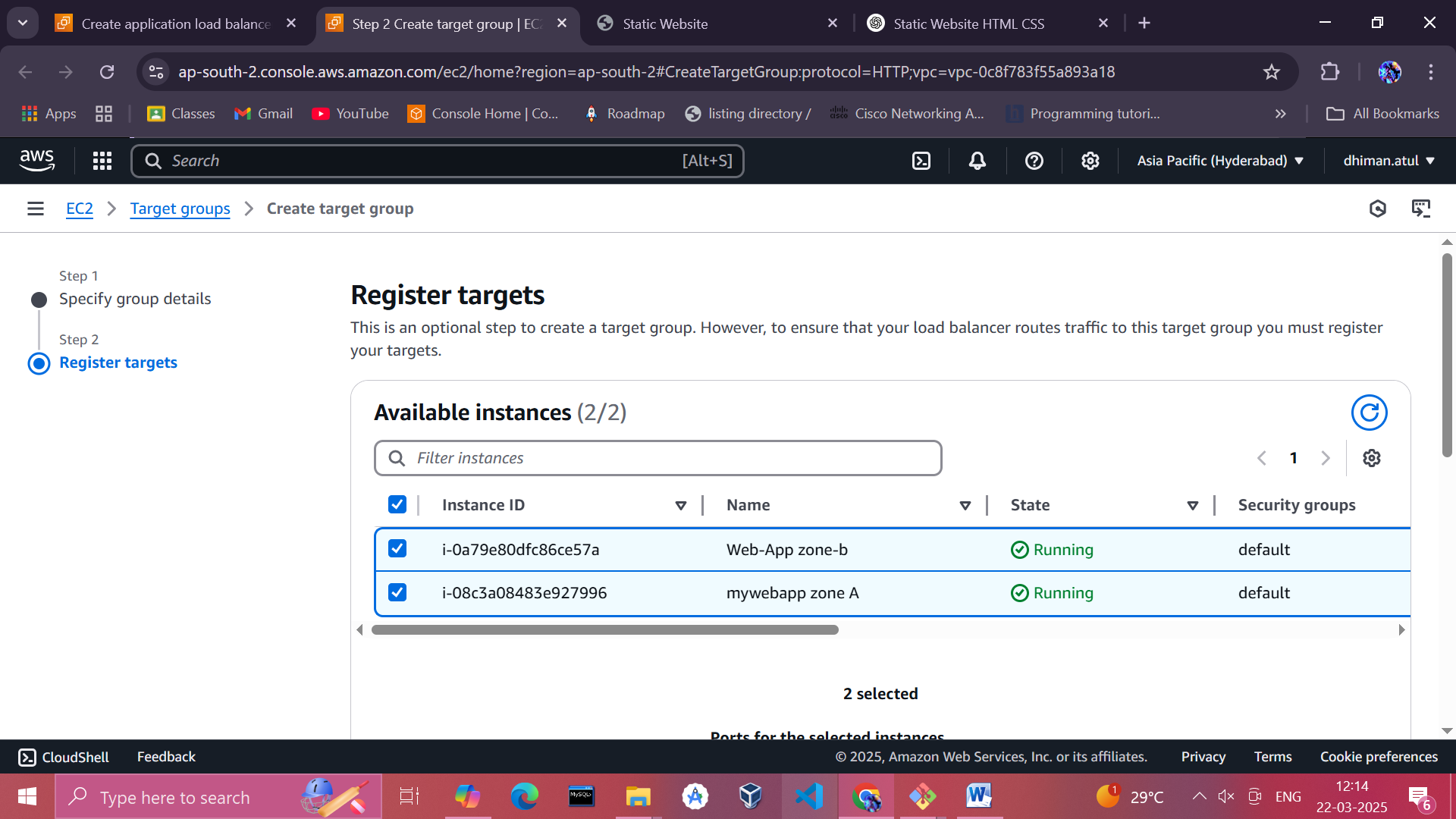
1. **Set Up Security Group**:
   * Assign a security group that allows **inbound traffic on port 80**.



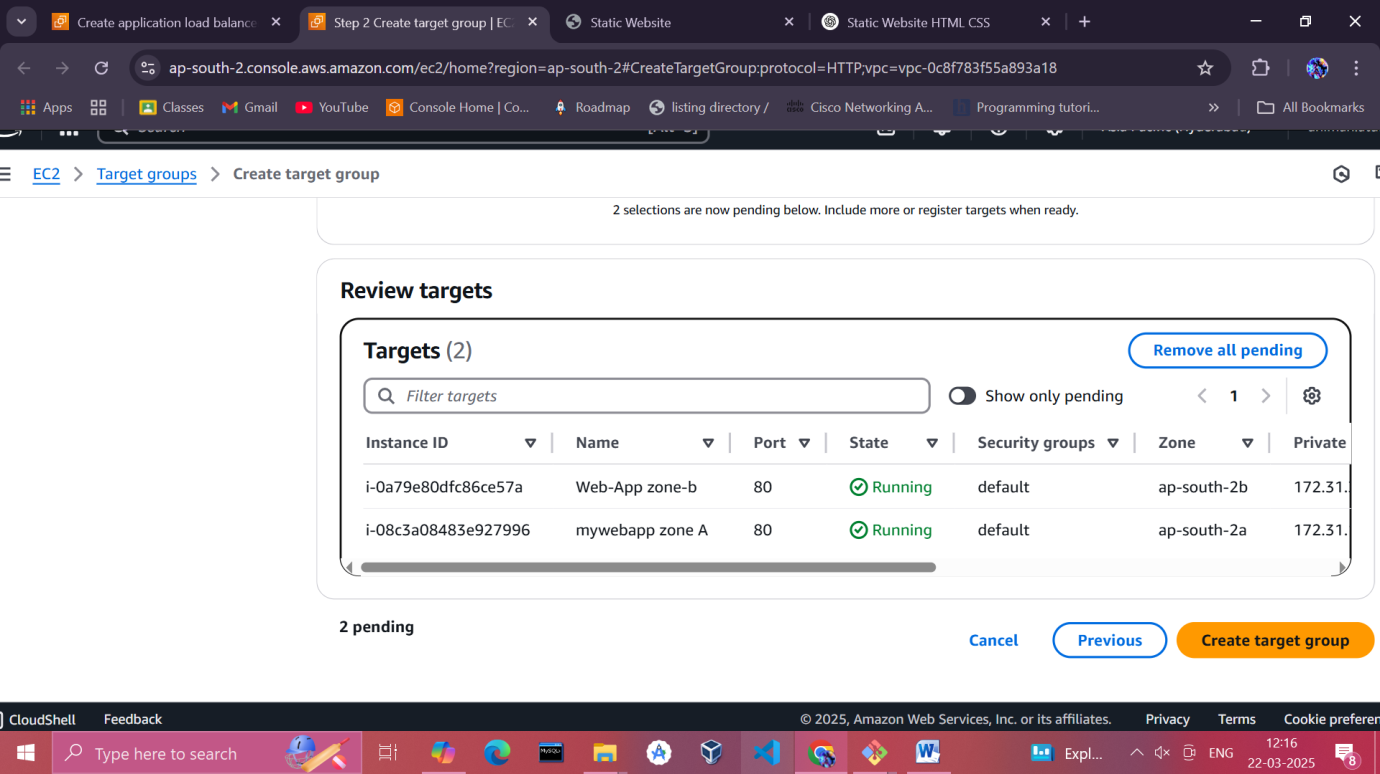
1. **Create Target Group**:
   * Set the target type to **Instance**.



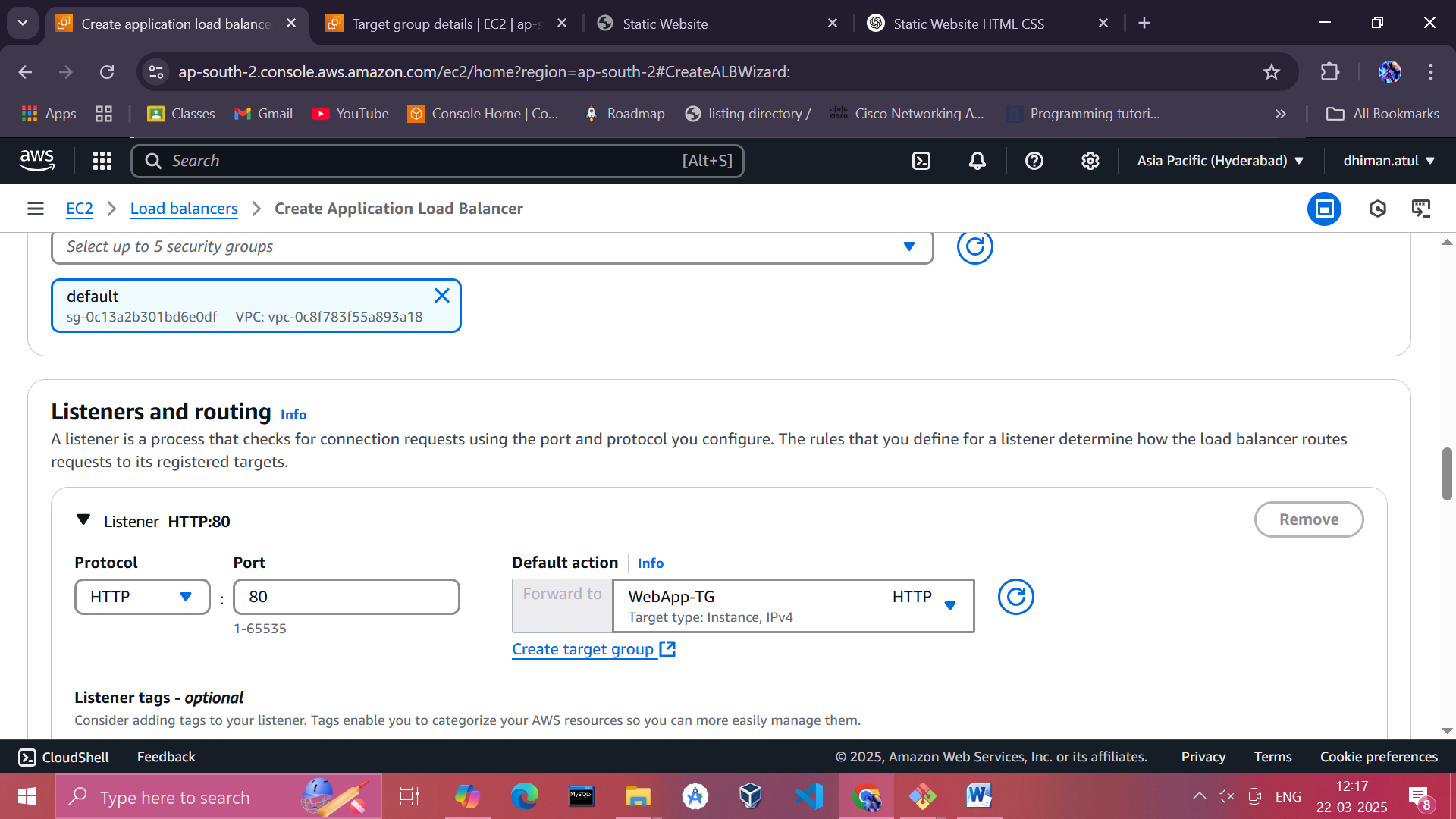
* + Set **Name** (e.g., WebApp-TG).
  + Use the default health check path (/).



1. **Register Targets**:
2. Add your EC2 instances to the target group.

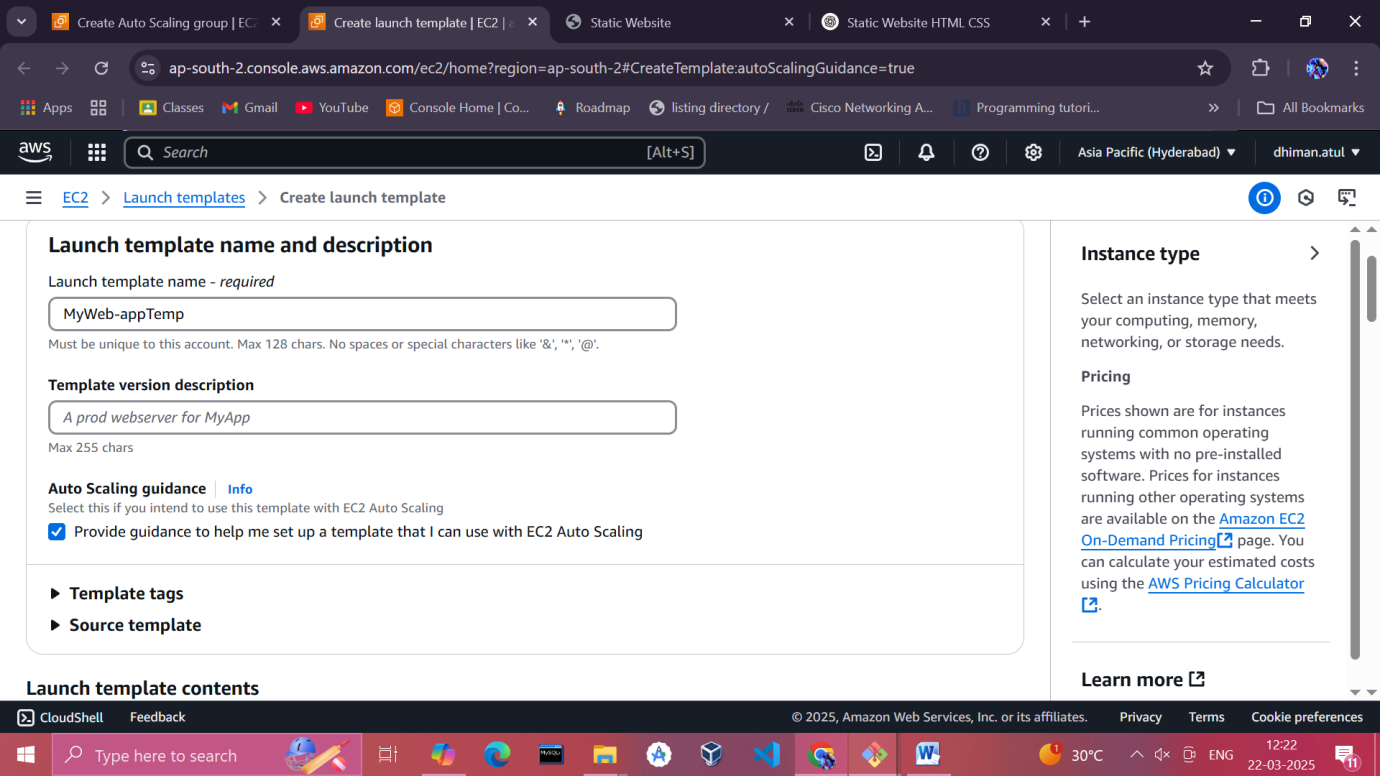


1. **Create and Verify**:
   * Complete the setup and verify the Load Balancer is distributing traffic across instances.

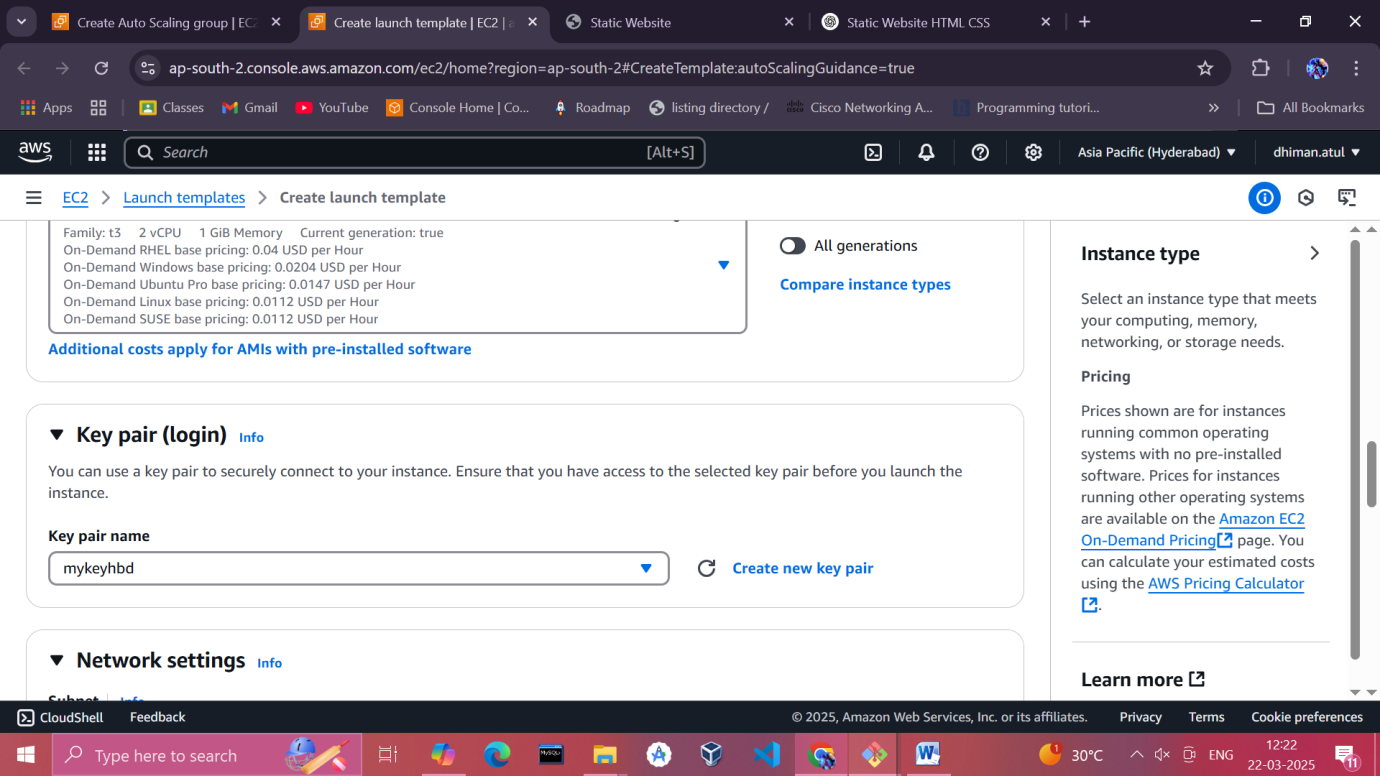


**3. Configure Auto Scaling Group (ASG)**

1. **Create a Launch Template/Configuration**:
   * Use the **Auto Scaling** section to create a launch template or configuration.
   * Specify the AMI and instance type (same as your EC2 instances).

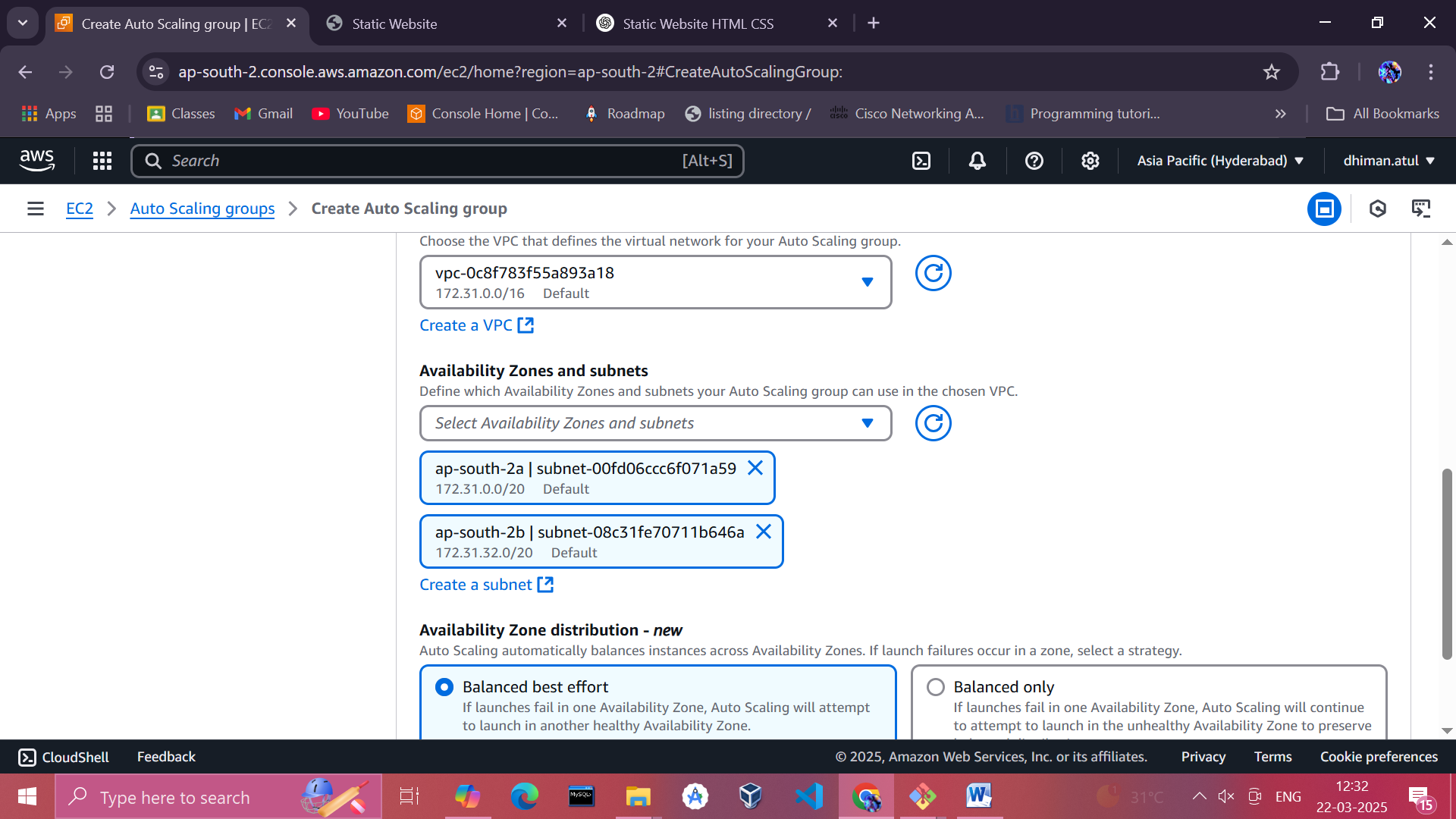


* + Set the network details (VPC and subnets).



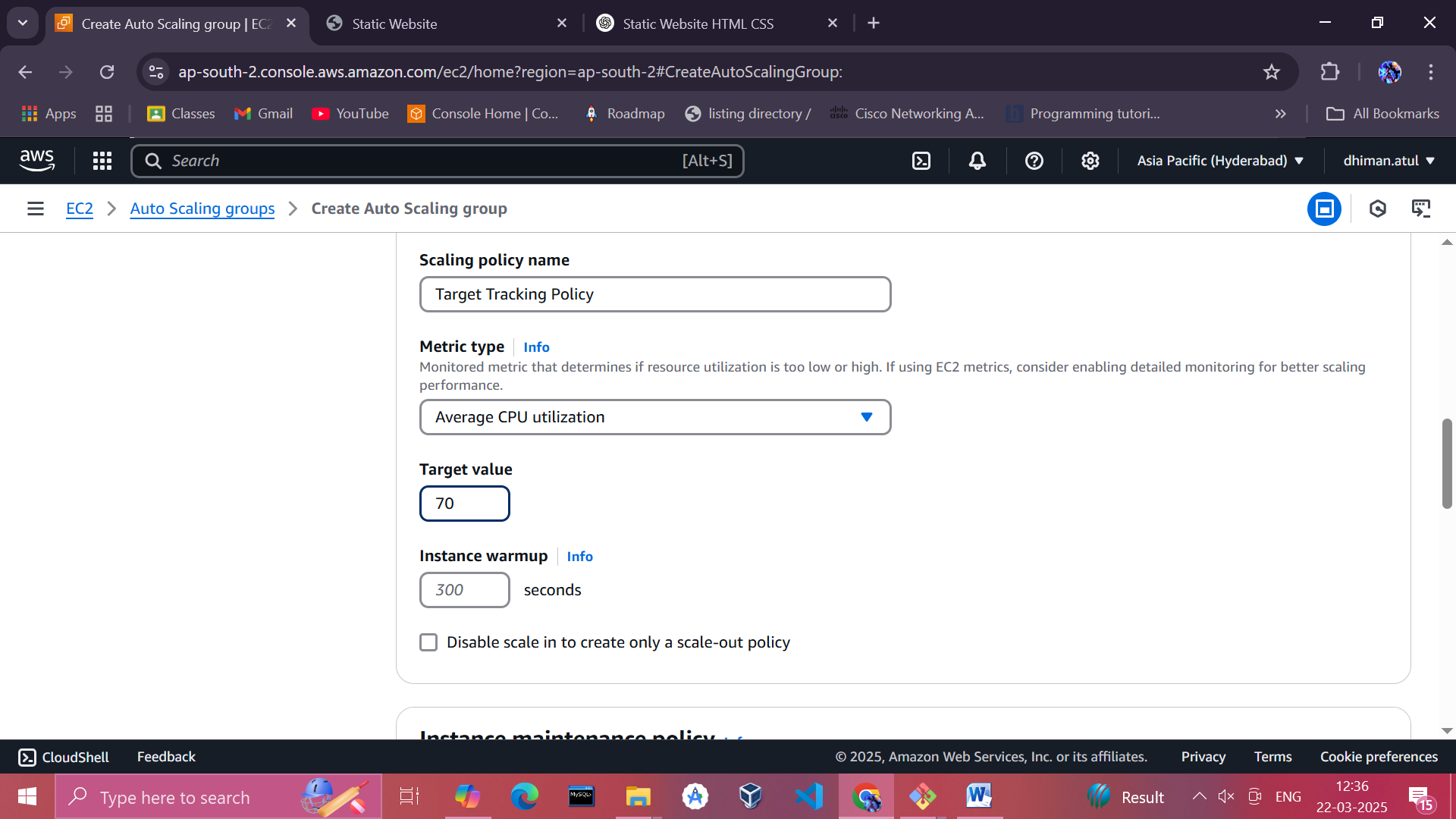
* + Attach a security group that allows access.

1. **Set Up the Auto Scaling Group**:
   * Specify the **desired capacity** (e.g., 2 instances), **minimum capacity** (e.g., 1), and **maximum capacity** (e.g., 4).



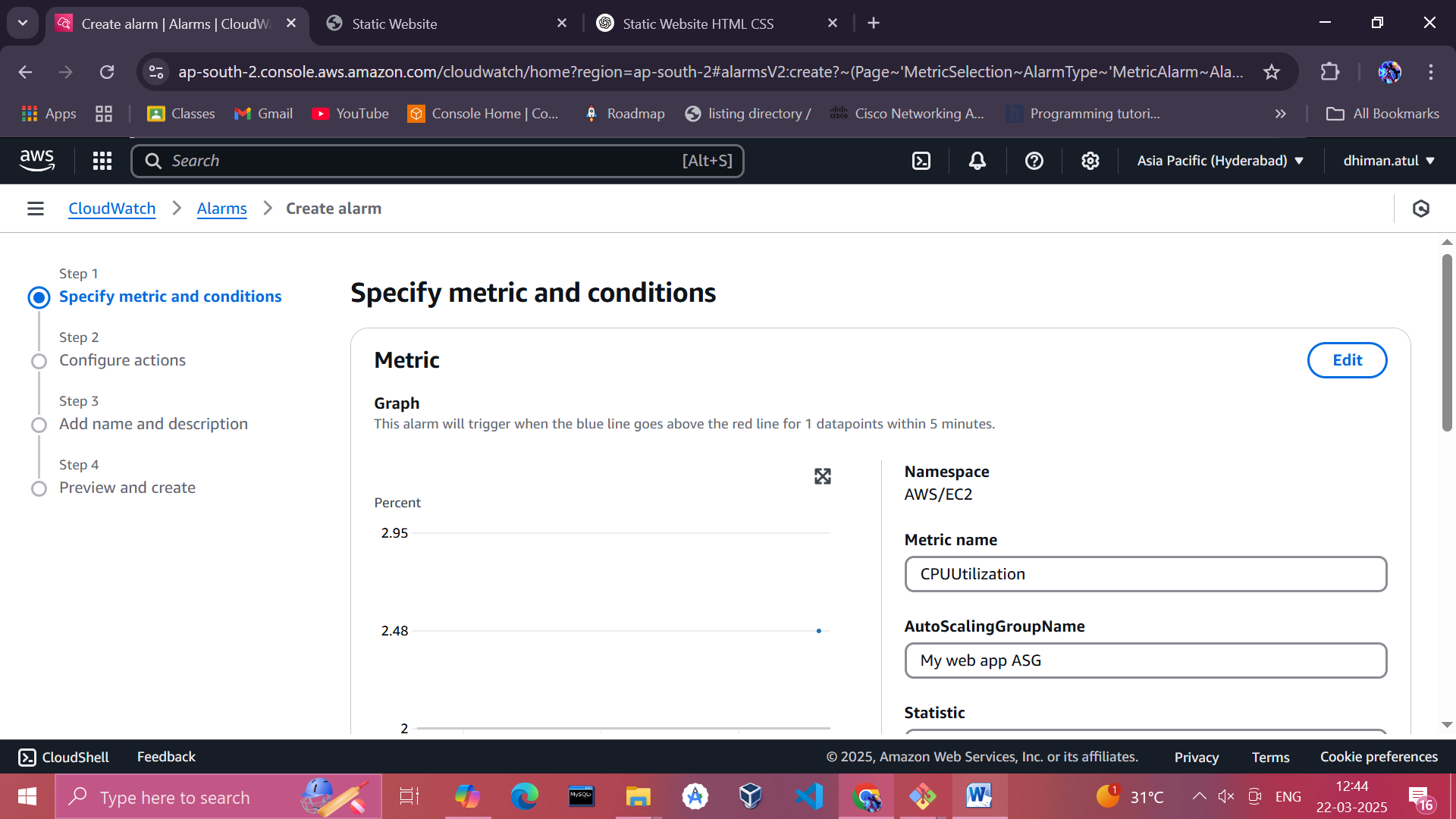
* + Choose two subnets in different Availability Zones.

1. **Add Scaling Policies**:
   * Add a scaling policy to automatically adjust the number of instances based on CPU utilization or another metric:
     + Target tracking example:
       - Metric: **Average CPU Utilization**
       - Target value: **70%**



**4. Set Up CloudWatch Monitoring**

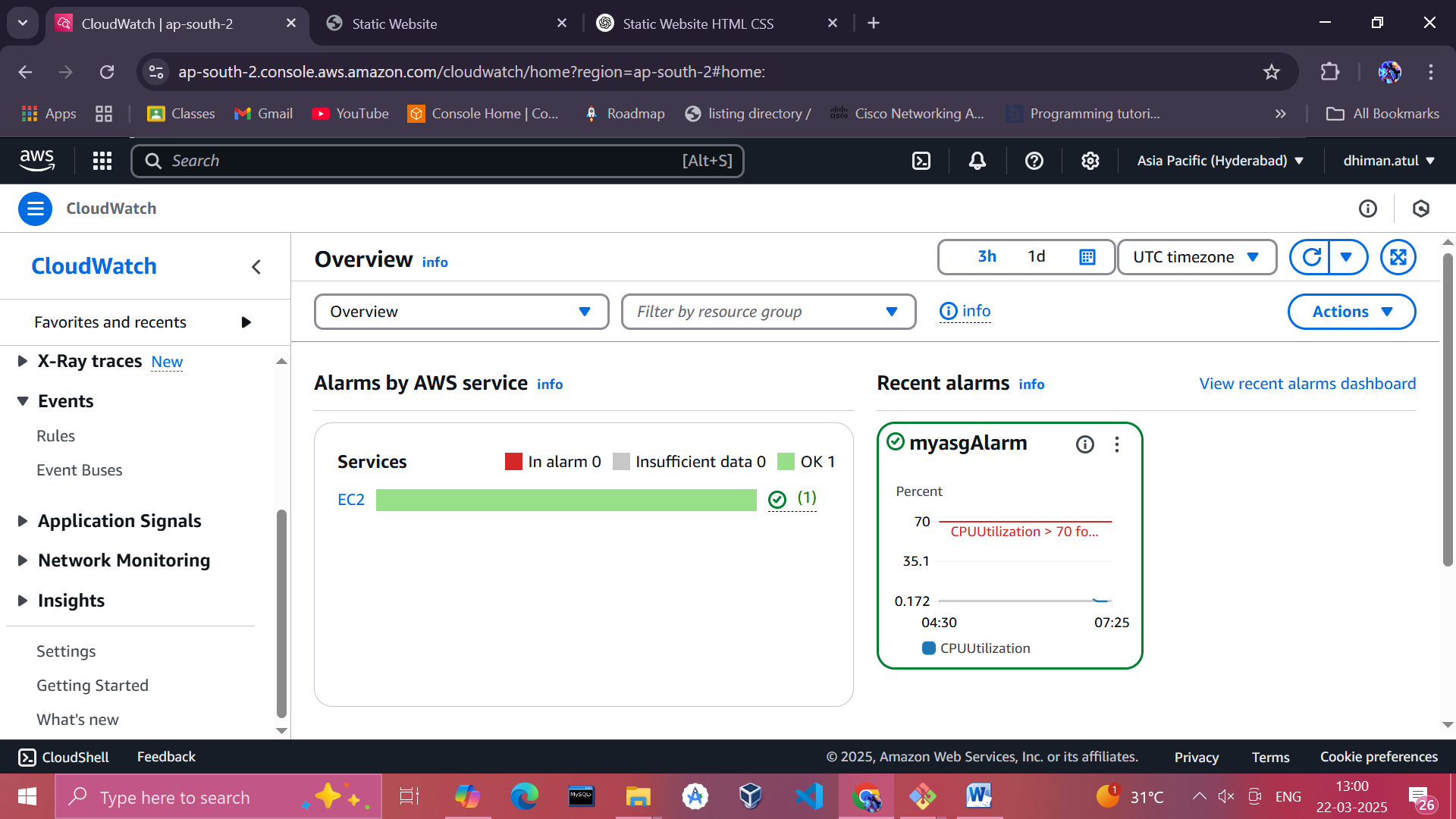
1. **Create CloudWatch Alarms**:
   * Navigate to the **CloudWatch Dashboard** > **Alarms** > **Create Alarm**.
   * Select the **metric** for your Auto Scaling group (e.g., **CPU Utilization**).
   * Define the threshold (e.g., **CPU Utilization > 70%**) and the action to trigger scaling.



**5. Verify Auto Scaling and Load Balancer**

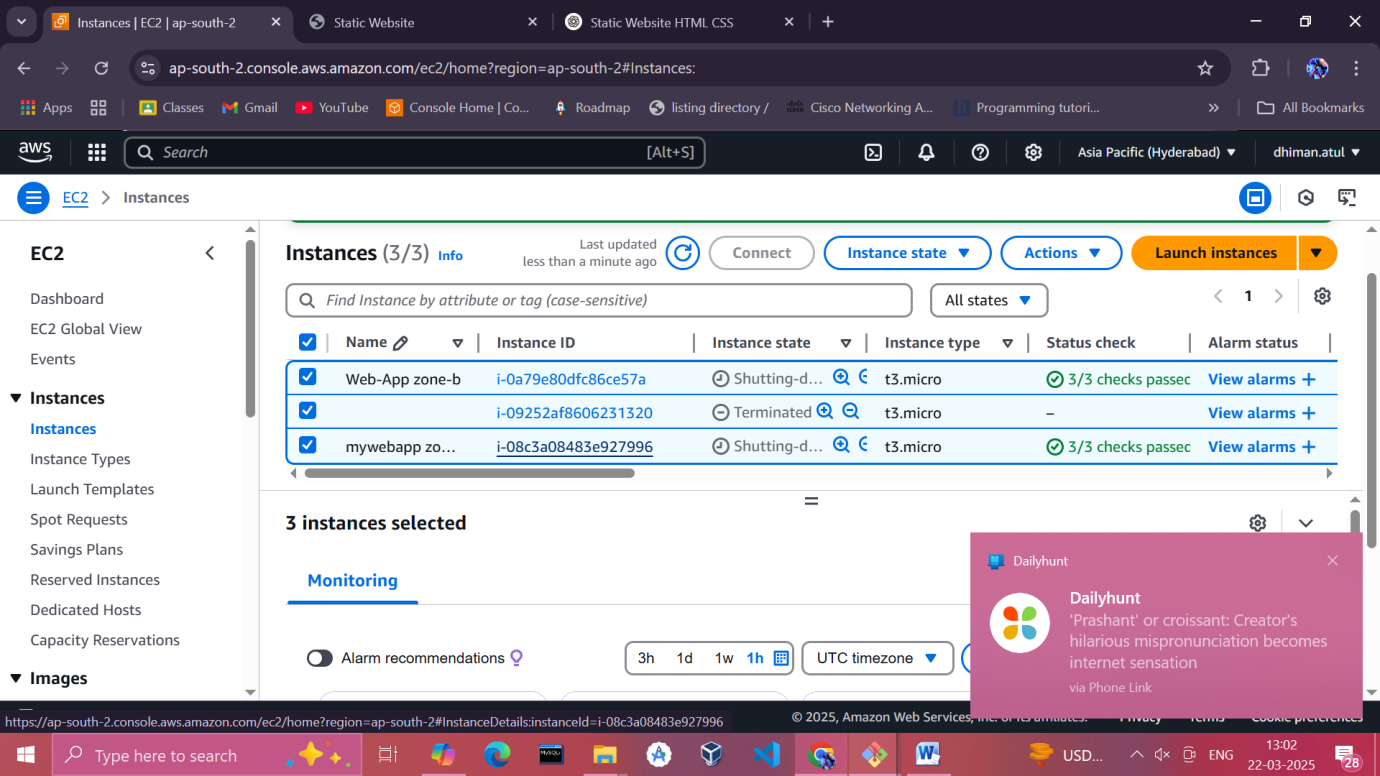
**1. Check CloudWatch Metrics:**

* Navigate to the **CloudWatch Metrics** section.
* Observe metrics like **CPU Utilization** or any custom metric you've configured.
* Confirm if the scaling activities are triggered when the defined thresholds (e.g., 70% CPU utilization) are reached.



**2. Review Scaling Activities:**

* Go to the **Auto Scaling Group** section in the AWS Management Console.
* Under the **Activity History** tab, check for records of scaling actions (e.g., launching or terminating instances).
* This provides real-time logs of what Auto Scaling is doing and why.



Observation: we can see a third instance created because of the Auto-Scaling group

**3. Simulate System Load with Stress Tools:**

* Install a stress tool like stress-ng on your EC2 instances.
* Run commands to artificially increase CPU usage, triggering Auto Scaling:
* sudo apt install stress-ng -y
* stress-ng --cpu 4 --timeout 60s

