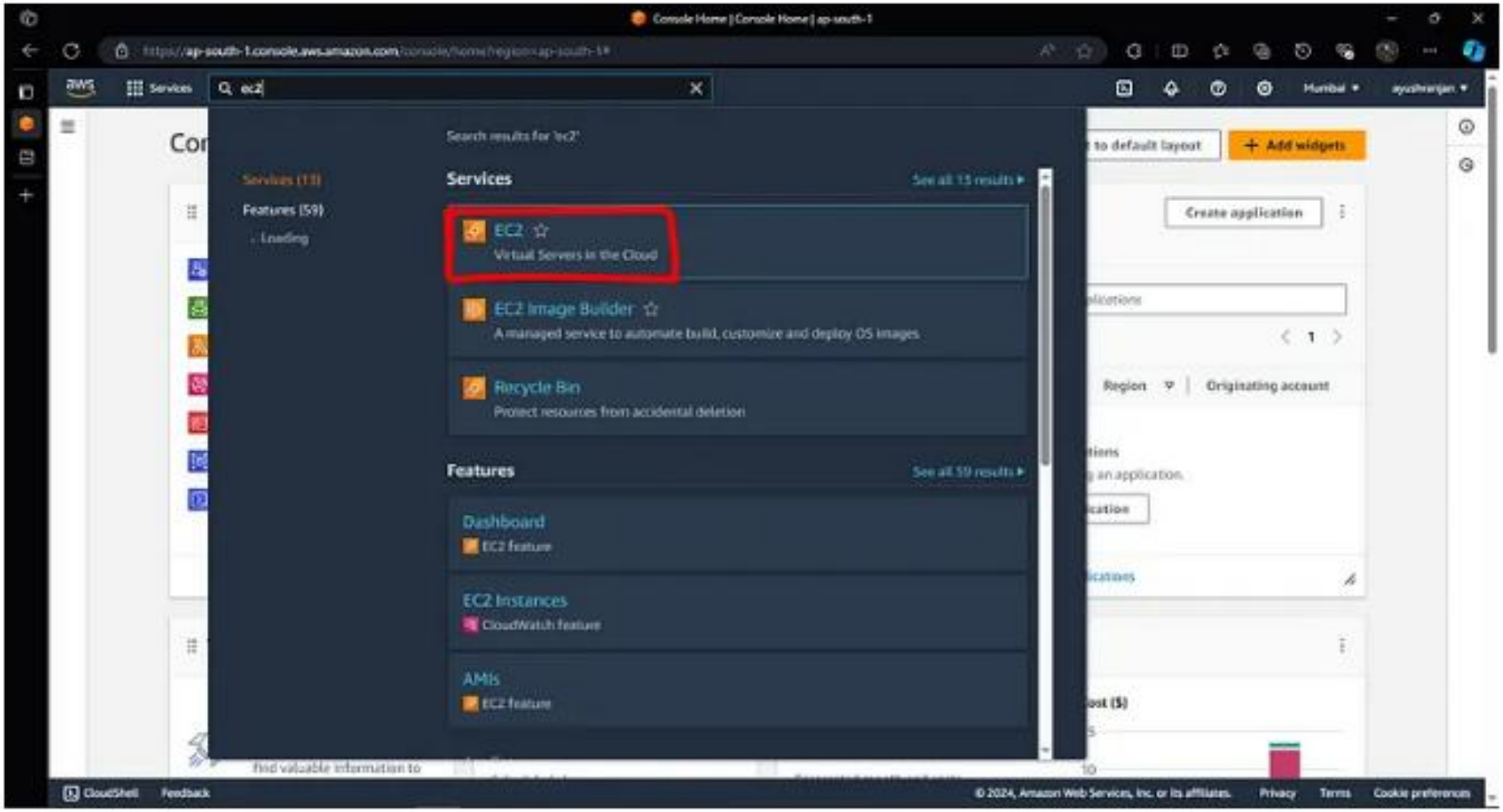
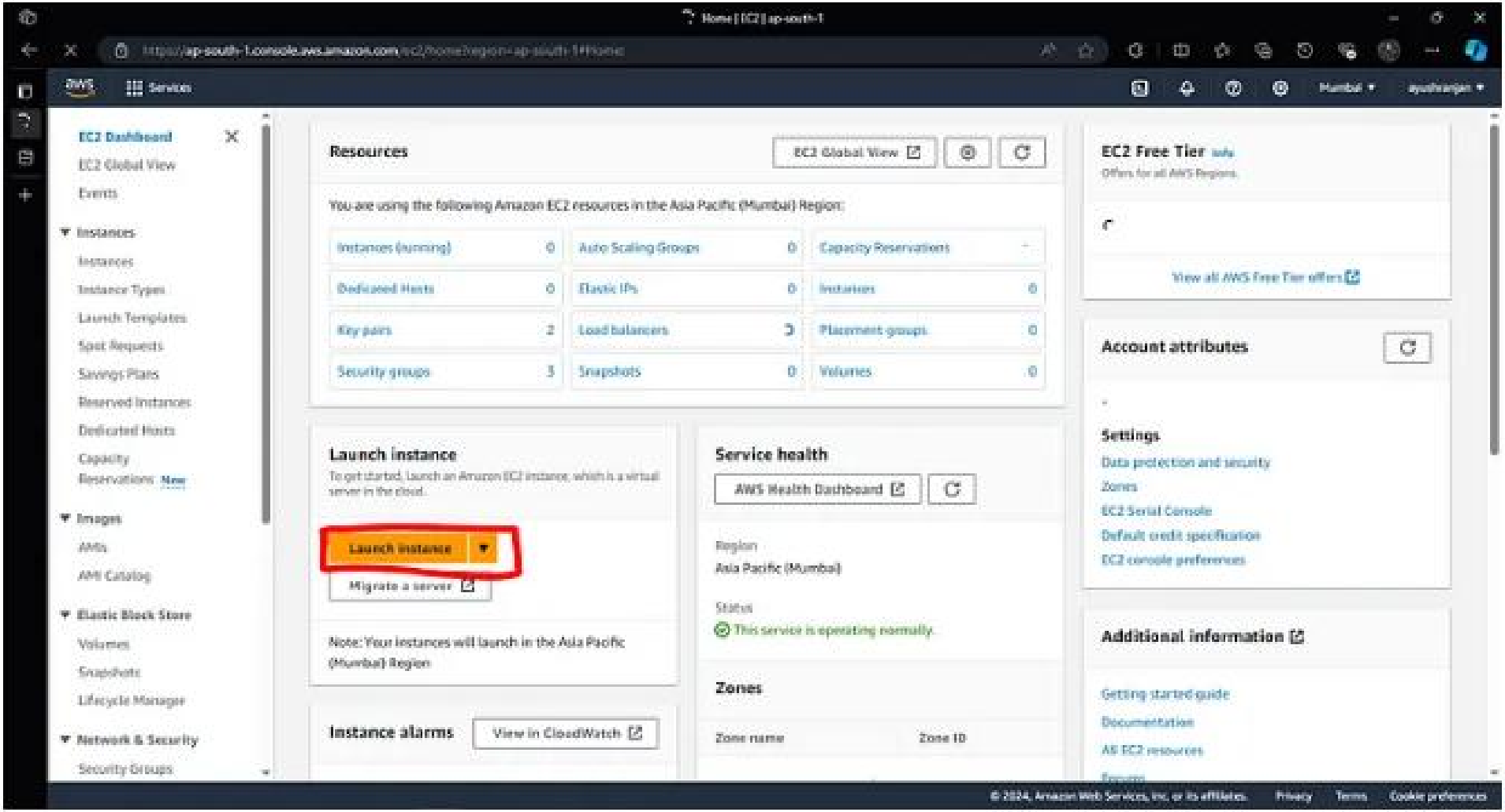
1. Login to AWS Management Console

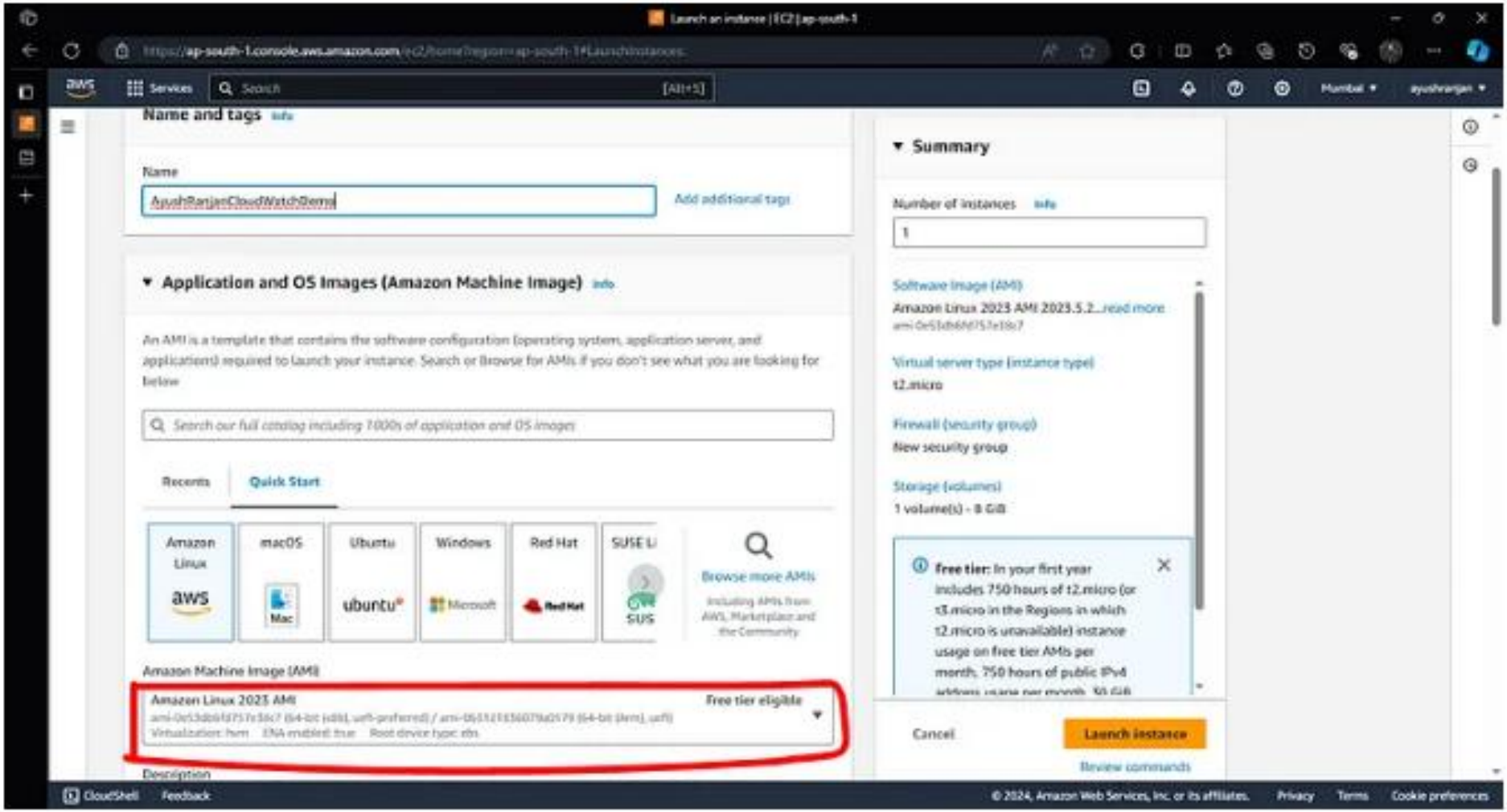
- Go to the AWS Management Console
- Search for **EC2** and click on the service.



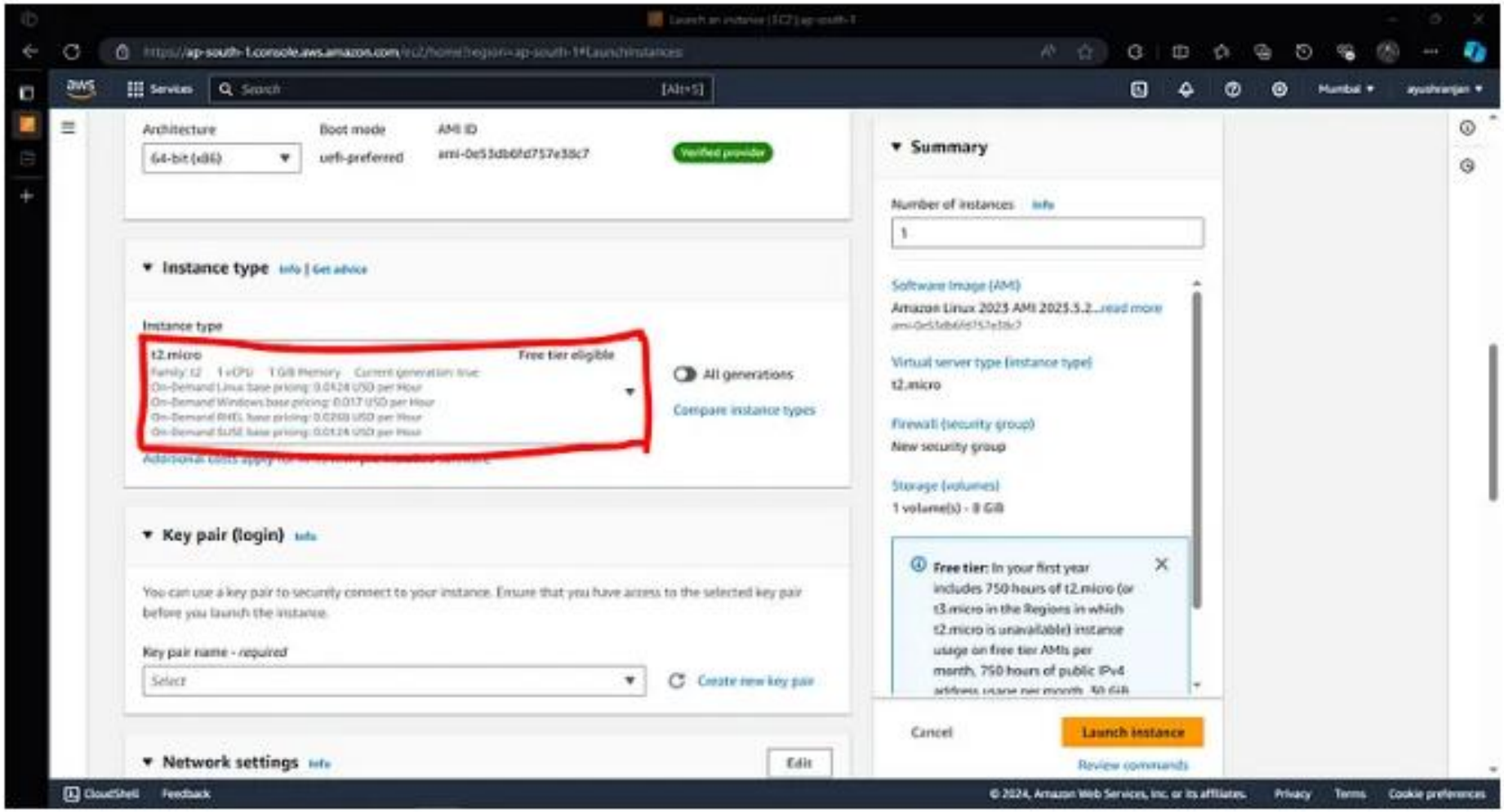
2. Launch an Instance

- In the EC2 Dashboard, click on the **Launch Instance** button.

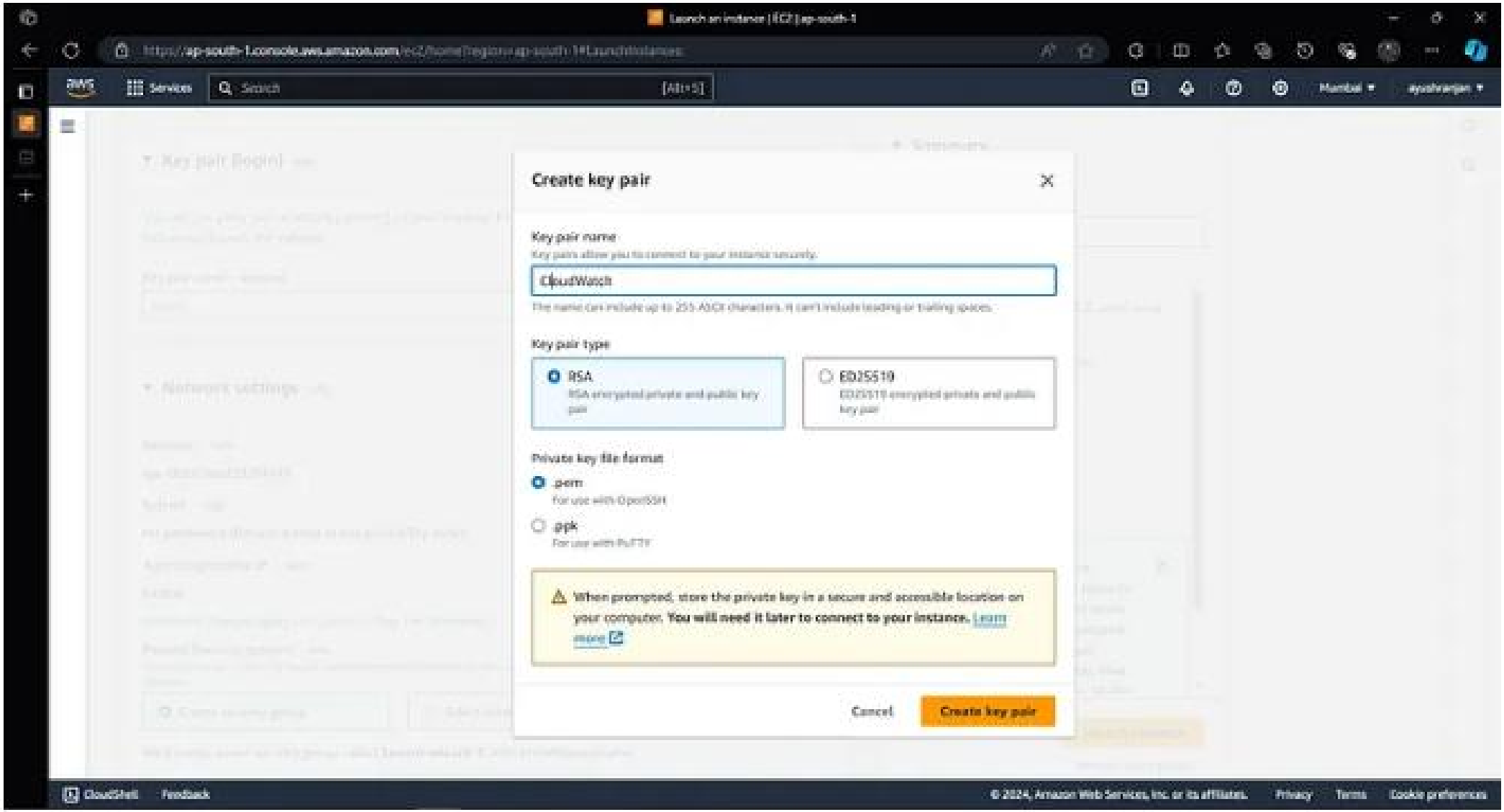



Launch an instance | EC2 | ap-south-1
Services
Search
[Alt+S]
Mumbai
ayushranjan
Name and tags Info
Name
AyushRanjanCloudWatchDemo
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 AMI
Free tier eligible
Description
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

- Choose **Instance Type:**
  Select **t2.micro** (Free Tier eligible) which provides 1 vCPU and 1 GB of memory.



- Create a key pair

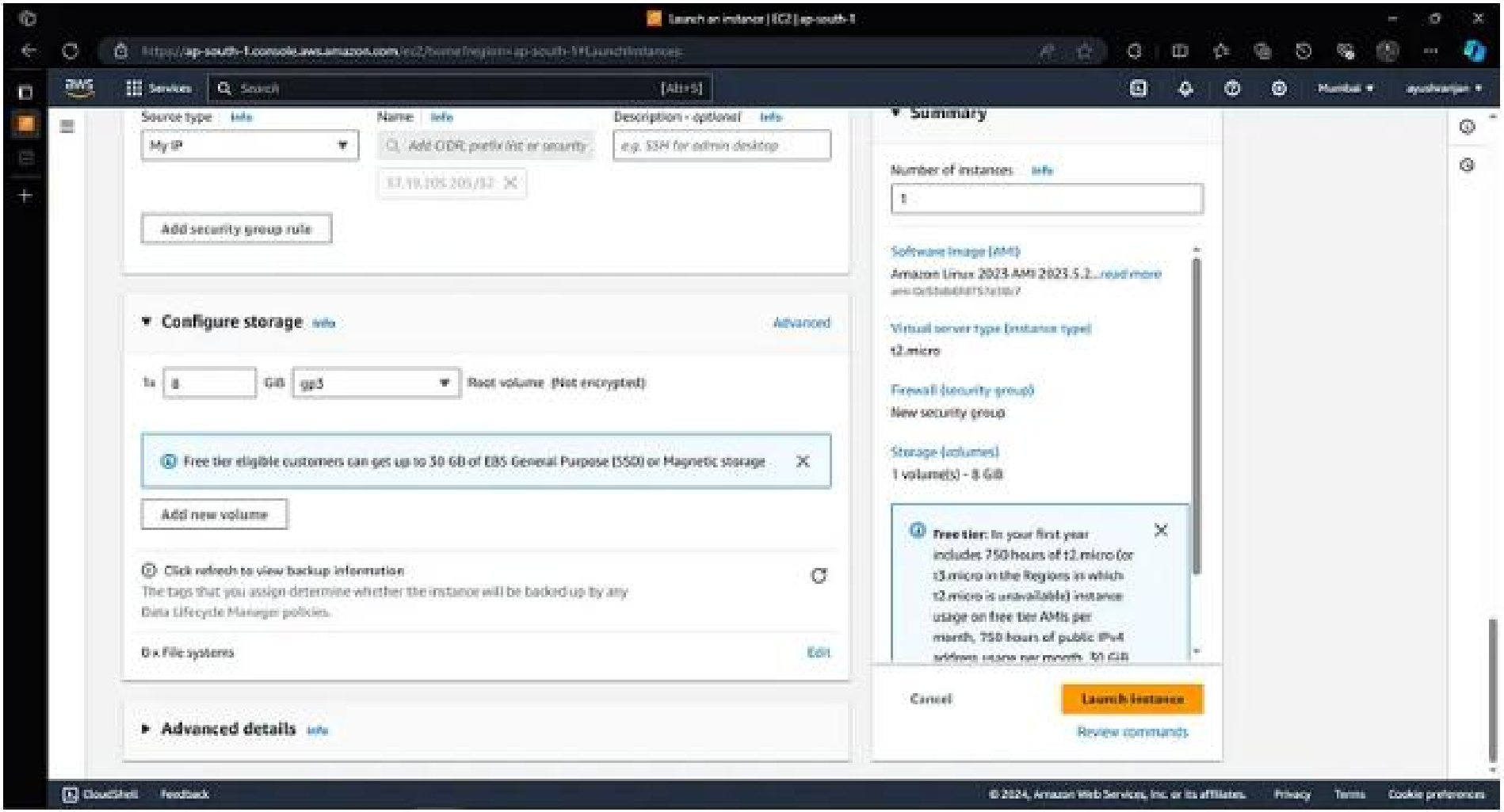


3. Configure Instance Settings

- **Network:** Use the default VPC and subnet.
- **Auto-assign Public IP:** Ensure it is set to “Enable”.
- **IAM Role:** Leave this empty for now, as it’s optional for this project.

4. Configure Storage

- Leave the default of 8 GB SSD (General Purpose), as this is free-tier eligible.

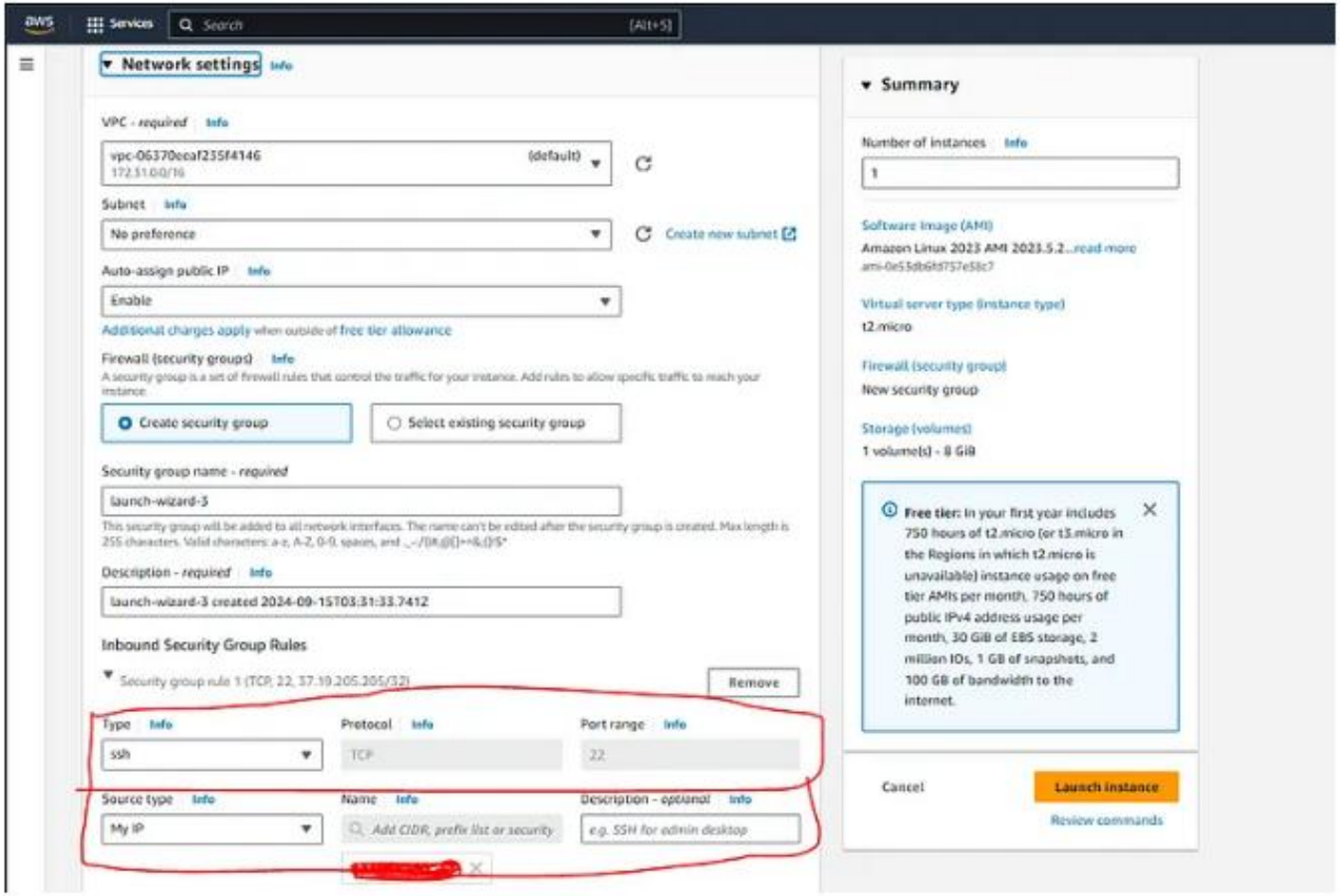


5. Add Tags

- Tags can help organize your resources, but you can skip this for now.

6. Configure Security Group

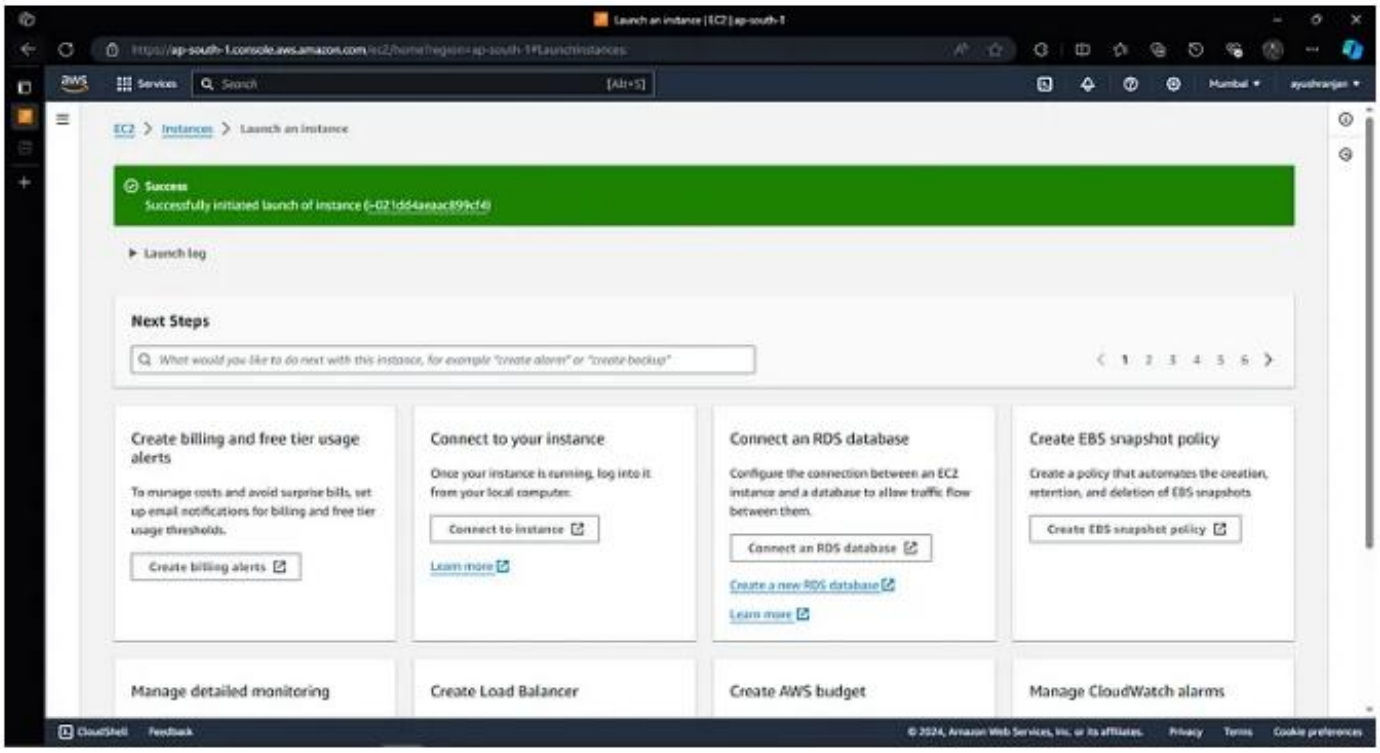
- Create a new security group and allow **SSH** (port 22) from your IP address.

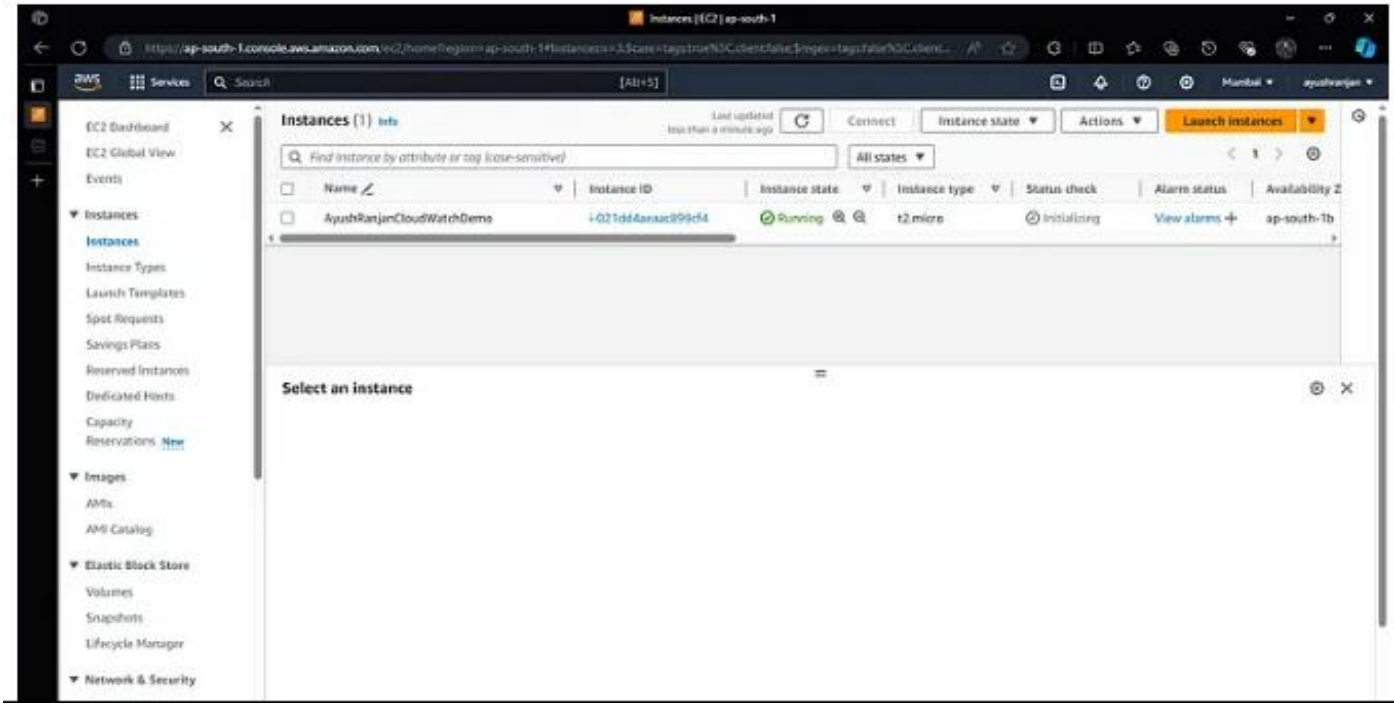


- Add **HTTP** (port 80) if you plan to host a web application, but this isn't necessary for our CloudWatch project.

7. Launch Instance

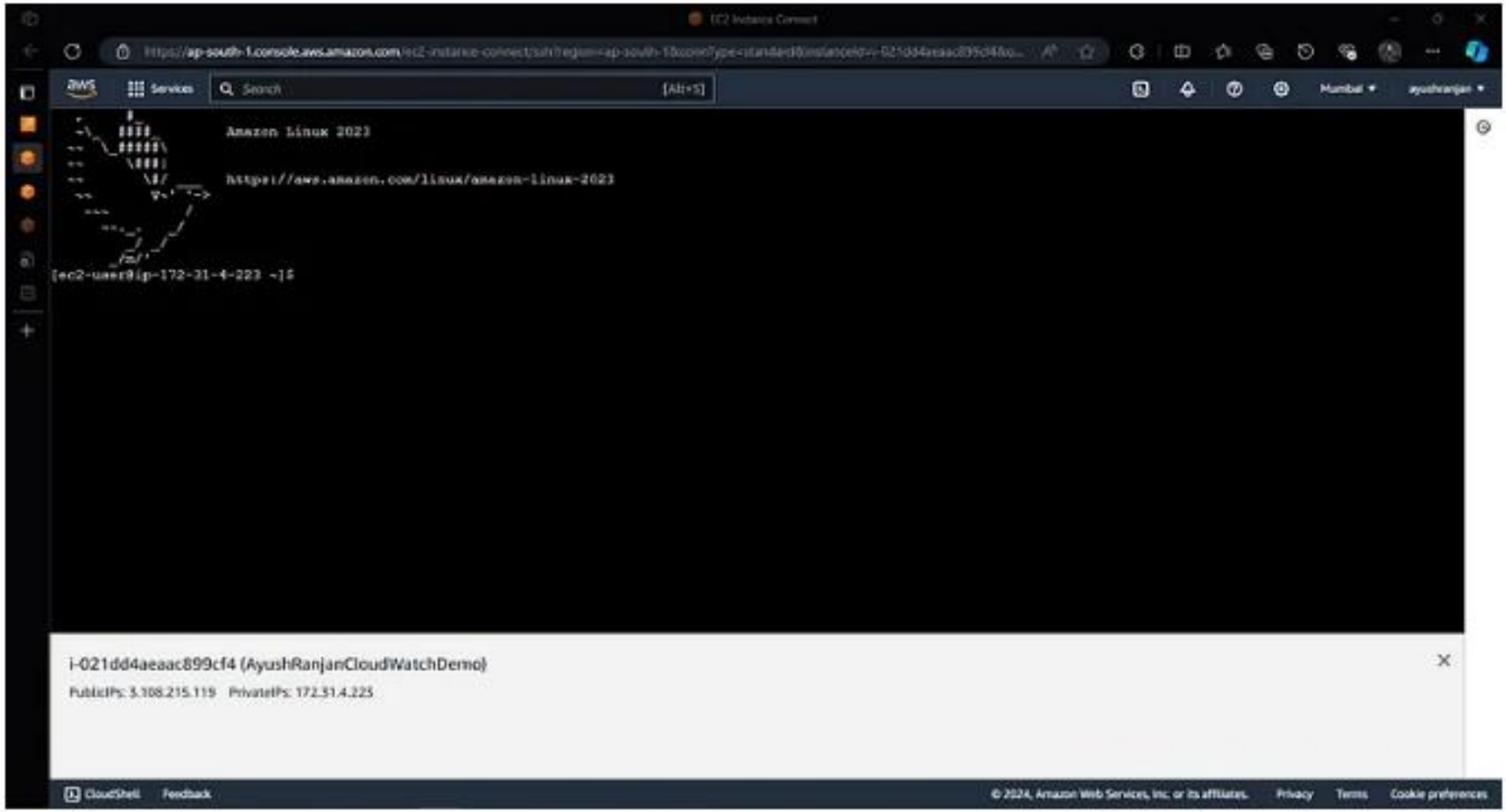
- Finally, click on **Launch Instance.**





8. Access Your EC2 Instance

- Once the instance is running, you can SSH or connect to instance using **EC2 Instance Connect**

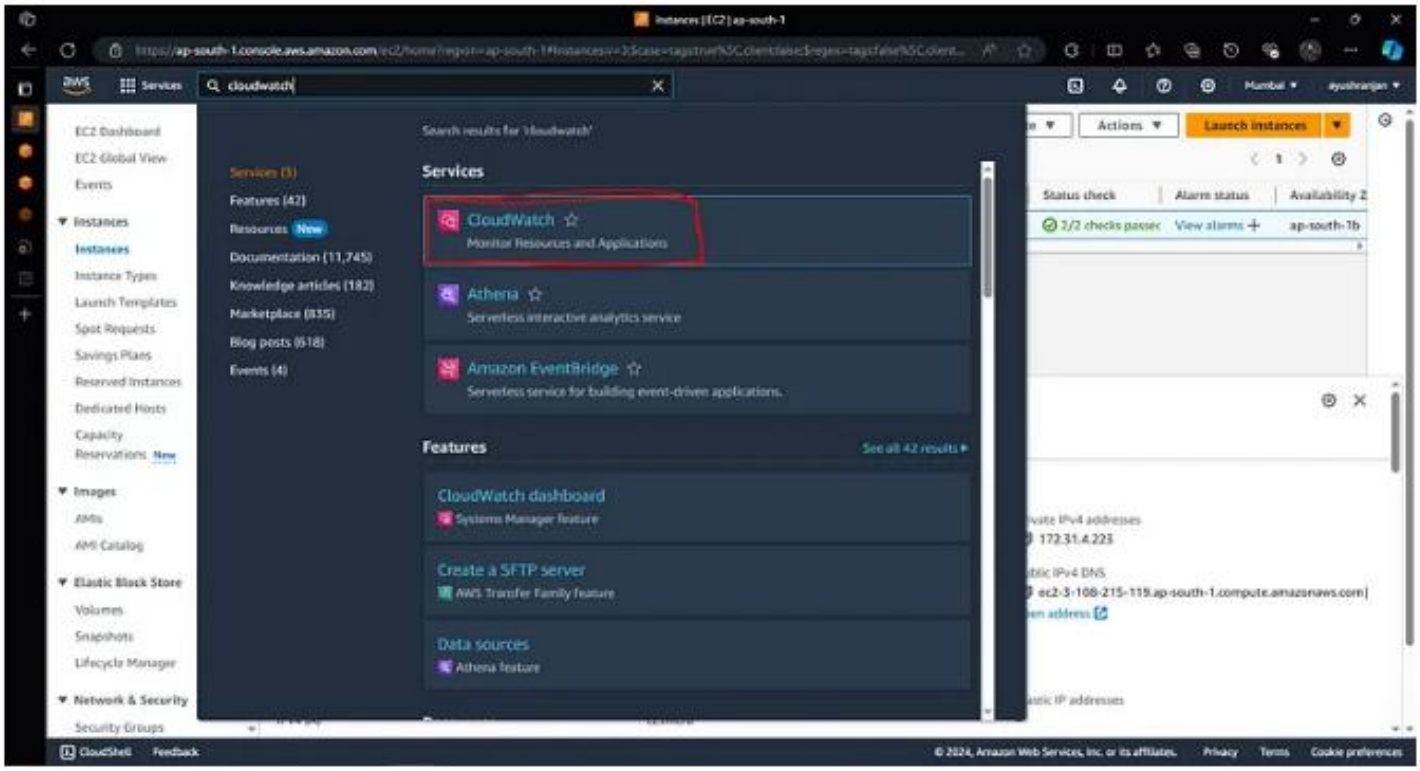


Step 2: Enable Basic CloudWatch Monitoring

By default, CloudWatch provides basic monitoring for EC2 instances, including metrics like CPU utilization, network in/out, and disk I/O. This data is collected at 5-minute intervals and is completely free within the AWS Free Tier.

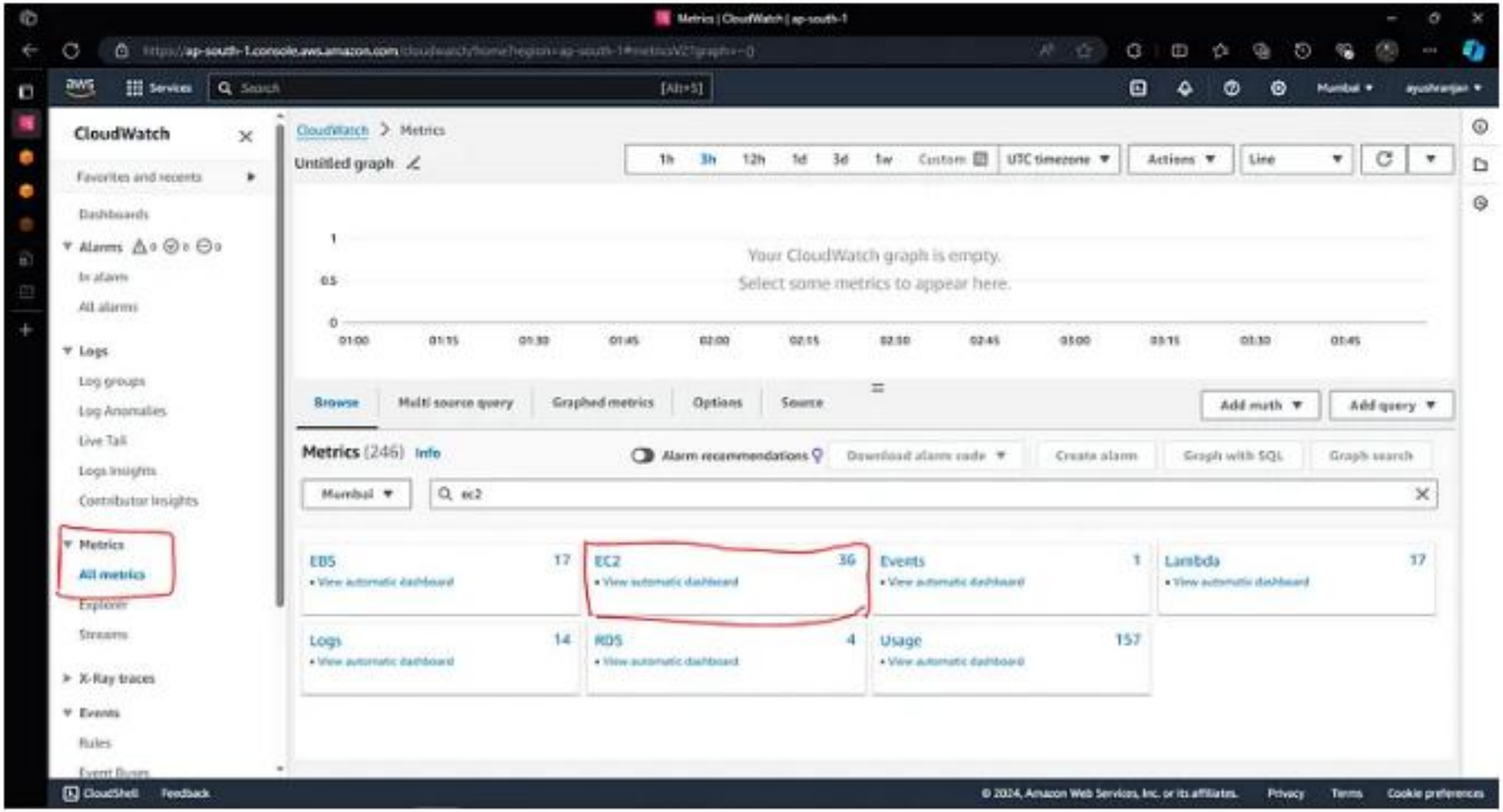
1. Navigate to the CloudWatch Console

- In the AWS Management Console, search for **CloudWatch** and click on the service.

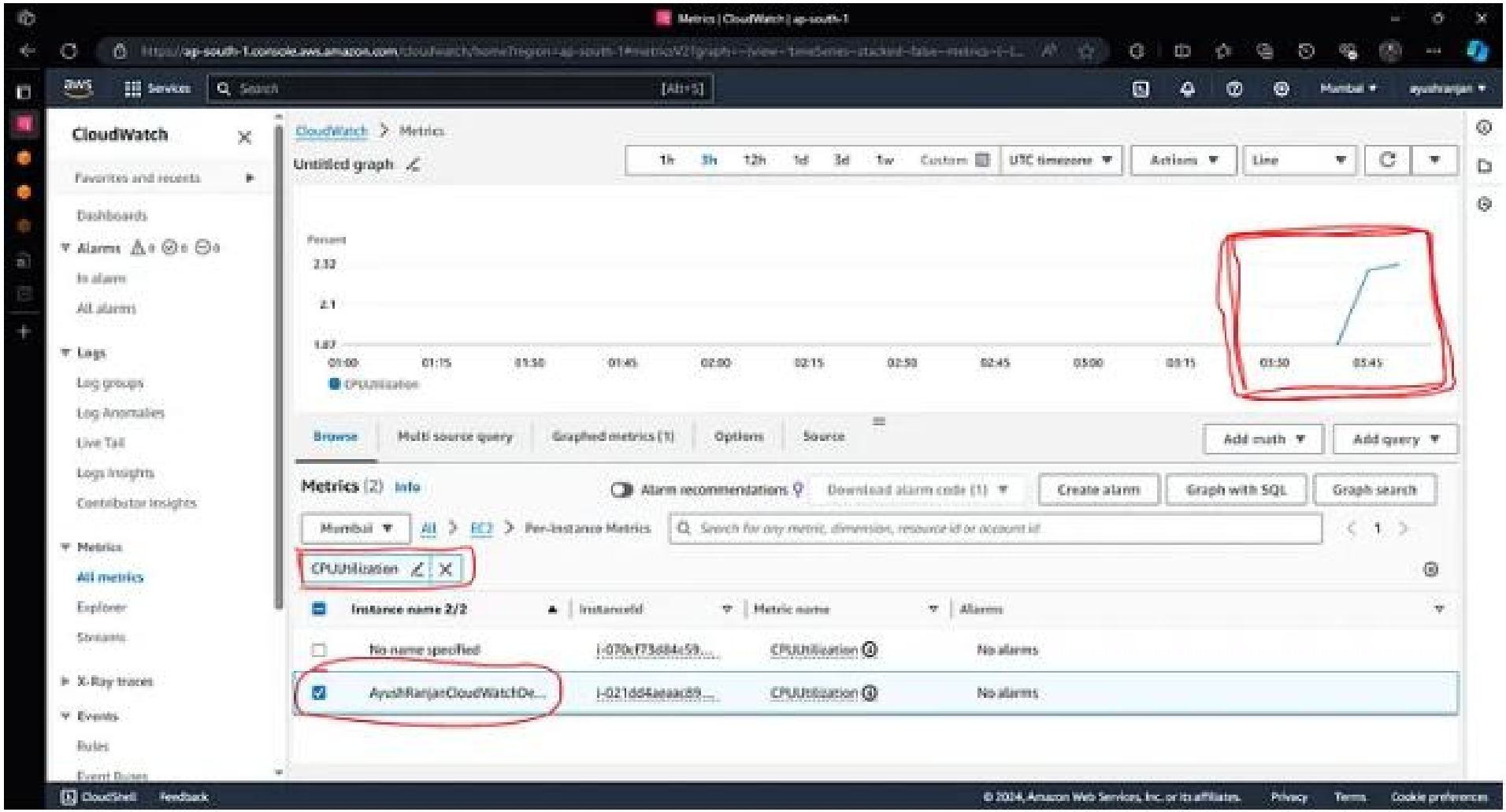


2. Check Default Metrics

- On the CloudWatch Dashboard, click on **Metrics** in the left-hand menu. In the **Browse** tab, click on **EC2** under the list of services.



- You will see metrics for your running EC2 instance, including **CPUUtilization, NetworkIn, and NetworkOut.**



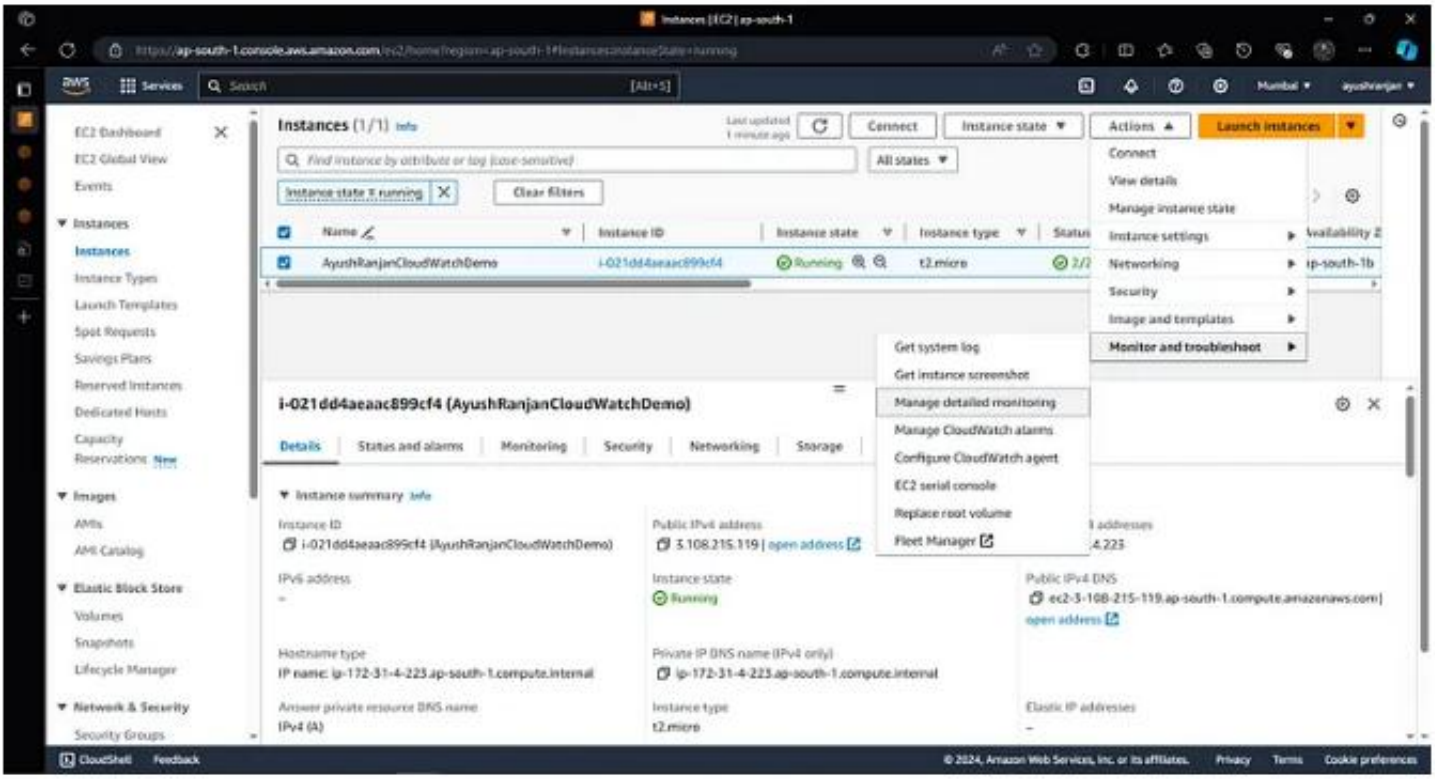
Step 3: Enable Detailed Monitoring

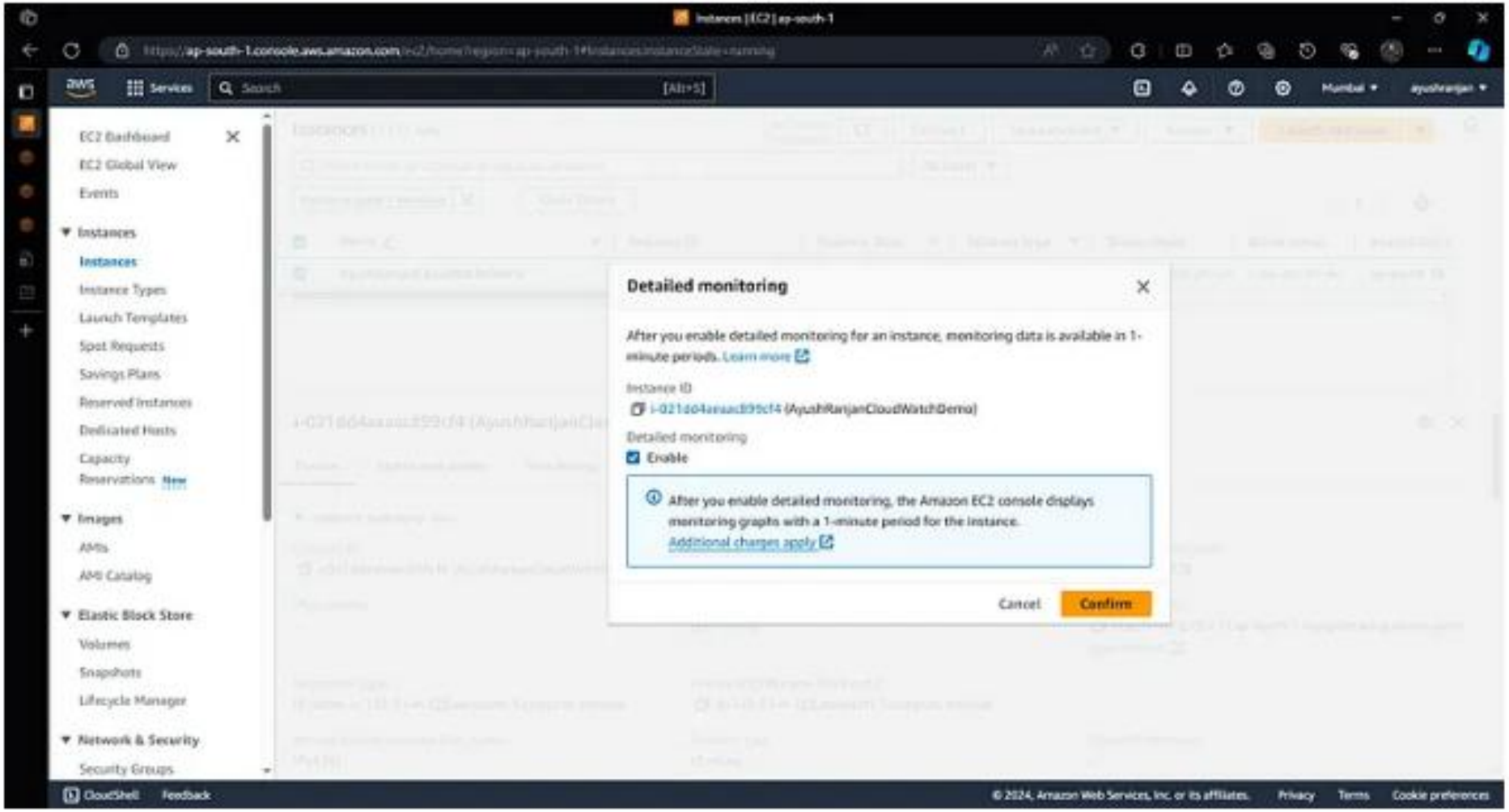
Detailed monitoring collects metrics at 1-minute intervals instead of the default 5-minute interval. While this is generally not needed for basic workloads, it can be useful for more precise monitoring.

It is not free but costs a minimal amount if you go beyond the free tier.

1. Enable Detailed Monitoring for EC2

- Navigate back to the EC2 Dashboard.
- Select your running instance.
- Under **Actions** in the top-right corner, choose **Monitor and troubleshoot** and click **Enable Detailed Monitoring**.





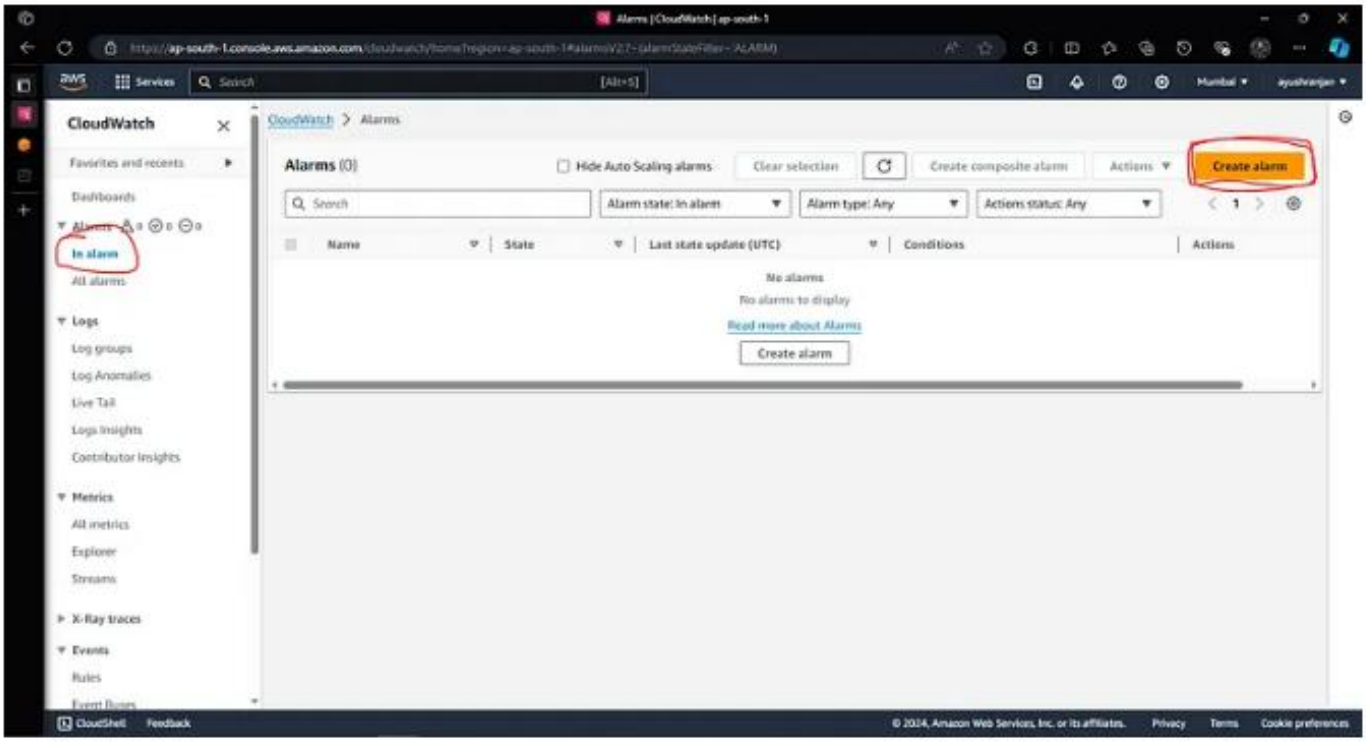
Important: Detailed monitoring is not free but is charged at around $0.30 per instance per month if you use it beyond free-tier limits. For this guide, it's optional.

Step 4: Set Up CloudWatch Alarms for EC2

Now, we will create a CloudWatch Alarm to notify you when the EC2 instance's CPU utilization exceeds a specific threshold (e.g., 70%). You can stay within the AWS Free Tier as alarms are included up to 10 alarms per month.

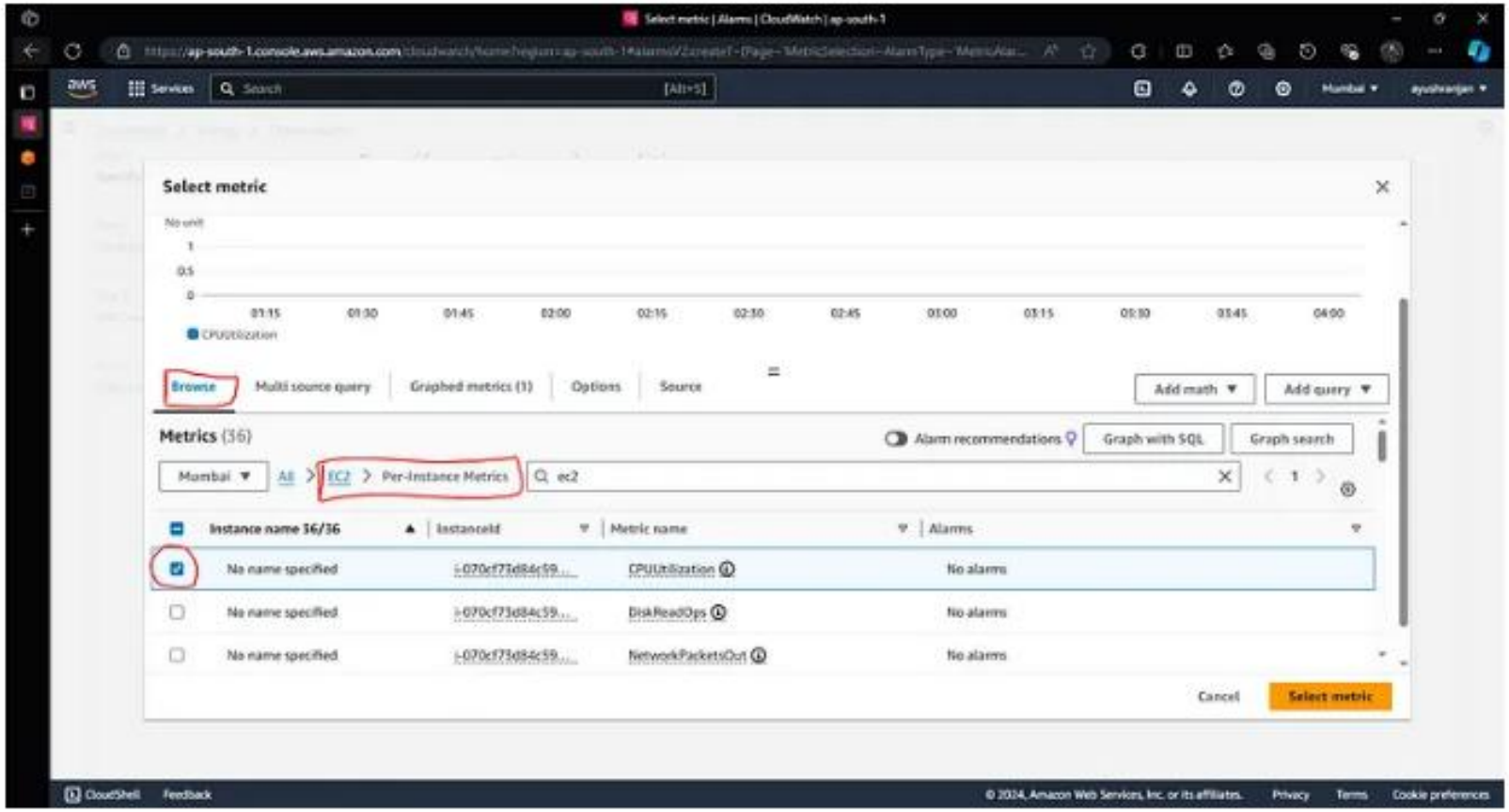
1. Navigate to Alarms in CloudWatch

- In the CloudWatch console, select **Alarms** from the left-hand menu.
- Click on **Create Alarm**.



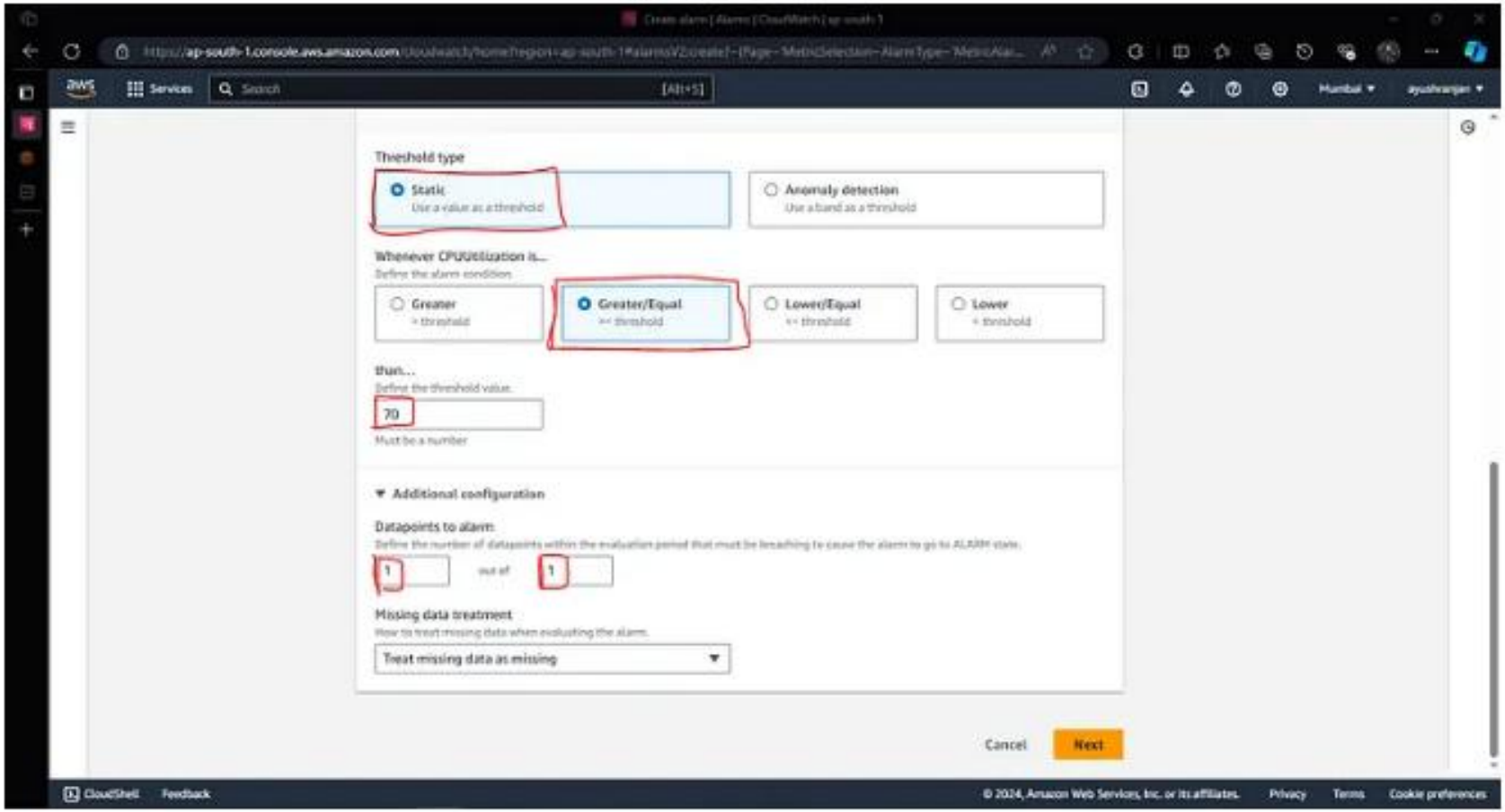
2. Select the Metric for CPU Utilization

- Click **Select metric.**
- In the **Browse** tab, choose **EC2 > Per-Instance Metrics.**
  Locate your EC2 instance and select **CPUUtilization** as the metric to monitor.
- Click **Select metric.**



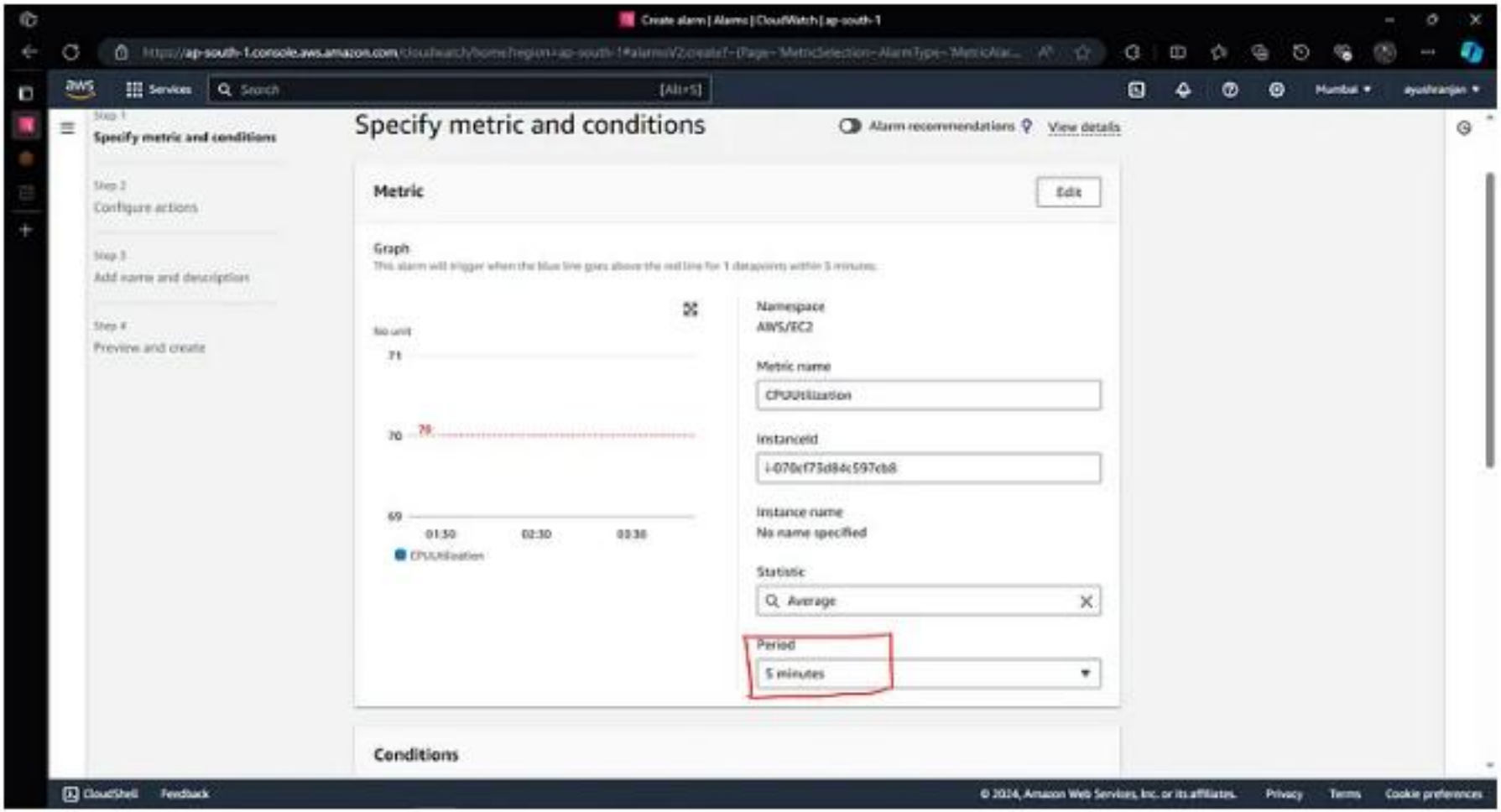
3. Set a Threshold

- In the conditions section, set the alarm threshold to notify you when CPU usage is higher than **70%**.
- For **Whenever CPU utilization is**, choose **Greater/Equal to** and set the threshold value to **70**.

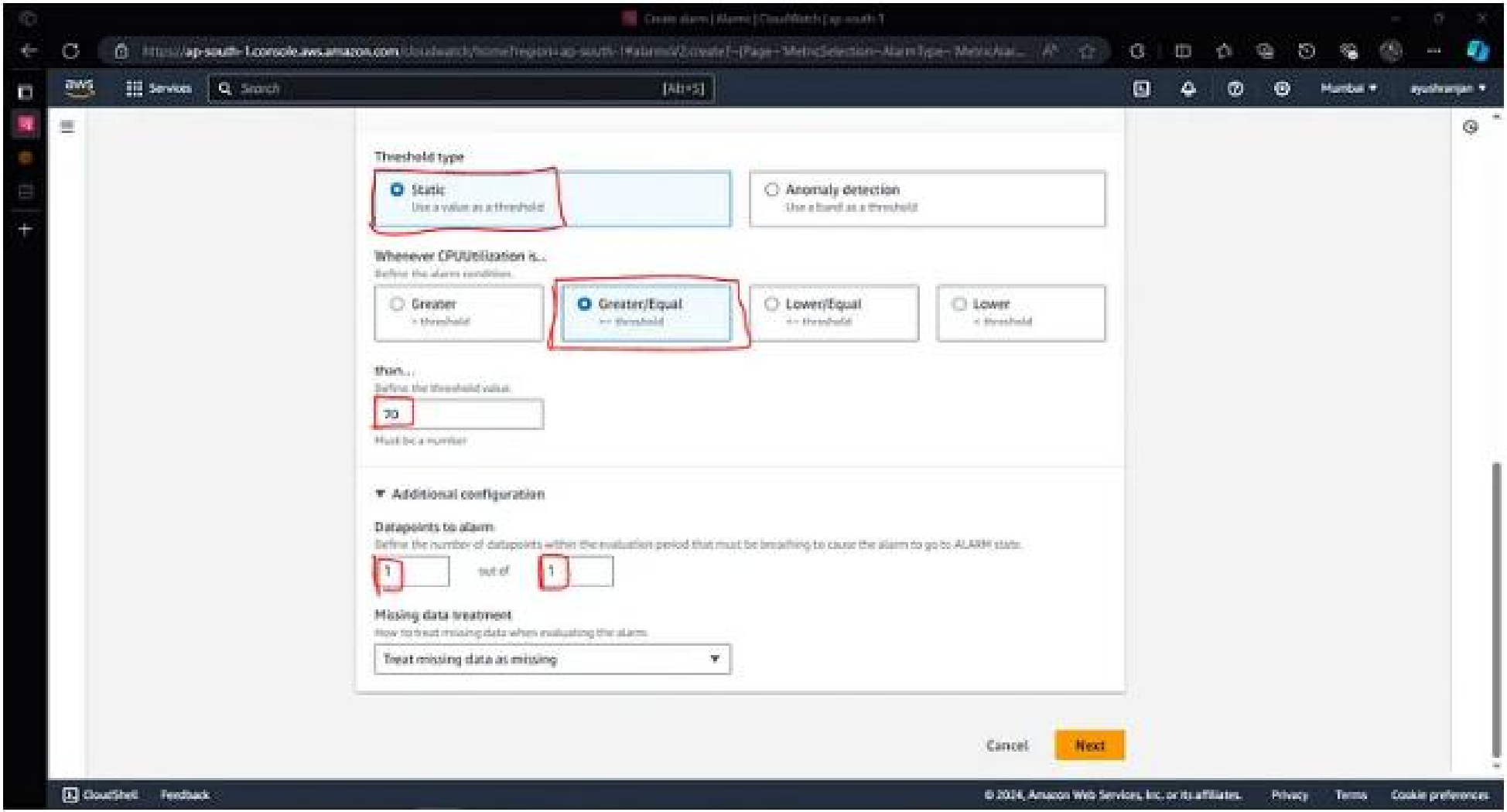


4. Set Period and Datapoints

- Set the evaluation period to **5 minutes** (free tier eligible).

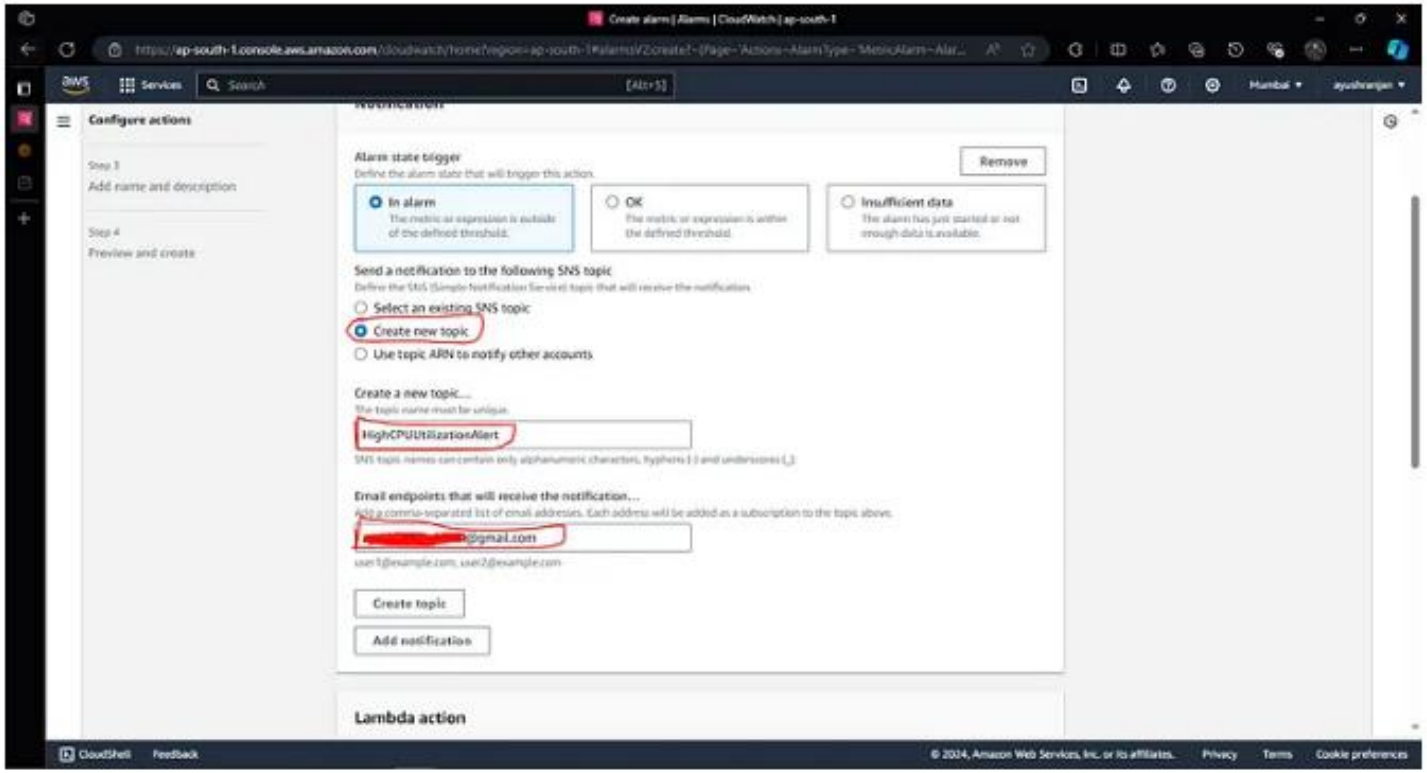


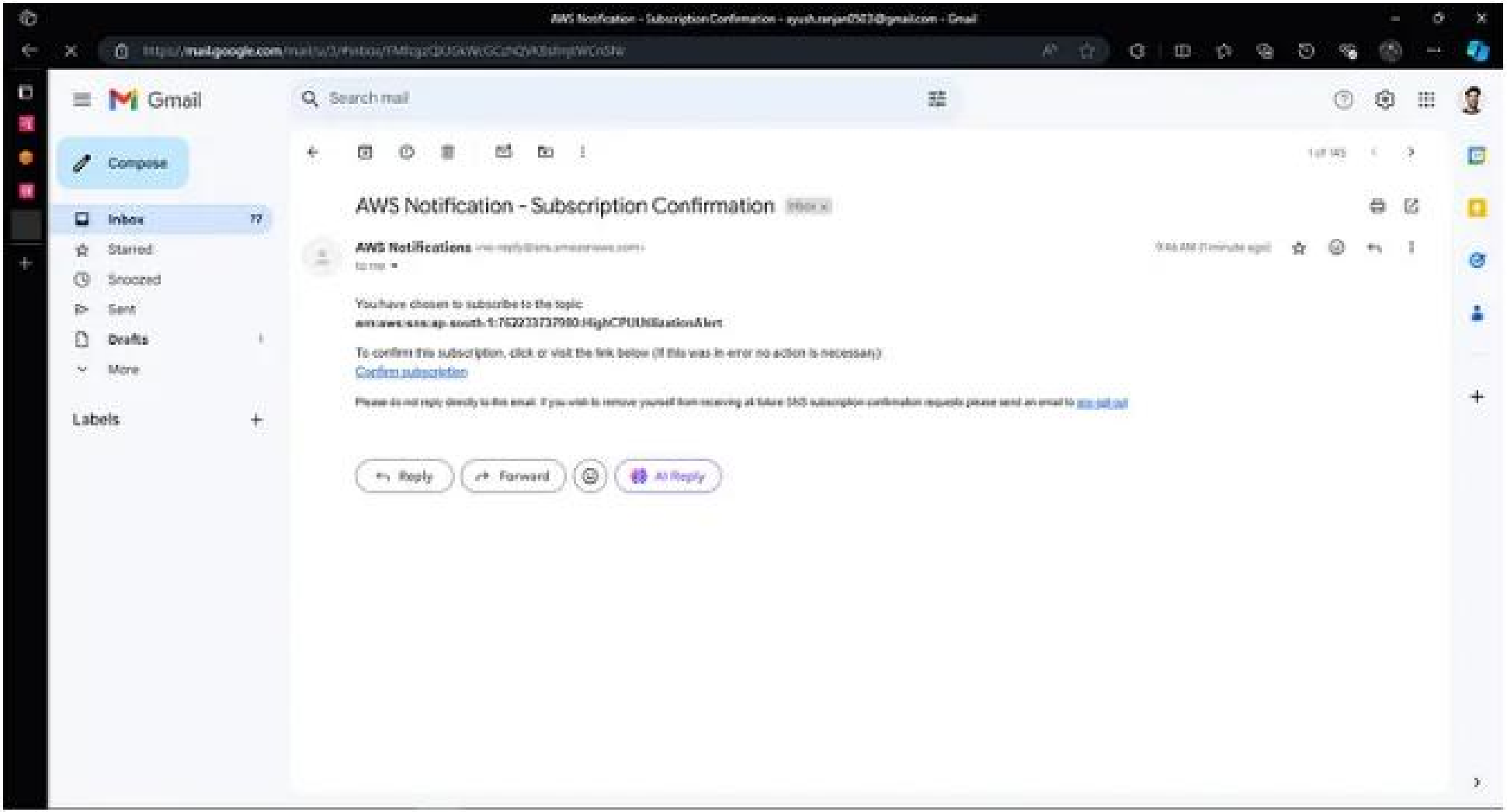
- Set the number of datapoints to **1 out of 1**.



5. Configure Actions

- Choose to trigger an action when the alarm state is reached. Select **Create new topic** to send an email notification.
- Enter a name for the topic (e.g., `HighCPUUtilizationAlert`) and input your email address.
- Confirm the SNS topic by clicking on the verification link sent to your email.

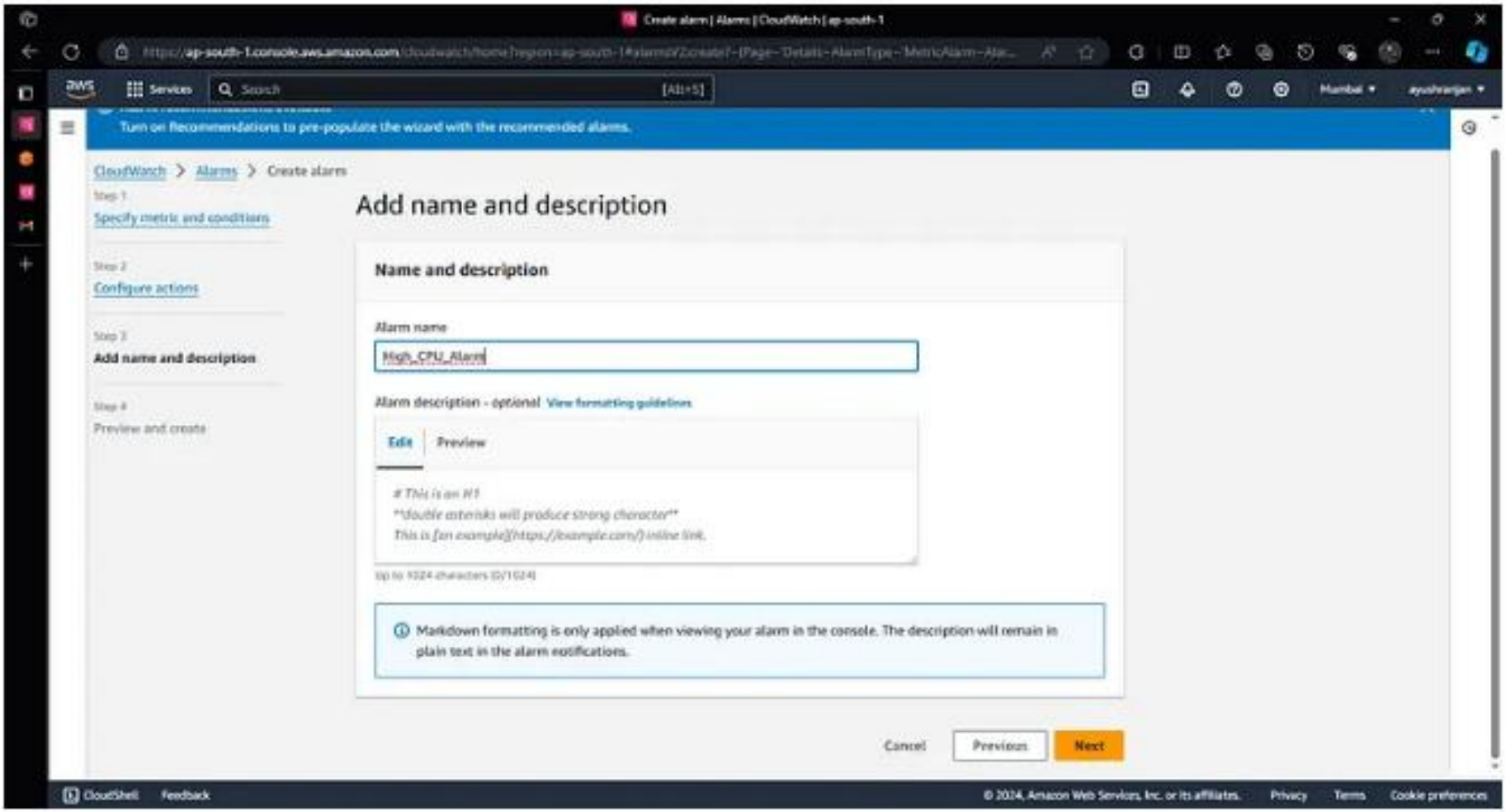




- Don't forget to subscribe the SNS by the link in the email.

6. Name and Create the Alarm

- Name your alarm (e.g., `High_CPU_Alarm`) for easy identification.
- Review the settings and click **Create Alarm**.



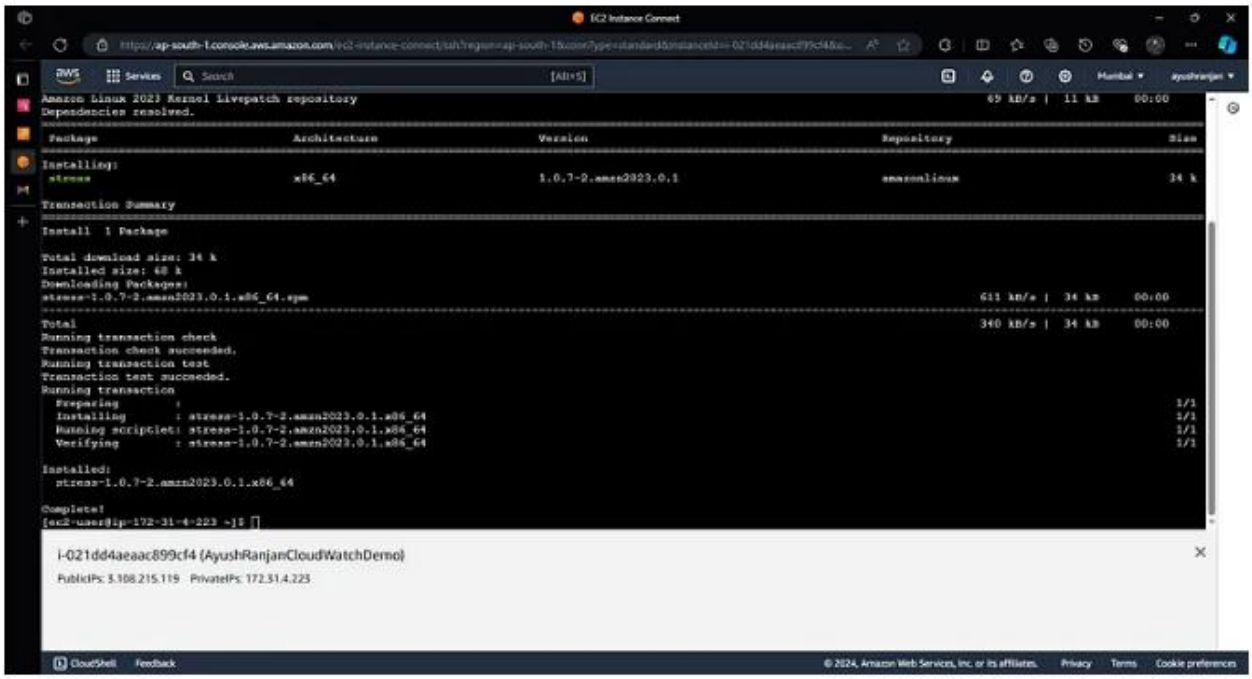
Step 5: Test the Alarm

- To ensure everything is set up correctly, you can test the alarm by artificially increasing CPU usage on your instance.

1. Install Stress Tool

- On Amazon Linux 2023, install the `stress` tool, which will generate CPU load:

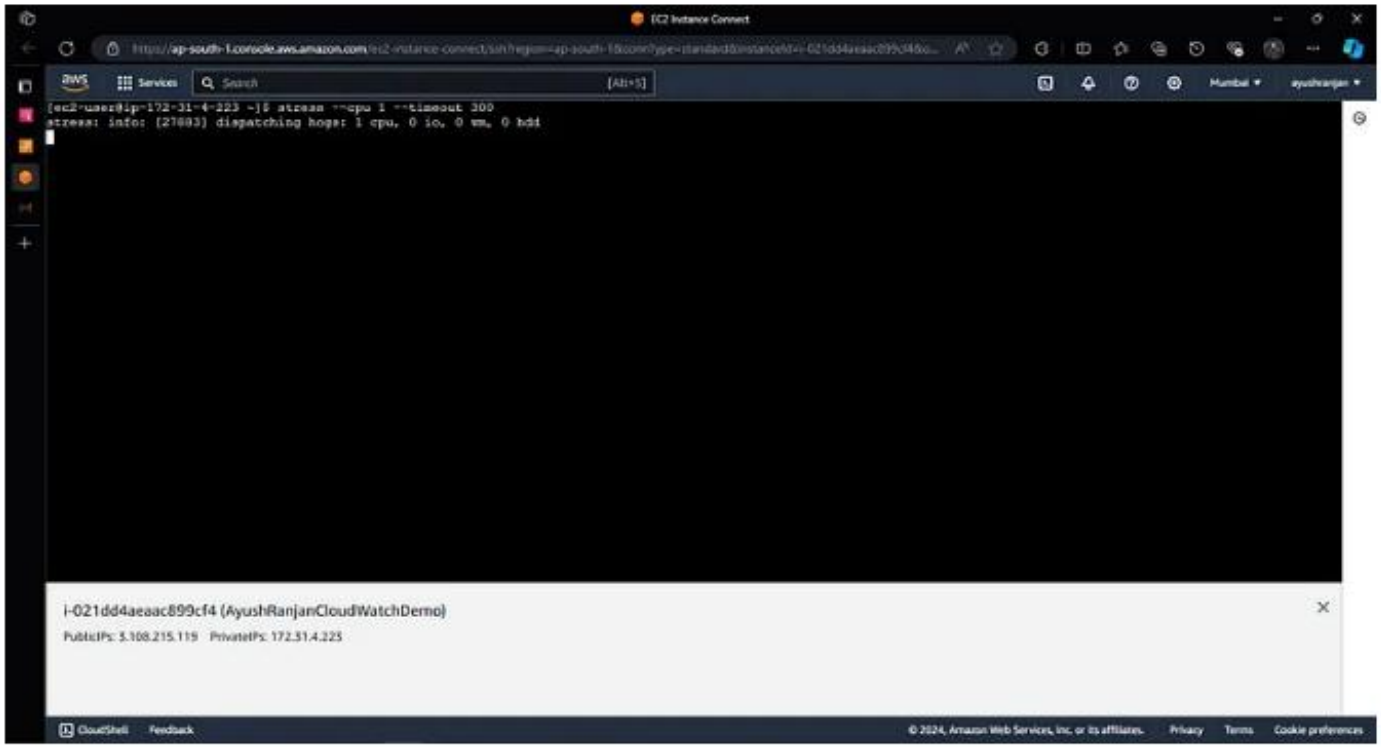
```
sudo yum install stress -y
```



2. Generate CPU Load

- Run the following command to stress the CPU for 5 minutes:

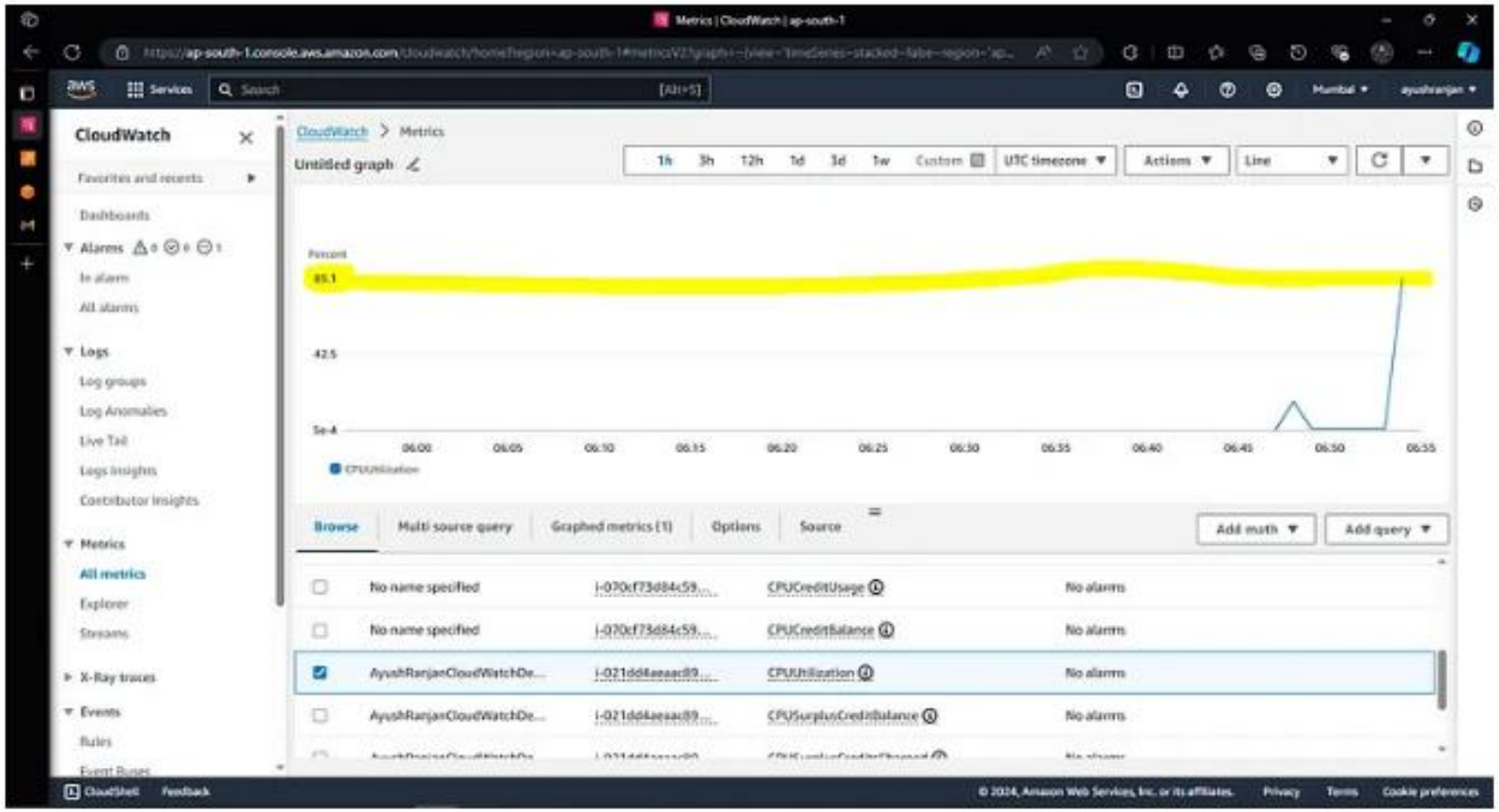
```
stress --cpu 1 --timeout 300
```



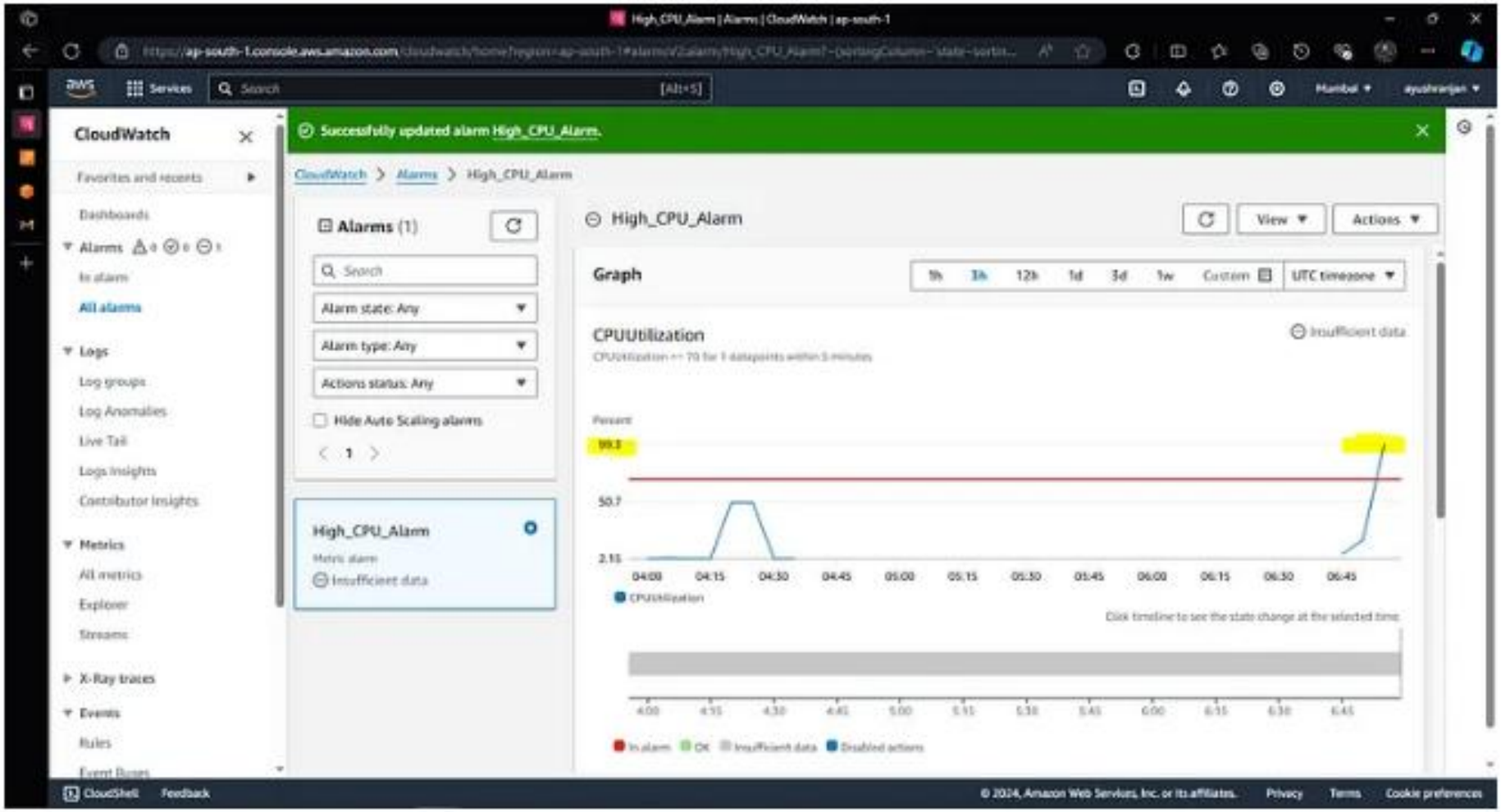
This command will use **1 CPU core for 5 minutes**, increasing the CPU utilization enough to trigger the CloudWatch alarm.

4. Monitor CloudWatch

- Head over to the CloudWatch console and observe the CPU utilization metrics for your EC2 instance.
- Within a few minutes, you should see the CPU utilization rise and cross the **70% threshold**, triggering the alarm.

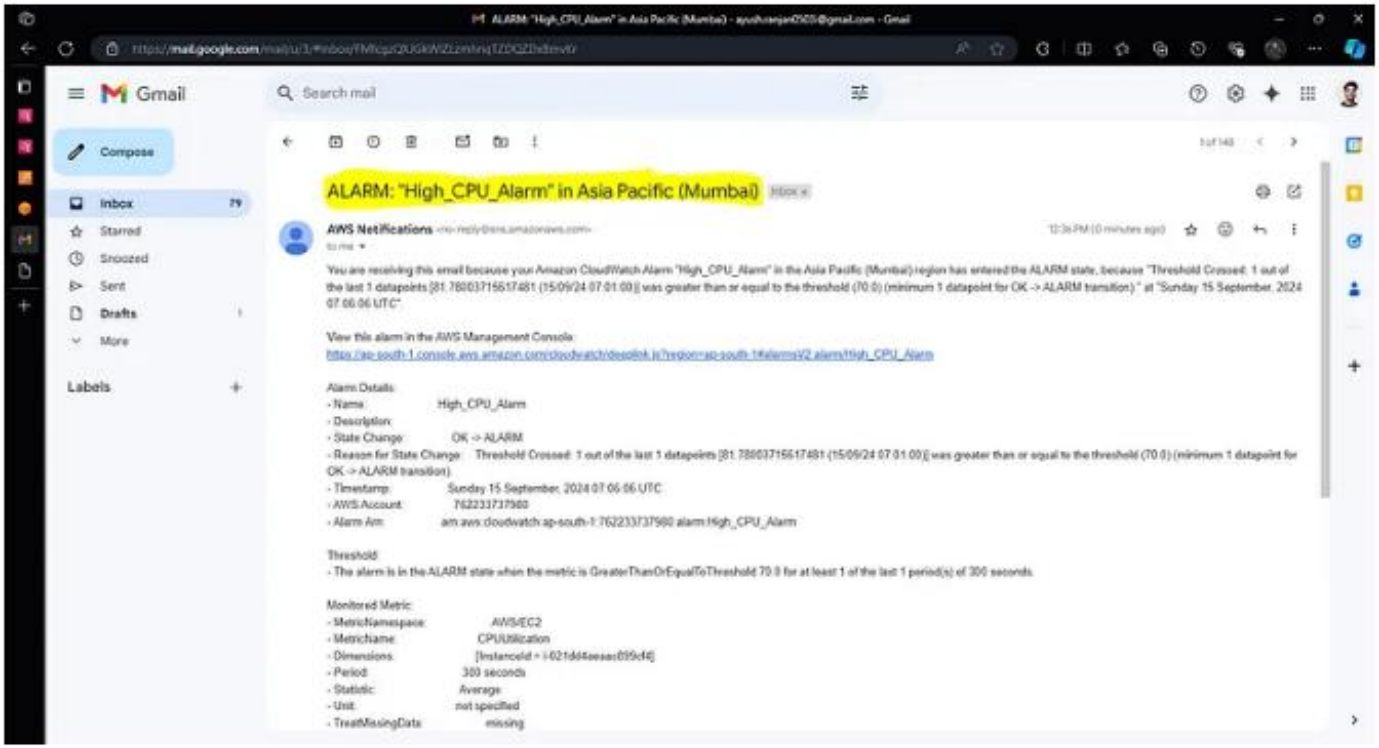


- See at the alarms section



Step 6: Respond to the Alarm

Once the alarm is triggered, you will receive an email notification.



If the load persists, you can:

- Scale up the instance (increase the instance size to handle more traffic).
  - Investigate the process causing high CPU utilization.
  - Implement auto-scaling based on CloudWatch metrics to handle varying traffic loads.