Project 1

AWS CloudWatch Dashboards for Comprehensive Monitoring

Batch

MCA 7 Feb 2025 &
DevOps 2 Jun 2025

Name

Suraj Molke

Name of the Mentor

Ravindra Bagle Sir
Swati Zampal Ma'am

Fortune Cloud Technologies

2nd Floor, Shirodkar House, Opposite To Amit Cafe, Congress House Road, Near Pune Municipal Corporation, Shivajinagar, Pune - 411005.

smolke9@gmail.com Suraj Molke Smolke9

Project: 1 https://github.com/Smolke9/AWS-CloudWatch-Dashboards-for-Comprehensive-Monitoring-.git

Title: AWS CloudWatch Dashboards for Comprehensive Monitoring

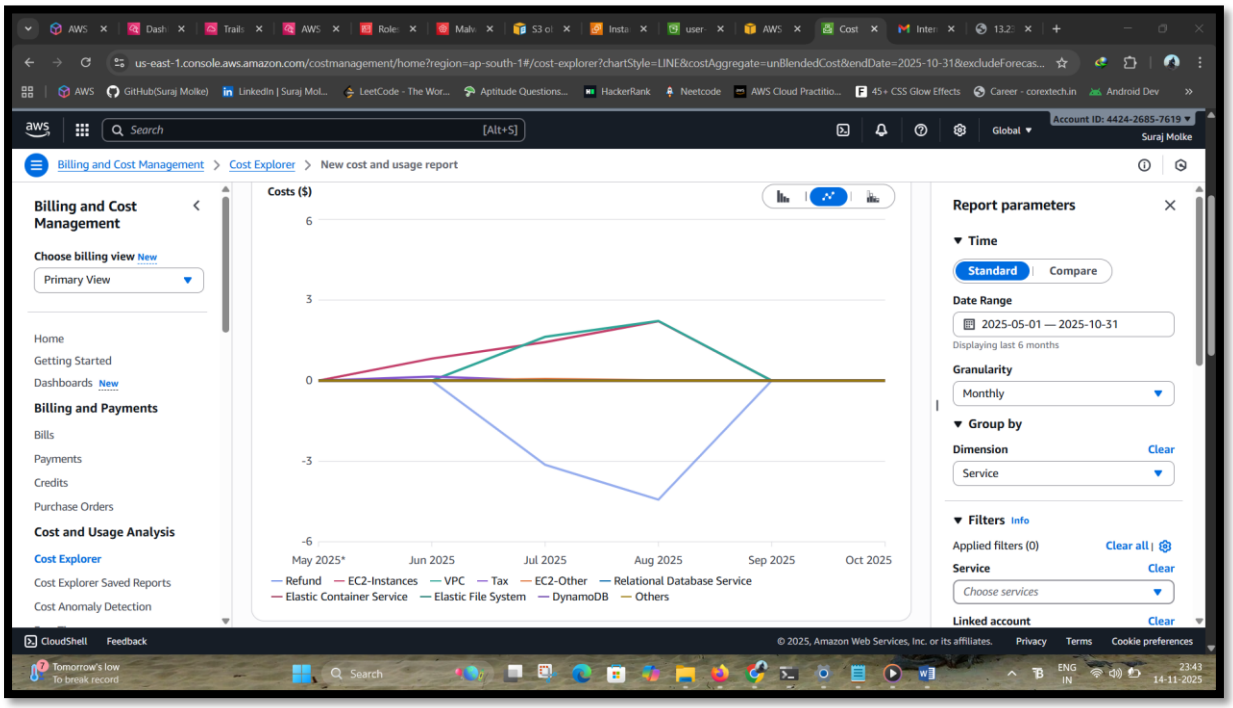
- ✓ Services used:
- ✓ Amazon CloudWatch (Dashboards, Metrics, Logs Insights)
- ✓ AWS Config (for compliance)
- ✓ AWS GuardDuty (for security threat detection)
- ✓ AWS CloudTrail (for API monitoring)
- ✓ IAM (for access control)
- ✓ EC2 (host to push logs)
- ✓ S3 (used for AWS Config)
- ✓ Application Load Balancer (for network monitoring)

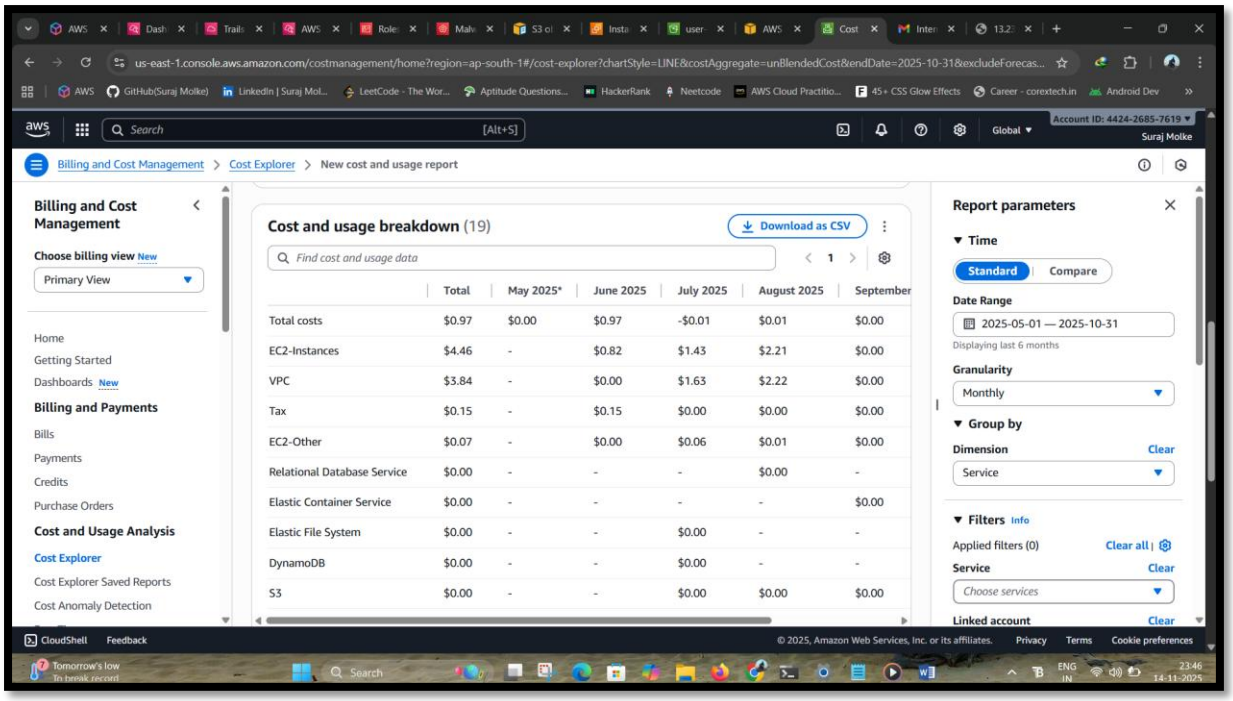
Objective:

Build and configure a CloudWatch Dashboard that provides real-time visibility into key operational and financial metrics across four focus areas:

Step 1. Enable Billing Tools

1. Go to **Billing → Cost Management**
2. Enable **Cost Explorer**

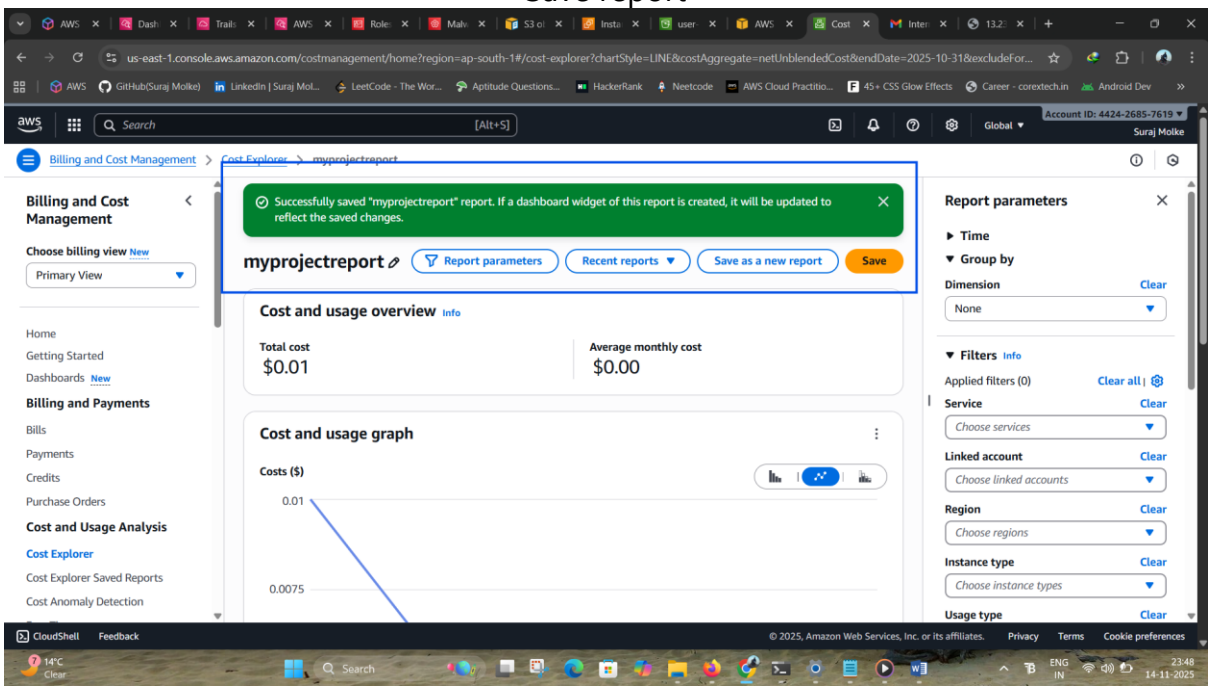




Step 2. Create Reports in AWS Cost Explorer

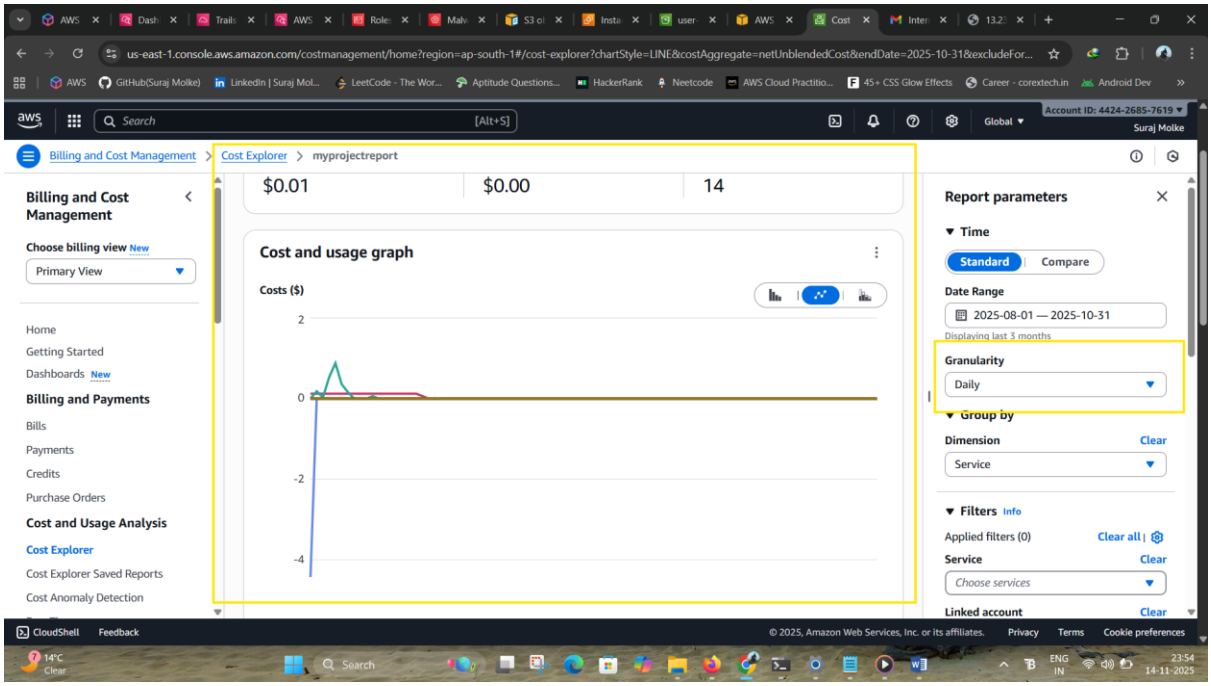
2.1 Total Cost Over Time

- Open **Cost Explorer**
- Granularity: **Daily**
- Metric: **Unblended Cost**
- Group by: **None**
- Save report



2.2. Daily Estimated Charges by Service

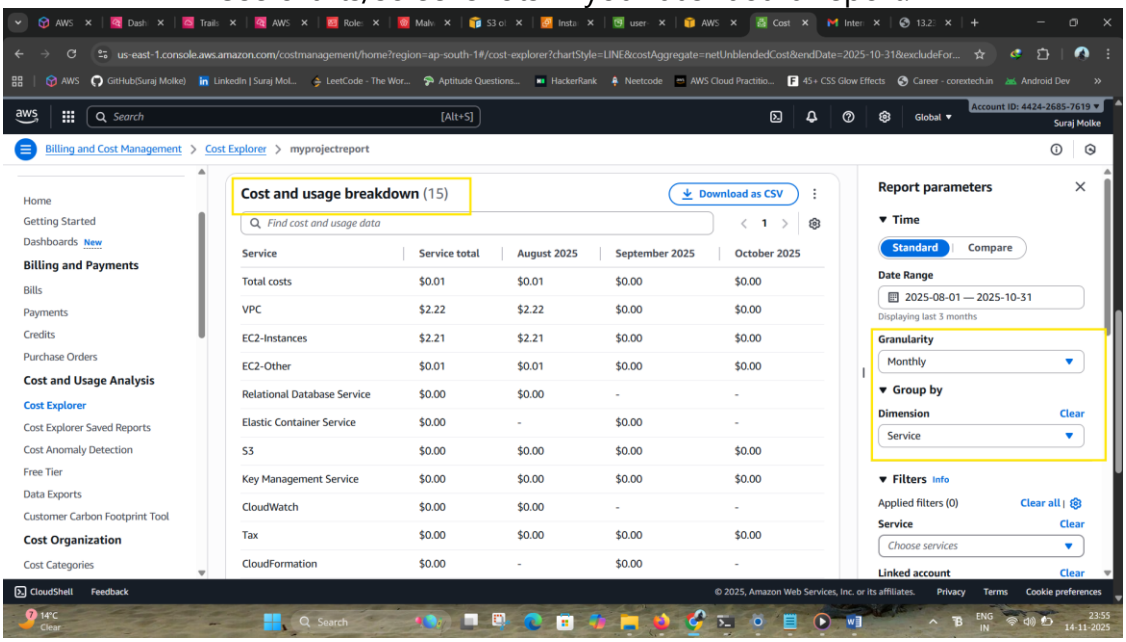
- Cost Explorer → Create Report
- Granularity: **Daily**
- Group by: **Service (EC2, S3, RDS, etc.)**
- Save report



2.3. Monthly Cost Breakdown

- Granularity: **Monthly**
- Group by: **Service**
- Save report

Use charts/screenshots in your dashboard report.



Step 3. Application & System Logs

Steps:

3.1. Send Logs to CloudWatch Logs

From EC2

Install and configure CloudWatch Agent

```
sudo yum install amazon-cloudwatch-agent -y
```

```
sudo nano /opt/aws/amazon-cloudwatch-agent/bin/config.json
```

Add log paths:

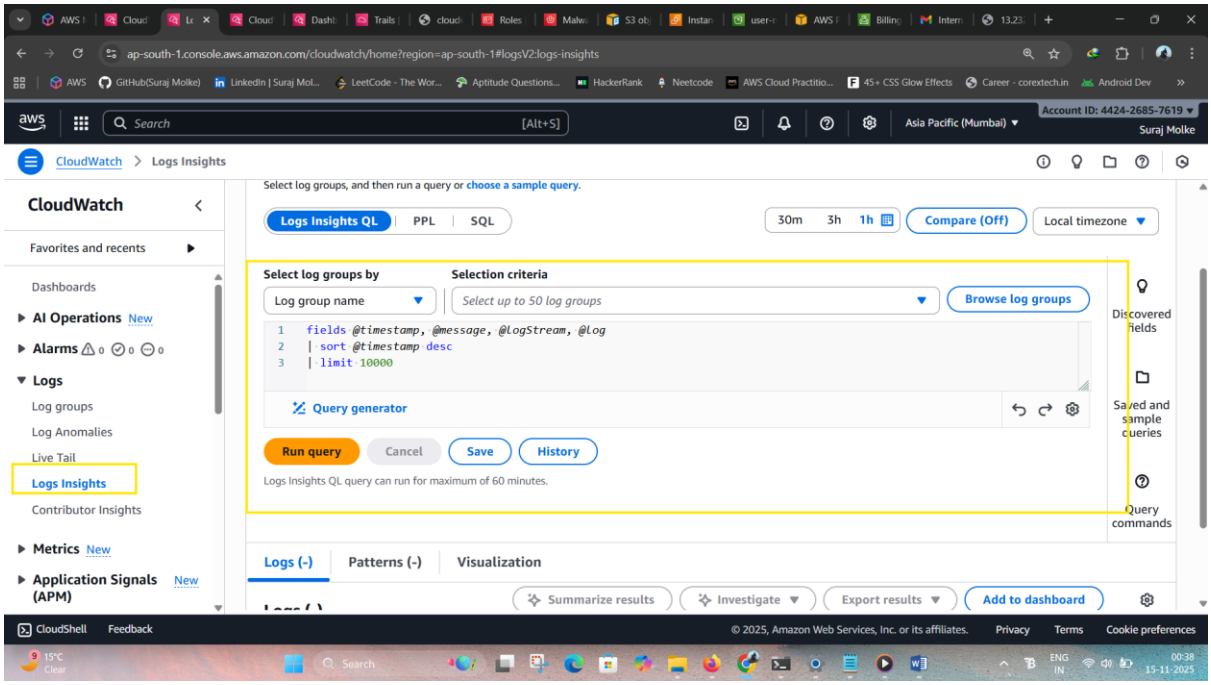
```
{
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          { "file_path": "/var/log/messages", "log_group_name": "system-logs" },
          { "file_path": "/var/log/nginx/error.log", "log_group_name": "nginx-logs" }
        ]
      }
    }
  }
}
```

Start agent:

```
sudo systemctl start amazon-cloudwatch-agent
```

Step 4. Create CloudWatch Log Insights Queries

Go to:
CloudWatch → Logs → Log Insights → Select a Log Group

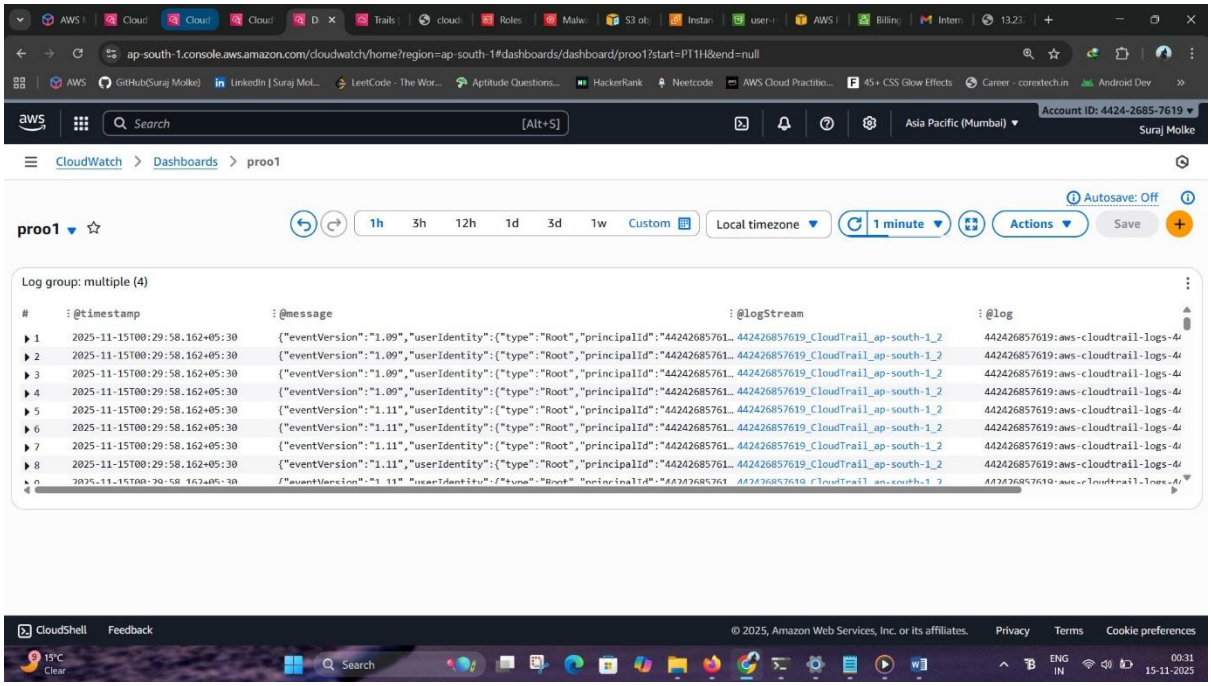


Step 5. Add Log Insights to Dashboard

Go to:
CloudWatch → Dashboards → Create Dashboard → Add Widget

Add widgets:

- Log Insights → Query results (Table)
- Line graph for response times
- Bar chart for error counts
- Number widget for total errors in last 1 hour



Step 6 System Logs for CloudWatch Logs Dashboard

6.1. /var/log/messages

- This is the **main system log file** on Linux servers (Amazon Linux, CentOS, RHEL).
- Contains:
  - Kernel events
  - Service start/stop logs
  - System warnings
  - General OS activity
- Useful for identifying:
  - Server crashes
  - Restart issues
  - Hardware/network errors
  - Authentication problems

Path: /var/log/messages

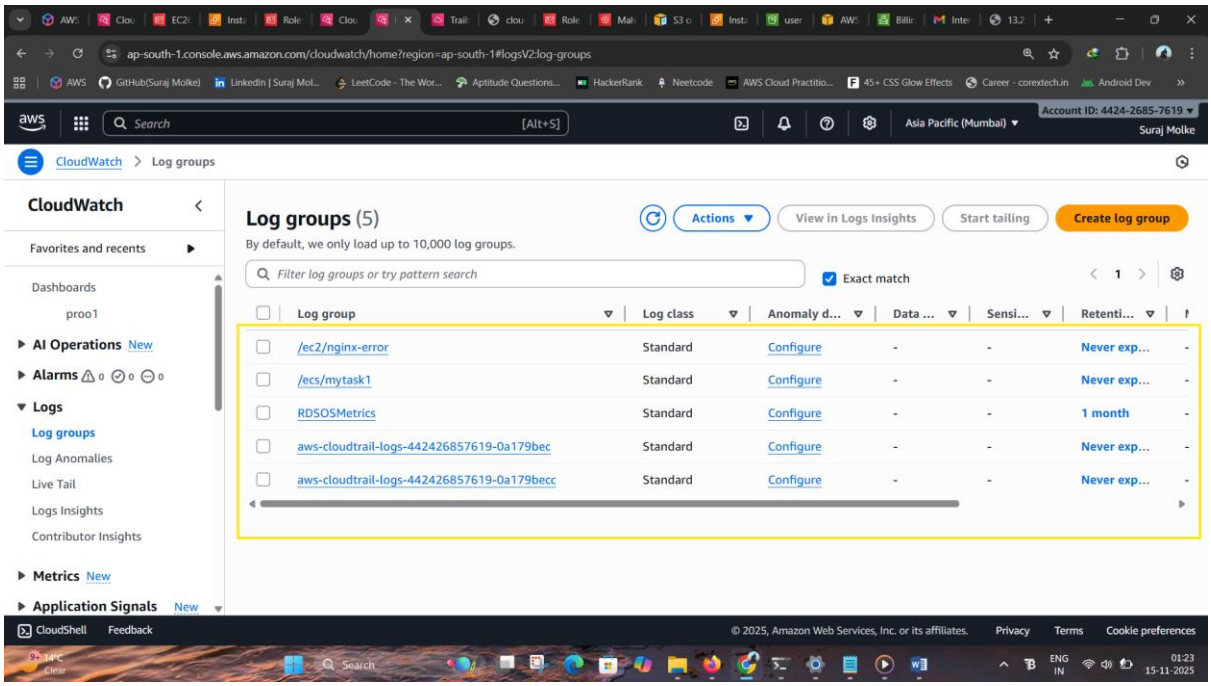
2. /var/log/nginx/error.log

- This file stores all **Nginx-related errors**.
- Contains:
  - HTTP errors
  - Backend connection failures
- **Install CloudWatch Agent**
- sudo yum install amazon-cloudwatch-agent -y
- **Edit agent config**

```
{

  "logs": {

    "logs_collected": {

      "files": {

        "collect_list": [

          {

            "file_path": "/var/log/messages",

            "log_group_name": "/ec2/system-messages",

            "log_stream_name": "{instance_id}-messages",

            "timestamp_format": "%b %d %H:%M:%S"
```

```
      },
      {
        "file_path": "/var/log/nginx/error.log",
        "log_group_name": "/ec2/nginx-error",
        "log_stream_name": "{instance_id}-nginx-error",
        "timestamp_format": "%Y/%m/%d %H:%M:%S"
      }
    ]
  }
 }
}
```

- **Start the agent**
- sudo systemctl start amazon-cloudwatch-agent
- Logs appear in:
  - **/aws/system-logs**
  - **/aws/nginx-logs**
  - Permission problems



Network Performan

Step 7. VPC Network Monitoring

Collect and visualize:

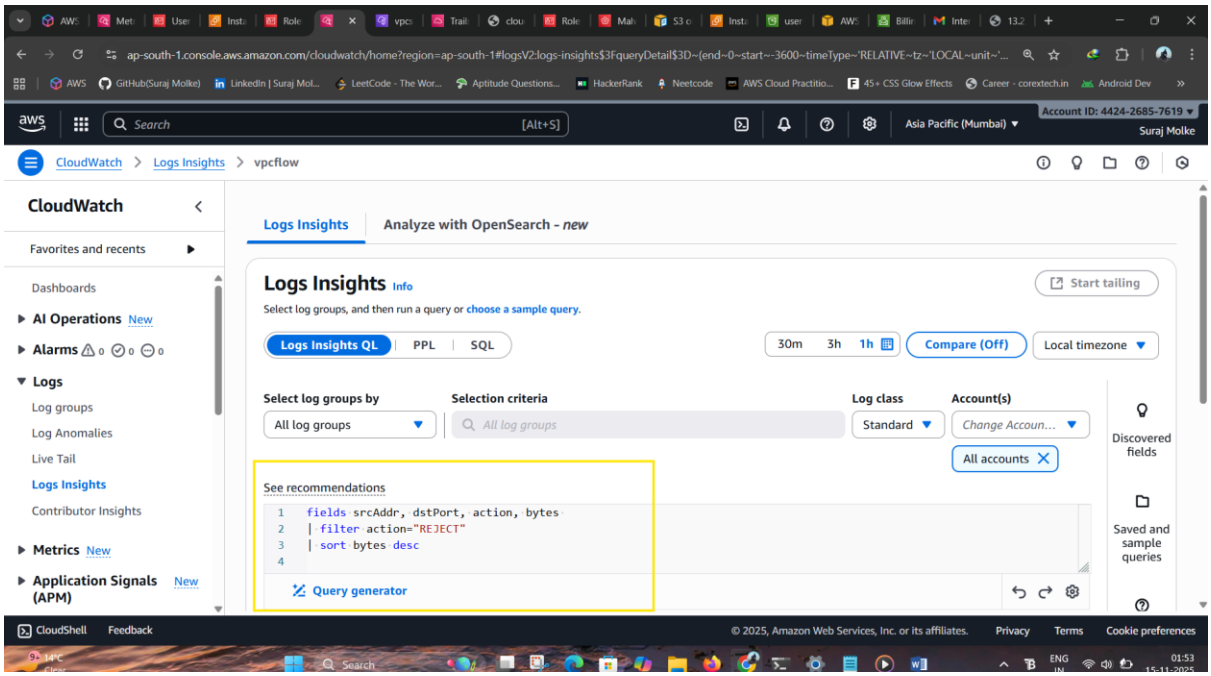
- **VPC Flow Logs (Accepted/Rejected traffic)**
- **Bytes in/out**
- **Packet count**
- **Rejected connections by security groups or NACLs**

Steps :

1. Enable **VPC Flow Logs** (VPC → Flow Logs → Create).
2. Send logs to **CloudWatch Logs**.
3. Use **Log Insights** to monitor:
    - Top IPs hitting instance
    - Rejected traffic
    - Port scans

Useful Log Insights Query:

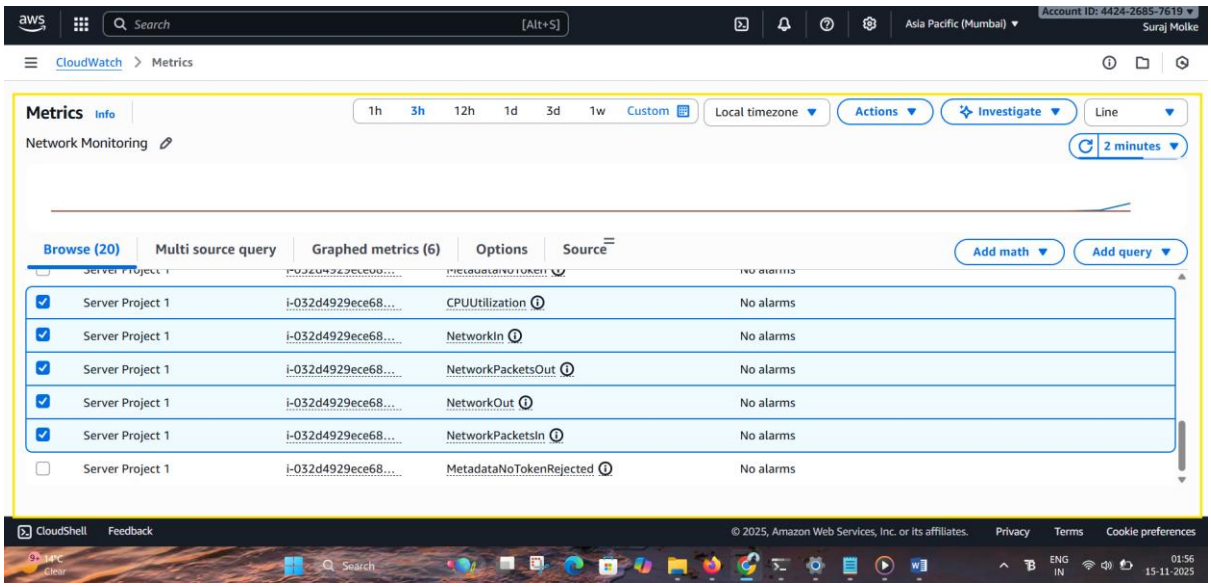
```
fields srcAddr, dstPort, action, bytes
| filter action="REJECT"
| sort bytes desc
```



Step 7.1. EC2 Network Metrics

From CloudWatch → Metrics → EC2 you can monitor:

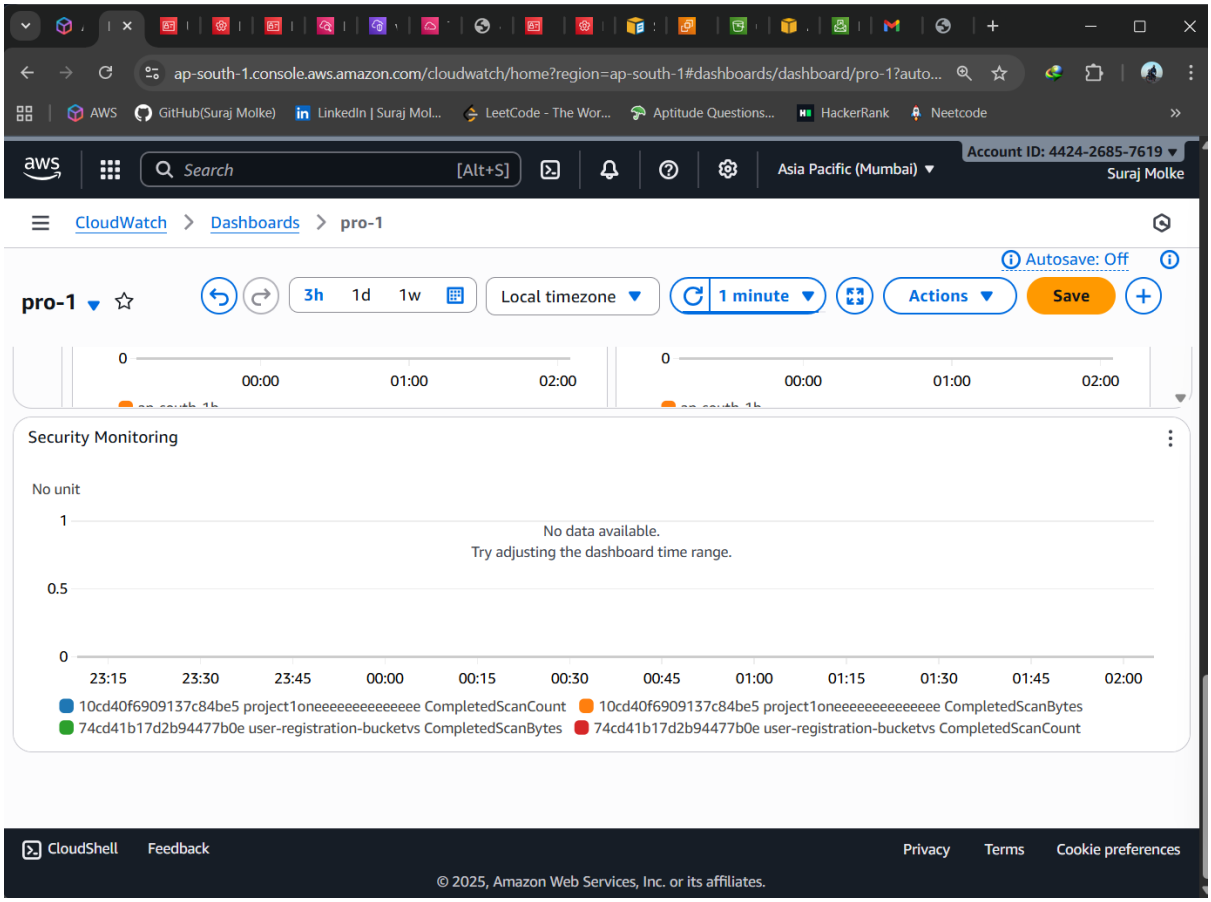
- **NetworkIn (bytes received)**
- **NetworkOut (bytes sent)**
- **NetworkPacketsIn/Out**
- **StatusCheckFailed**

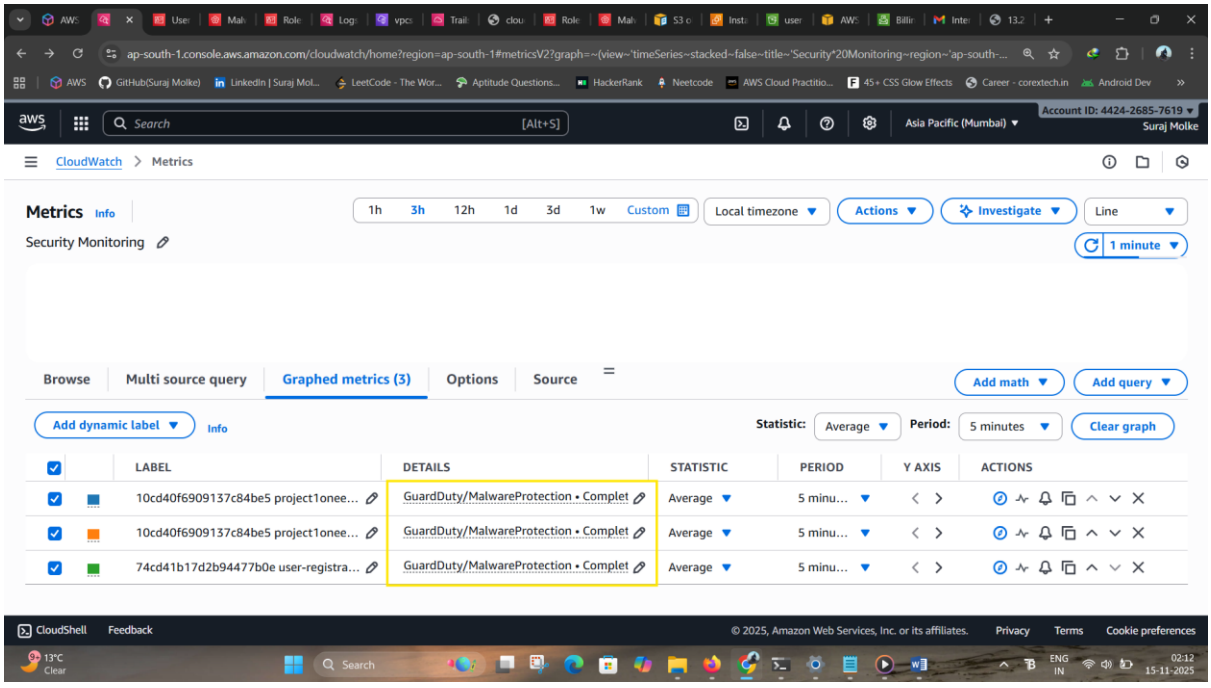


Step 8 Security Monitoring Dashboard — Short Steps

8.1. GuardDuty Monitoring

- Enable **GuardDuty** from the AWS Console.
- Go to **CloudWatch → Metrics → GuardDuty**.
- Monitor:
  - **Findings** (Severity: Low, Medium, High)
  - **Threat detections over time**
- Add widgets:
  - Line chart → *Total Findings Over Time*
  - Bar chart → *High Severity Findings*

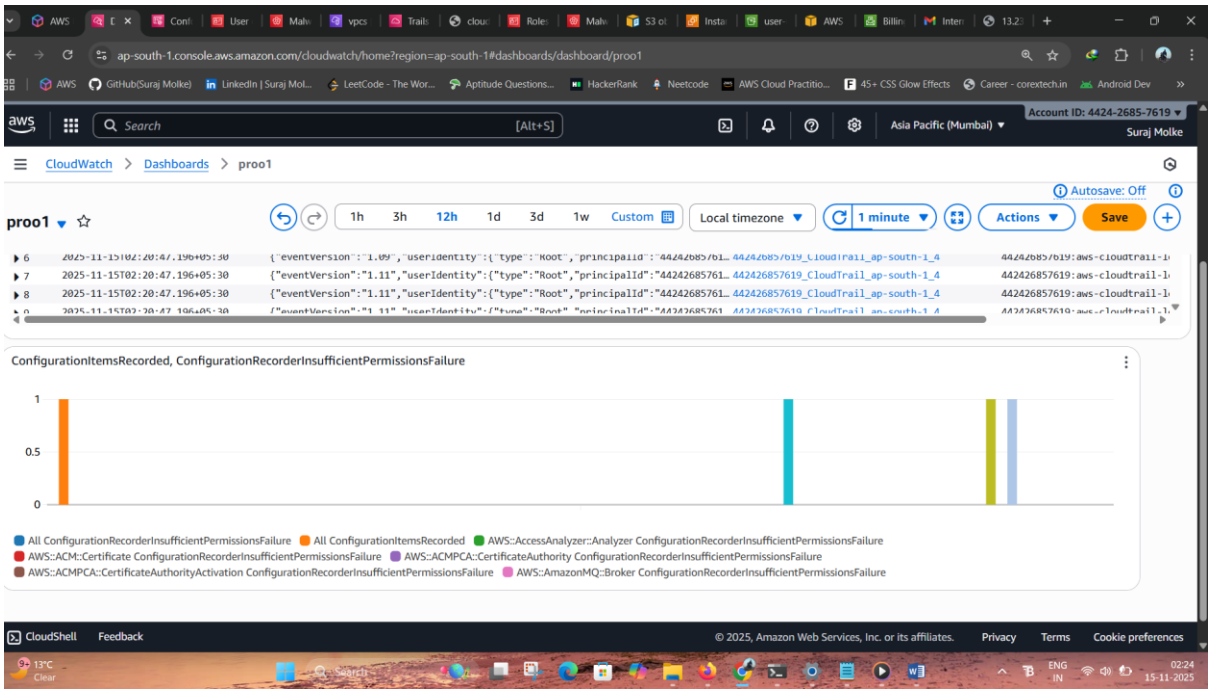




8.2. AWS Config (Compliance Monitoring)

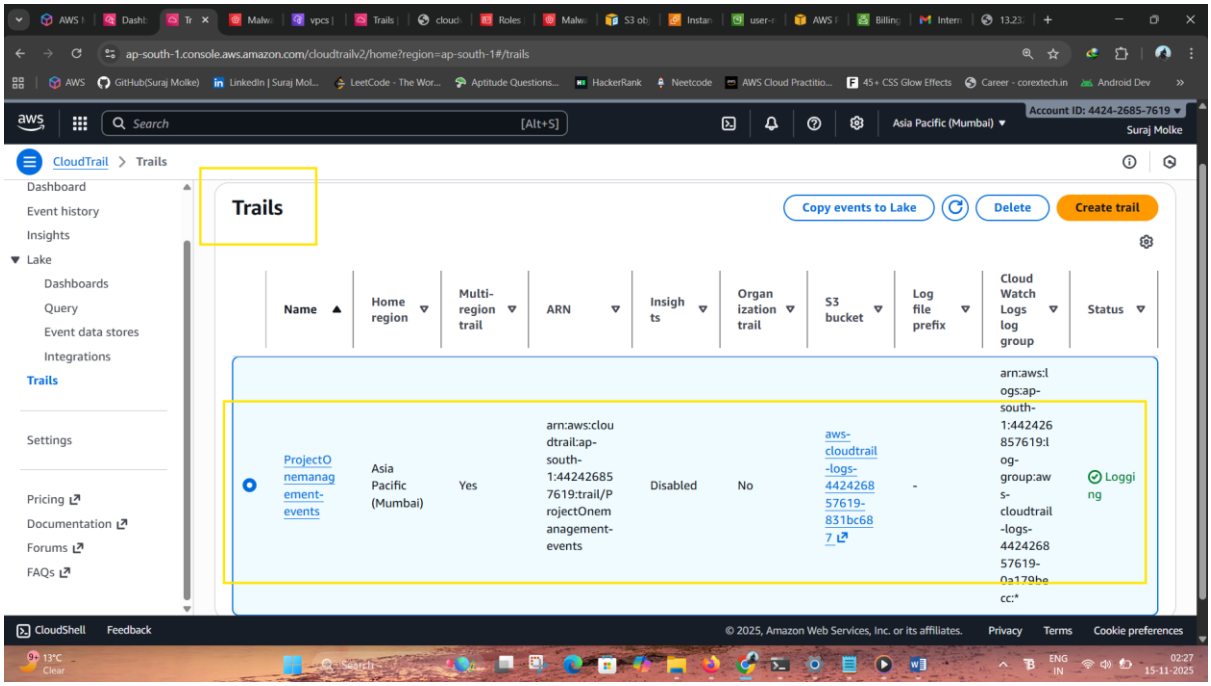
- Enable AWS Config → Choose **recording** & **S3 log bucket**.
- Go to **CloudWatch → Metrics → Config**.
- Monitor:

    - **NonCompliantResourceCount**
    - **ComplianceStatus**
- Dashboard widgets:
    - Number widget → *Total Non-Compliant Resources*
    - Gauge widget → *Compliance %*



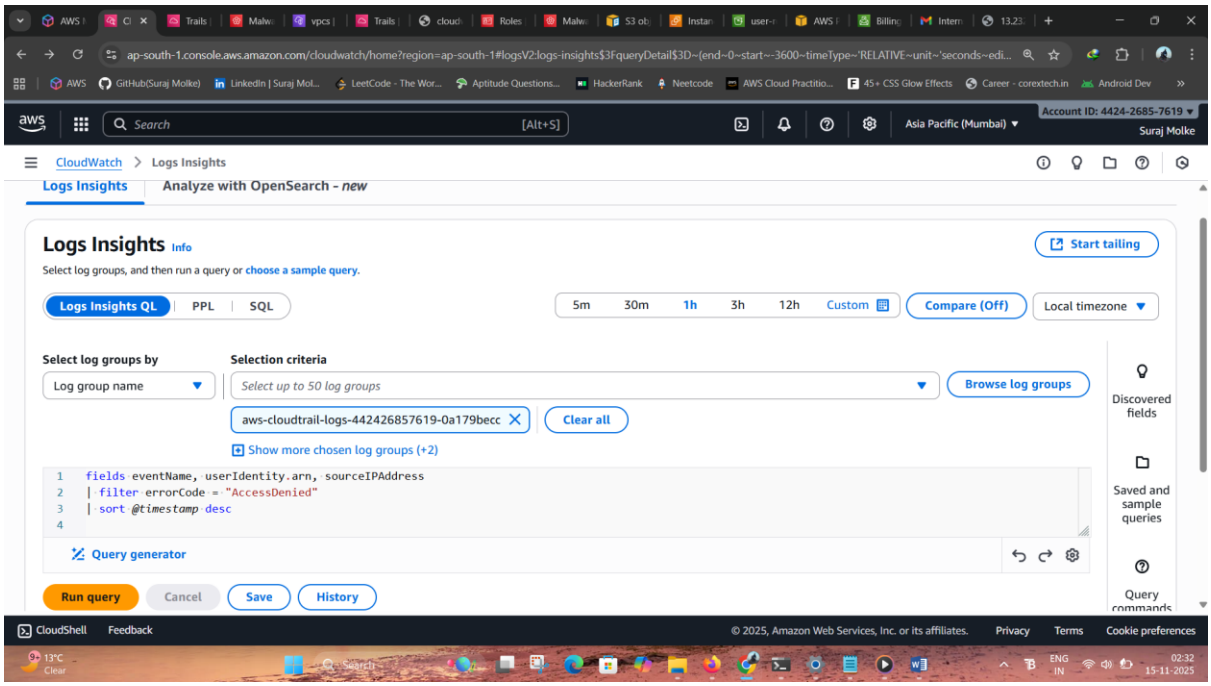
8.3. CloudTrail (API Activity & Security Events)

- Ensure CloudTrail is enabled (1 trail per region).
- Use **CloudWatch Log Insights** to analyze activity:
    - Unusual logins
    - Root account usage
    - Deletion/modification events



Log Insights Query:

```
fields eventName, userIdentity.arn, sourceIPAddress
| filter errorCode = "AccessDenied"
| sort @timestamp desc
```



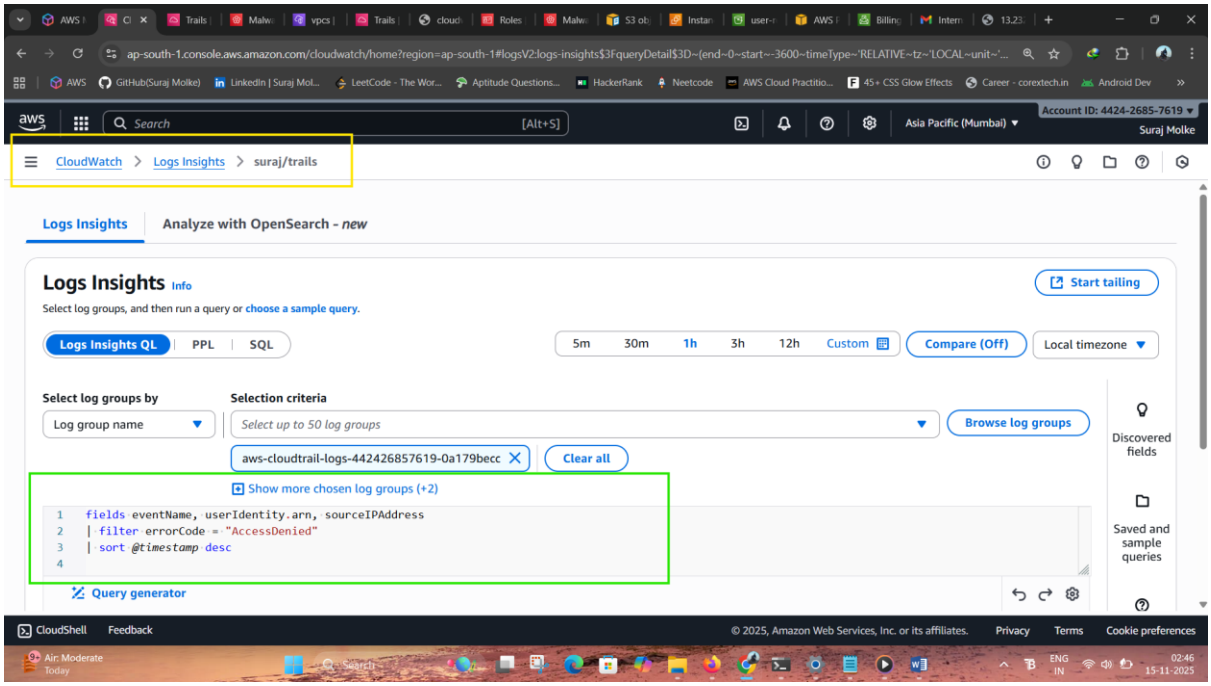
8.4. IAM Security Events

Monitor via CloudWatch metrics and CloudTrail logs:

- Access key usage
- Unauthorized attempts
- Password policy compliance

Useful query for Unauthorized attempts:

```
fields userIdentity.userName, eventName, errorCode, sourceIPAddress
| filter errorCode like /AccessDenied/
| sort @timestamp desc
```



Dashboard widgets:

- Number widget → *Unauthorized API Calls Today*
- Status icon → *Access Key Rotation Age*

Conclusion

This project successfully demonstrates a complete, cost-efficient monitoring setup using AWS CloudWatch without relying on additional paid services like Lambda. By configuring four dedicated dashboards—Billing, Logs, Network Traffic, and Security—you

achieved full visibility across system performance and operational health. The EC2 instance, integrated with CloudWatch Agent and supported by appropriate IAM roles, enabled seamless log streaming and remote management through SSM. Realistic traffic generated using a load balancer with NGINX ensured practical network insights, while custom log groups and queries provided actionable details on failed logins, system events, and API activity. Overall, the solution delivers a fully functional, scalable, and low-cost monitoring architecture suitable for both learning and real-world cloud operations.