

Name : Nakul Lahode

Task - 02 : Set up monitoring for a cloud-based application using AWS CloudWatch, Google Cloud Monitoring, or Azure Monitor, with configured alerts and a dashboard showcasing metrics

Overview of AWS CloudWatch :

- 1. Monitoring Service**
 - CloudWatch monitors AWS resources and applications in real-time.
- 2. Metrics Collection**
 - Automatically collects metrics like **CPU usage, network traffic, and disk I/O** from services like EC2, RDS, and Lambda.
- 3. Custom Dashboards**
 - Allows creation of **visual dashboards** to monitor selected metrics.
- 4. Alarms & Notifications**
 - Set alarms on thresholds and get alerts via **SNS** (email/SMS) or trigger actions like **Lambda functions**.
- 5. Log Management**
 - Collects and stores logs from applications and services using **CloudWatch Logs**.
- 6. Event-Driven Actions**
 - Responds to system events automatically using **CloudWatch Events (EventBridge)**.
- 7. Custom Metrics Support**
 - Supports pushing custom metrics from applications using AWS SDK or CLI.

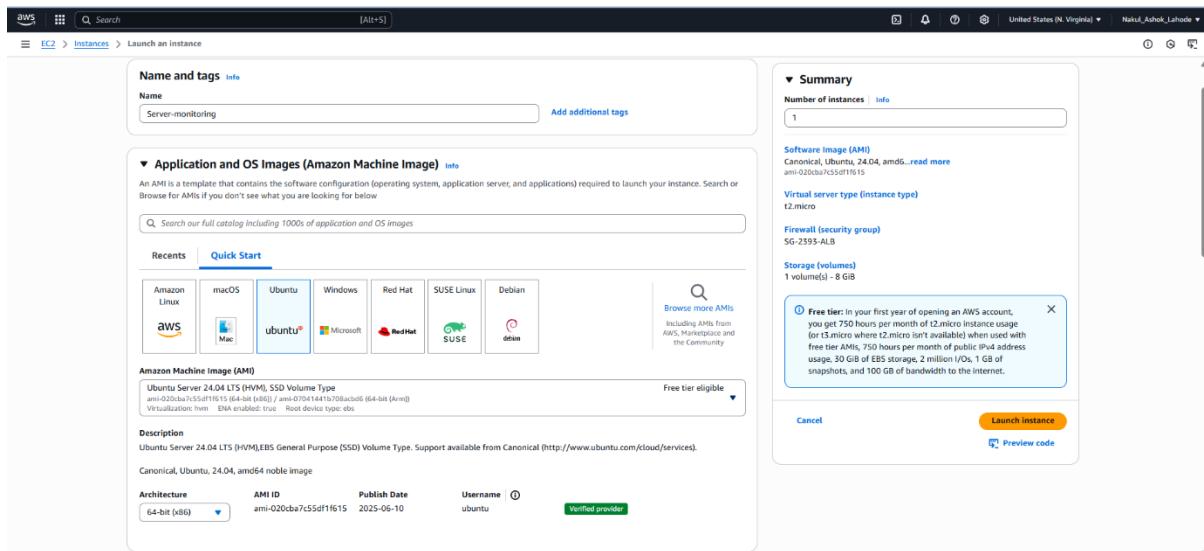
Step 1: Launch EC2 Instance & Install Apache

1. Log in to the AWS Management Console and navigate to **EC2**.
2. Click on "**Launch instance**" to create a new EC2 instance.

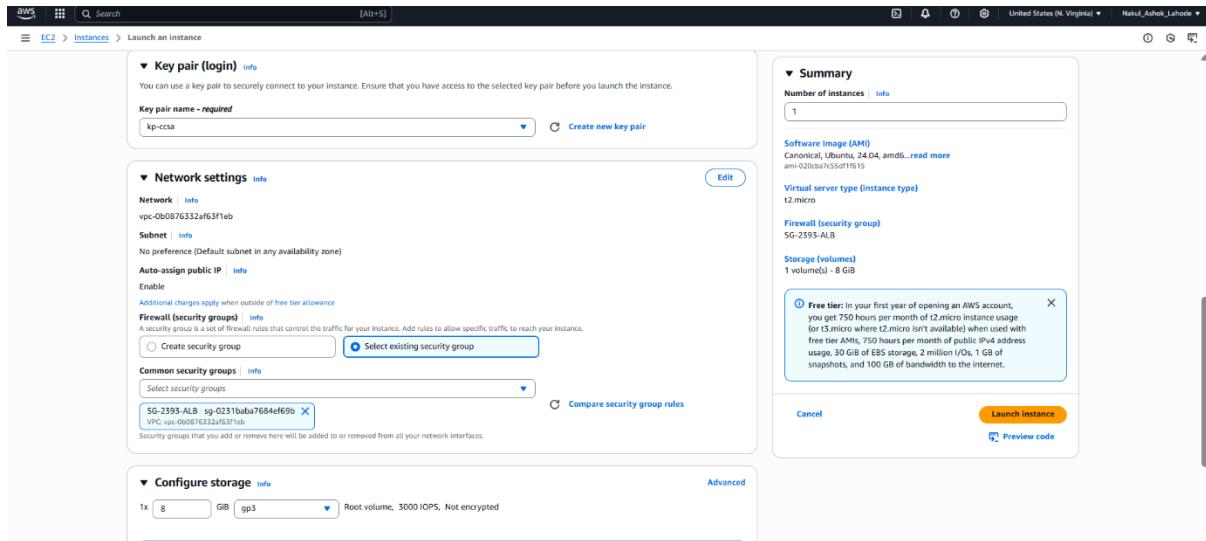


3. Configure the instance with the following:

- Instance Name: Ec2-Monitoring
- Amazon Machine Image (AMI): **Ubuntu**



- Select an existing **Security Group** (or create a new one as needed).



4. Launch the instance and wait for it to initialize.
5. Once running, connect to the EC2 instance using SSH.
6. Install **Apache Server** by running the following commands:

 - sudo apt update
 - sudo apt install apache2 -y

```
ubuntu@ip-172-31-84-112: ~ + 
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-84-112:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
      Active: active (running) since Sat 2025-06-28 12:17:47 UTC; 22s ago
        Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 1888 (apache2)
       Tasks: 55 (limit: 1124)
      Memory: 5.3M (peak: 5.6M)
         CPU: 37ms
        CGroup: /system.slice/apache2.service
            └─1888 /usr/sbin/apache2 -k start
                ├─1811 /usr/sbin/apache2 -k start
                ├─1812 /usr/sbin/apache2 -k start
                └─1813 /usr/sbin/apache2 -k start

Jun 28 12:17:47 ip-172-31-84-112 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jun 28 12:17:47 ip-172-31-84-112 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-84-112:~$
```

7. Enable **Detailed Monitoring** for the instance in the EC2 settings (if not enabled by default).

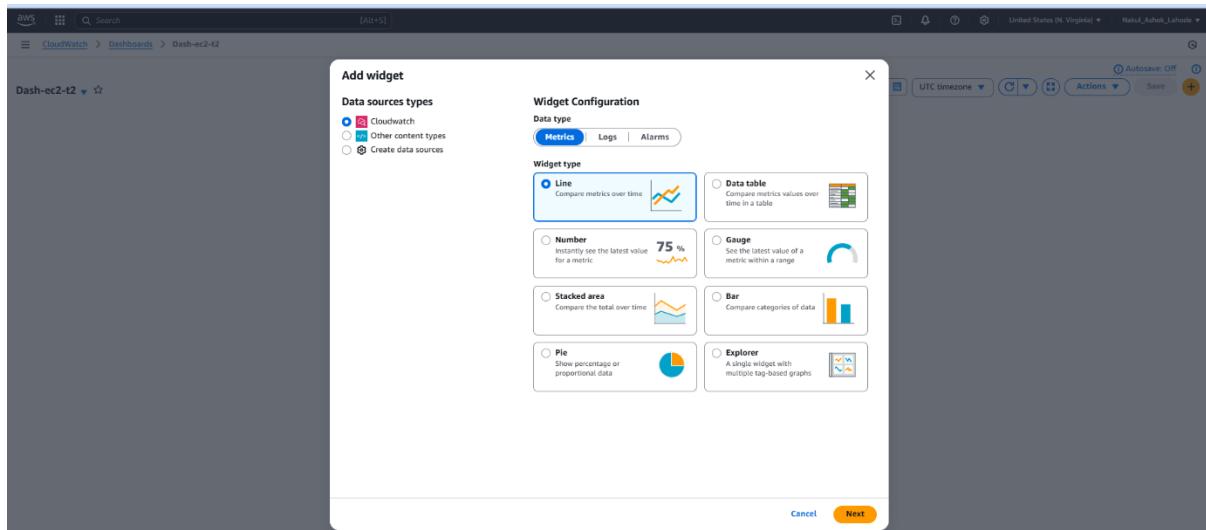
The screenshot shows the AWS CloudWatch Instances console. At the top, there's a search bar and navigation links for 'United States (N. Virginia)' and 'Nakul_Ashok_Lahode'. Below the search bar, a table lists one instance: 'i-035ce63a157baa5dc (Server-monitoring)'. The instance is 'Running' and has an 'Initiating' status check. It's located in 'us-east-1b' with a Public IPv4 of 'ec2-13-222-29-2.com...' and an Elastic IP of '13.222.29.2'. A modal window titled 'Detailed monitoring' is open, explaining that after enabling monitoring, data will be available in 1-minute periods. It shows the instance ID 'i-035ce63a157baa5dc (Server-monitoring)' and a checkbox for 'Enable' detailed monitoring, which is checked. A note says 'After you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance. Additional charges apply.' There are 'Cancel' and 'Confirm' buttons at the bottom of the modal.

Step 2: Create a CloudWatch Dashboard

1. Navigate to **CloudWatch** in the AWS console.
2. Click on “Create dashboard”.

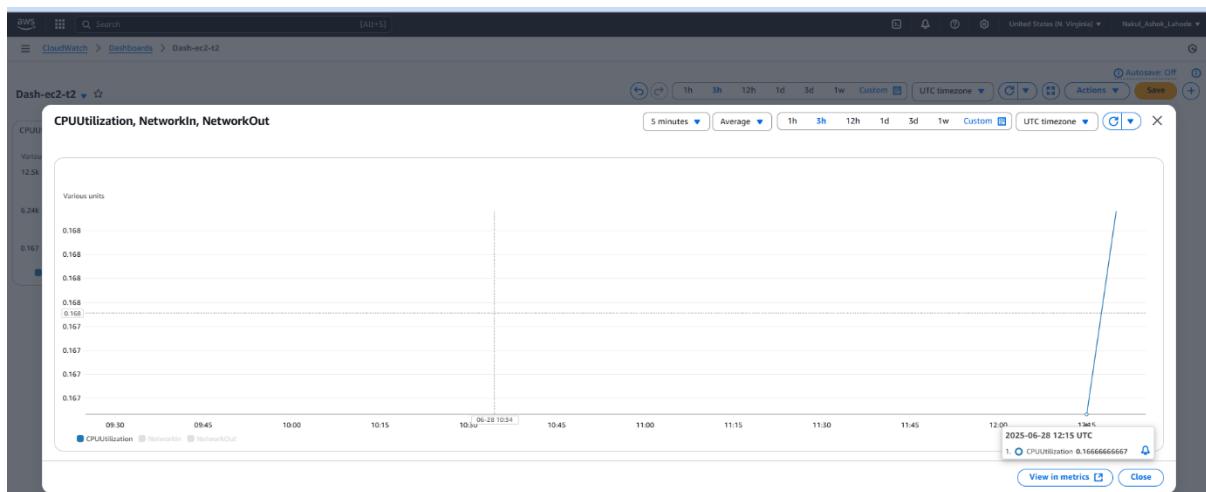
The screenshot shows the AWS CloudWatch Dashboards console. The left sidebar has sections for 'Dashboards', 'Al Operations', 'Alarms', and 'Logs'. The main area is titled 'Custom dashboards' and shows a table for 'Custom Dashboards (0)'. The table has columns for 'Name', 'Sharing', and 'Favorite'. A note says 'No dashboards' and 'You have not created any dashboards.' with a 'Read more about Dashboards' link. At the bottom right, there are 'Share dashboard', 'Delete', and 'Create dashboard' buttons. A 'Create dashboard' button is also located in the center of the dashboard area.

3. Enter a meaningful name for your dashboard (e.g., EC2-Monitoring-Dashboard).
4. Choose the "Line" widget type to visualize metrics.
5. In the metric selector:
 - Choose **EC2 metrics**.
 - Select the instance you launched.
6. Add the following key metrics to the graph:
 - **CPUUtilization**
 - **NetworkIn**
 - **NetworkOut**

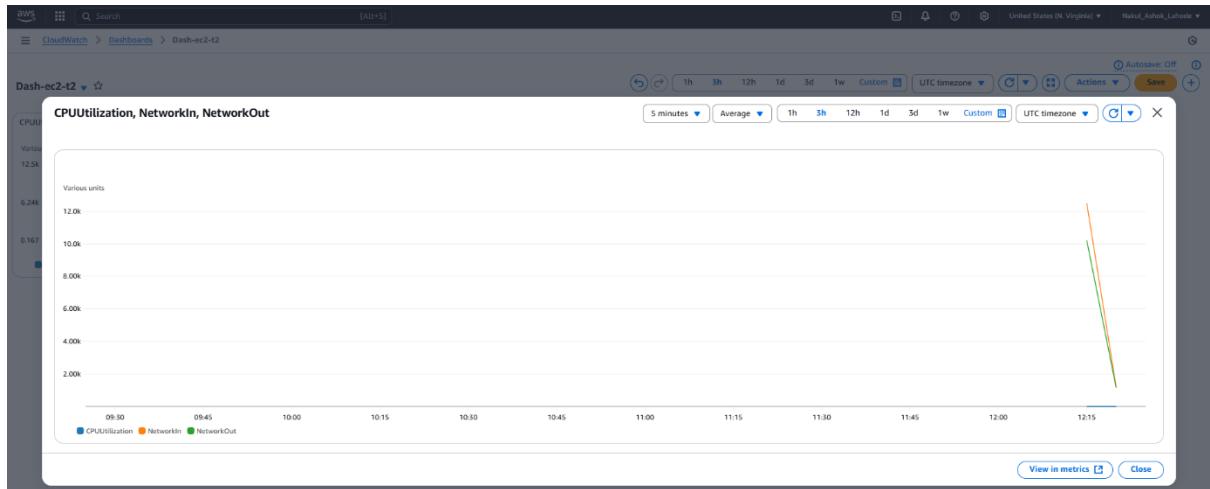


Step 3: Monitor Resources Using the Dashboard

1. Open the dashboard you created.
2. Review the **live monitoring** data for the EC2 instance.
 - o CPU usage trends



- Incoming and outgoing network traffic



3. Use this visualization to:

- Detect performance bottlenecks
- Track system health