**High CPU Utilization Monitoring and Visualization using AWS CloudWatch and Grafana**

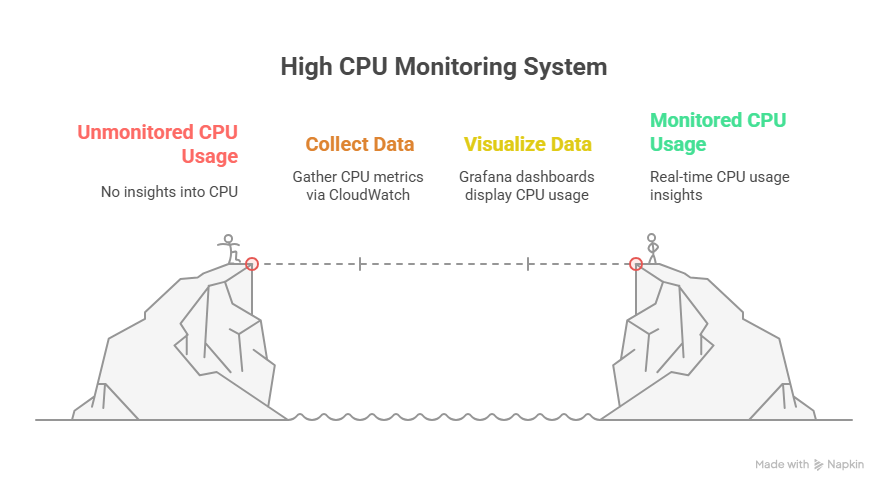
# Table of Contents

1. Introduction
2. Setup Checklist
3. Problem Statement
4. Objective
5. Technology Used
6. Guidelines on the Functionality to be Built
7. Implementation
8. Abstract of the Project
9. Summary of the Functionality to be Built

10. Appendix (Source Files)

**Introduction**

This document describes a project that aims to create a system for monitoring high CPU usage on AWS EC2 instances. The system will collect live data about how much CPU power is being used by the virtual machines (EC2 instances). It uses AWS CloudWatch to gather this performance data continuously. The collected information is then displayed in easy-to-understand charts and graphs using Grafana dashboards. This setup helps users keep an eye on the system’s performance, identify any sudden increase in CPU usage, and take action before it affects the system.



**Setup Checklist**

Hardware:

* Any system capable of accessing AWS Console
* Software:
* AWS Account
* EC2 Instance (Amazon Linux 2)
* Grafana installed on EC2
* Internet connectivity

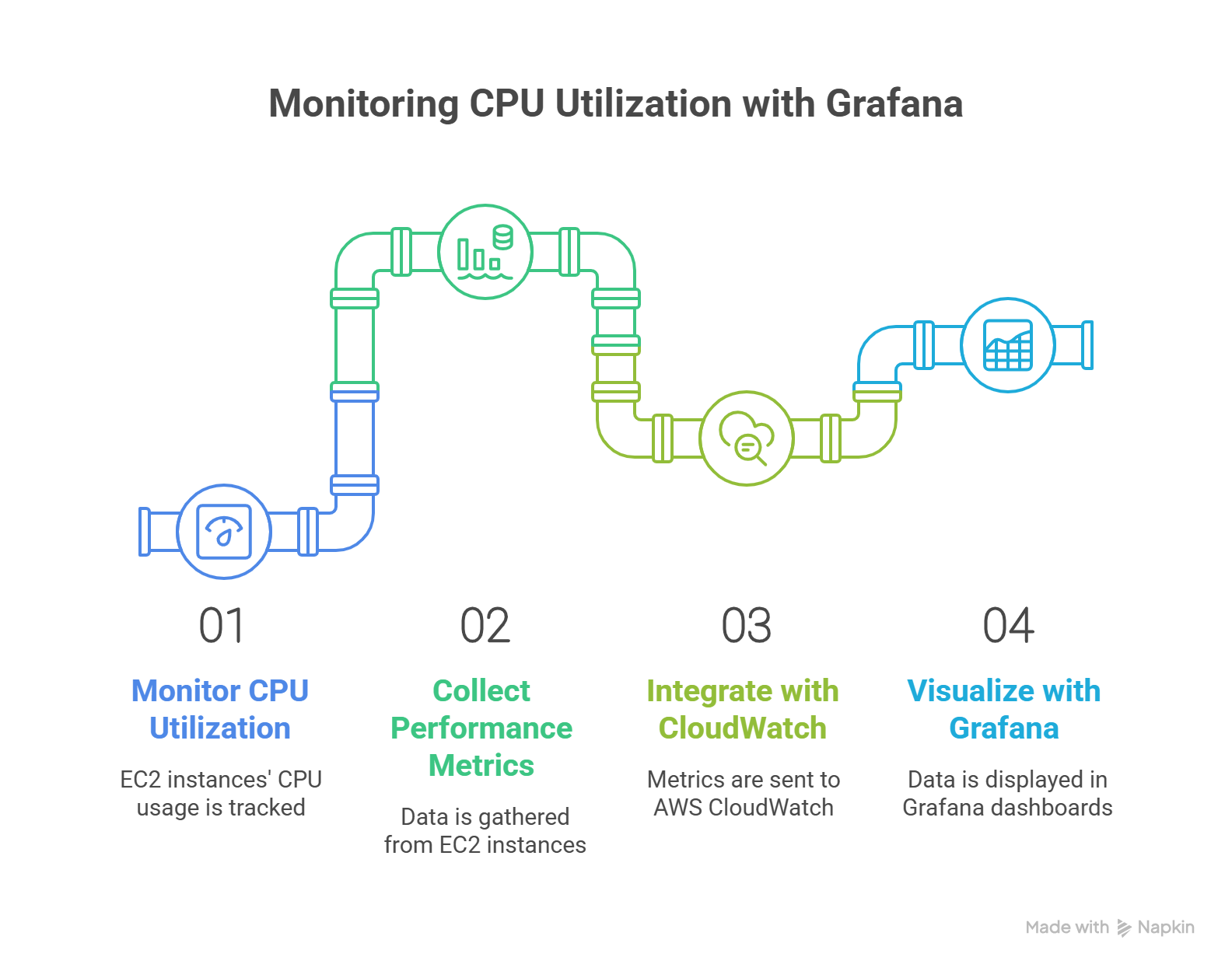
Software:

* Chat GPT
* Google
* Browser
* AWS Services

1. EC2
2. Grafana
3. CloudWatch

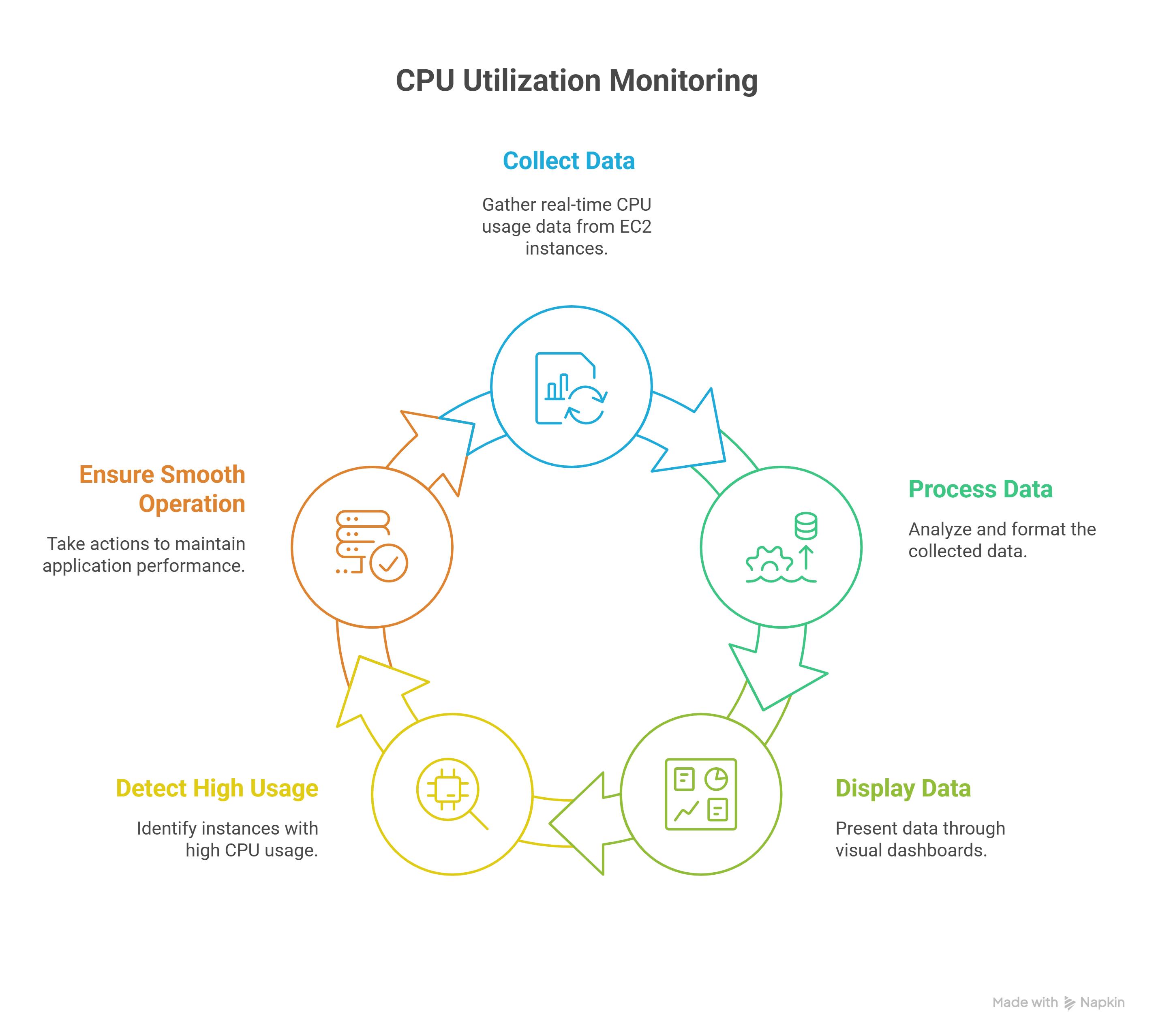
**Problem Statement**

Monitor CPU utilization of EC2 instances and visualize performance metrics using Grafana via CloudWatch integration.



**Objective**

This project aims to build a cloud-native monitoring system that keeps track of the CPU usage of Amazon EC2 instances. It will continuously collect real-time data and display it through easy-to-understand visual dashboards. The main goal is to help users monitor how much CPU their servers are using, detect high usage quickly, and ensure that their applications are running smoothly.



**Technology Used**

- AWS CloudWatch

- Amazon EC2

- Grafana

- Linux Shell (for stress testing)

- IAM for access control

Guidelines on the Functionality to be Built

- Ensure EC2 instance has CloudWatch metrics enabled

- Port 3000 (Grafana) must be open

- Access keys should be secured

- Queries in Grafana must target AWS/EC2 and CPUUtilization metric

- Use 1-minute interval for better granularity

**Implementation**

* **Launch EC2 instance with necessary ports open (SSH, Grafana).**

**Task 1: Sign in to AWS Management Console**

1. Click on the **Open Console**button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

**Task 1: Create A Security Group**

In this task, we are going to create a new Security Group in the US East (N. Virginia) region with a unique name enabling ports 22 (SSH),3000 (Grafana) that all EC2 instance can easily acces the grafana on web.

1. Navigate to **EC2** by clicking on the **Services** menu at the top, then click on **EC2** in the **Computational Services** section.

* **Note :** Just check you are log in in US East (N. Virginia) region.

1. In the EC2 dashboard,Navigate to **Network and Security** ,then click on **Security Groups** then click on **Create Security Group**.
2. Now in Basic Details , Enter **Security group name:** Grafana
3. **Security group details :** Grafana Integration in ec2.
4. In Inbound Rules ,Click on **Add Rule.**

* In **Type** Select : **All Traffic**
* It Automateclly select port and protocol.
* In **Source** Select **0.0.0.0/.0** in CIDR Blocks.

6. Then again click on **Add Rule.**

* In **Type** Select : **SSH**
* It Automateclly select port and protocol.
* In **Source** Select **0.0.0.0/.0** in CIDR Blocks.

7. Then again click on **Add Rule.**

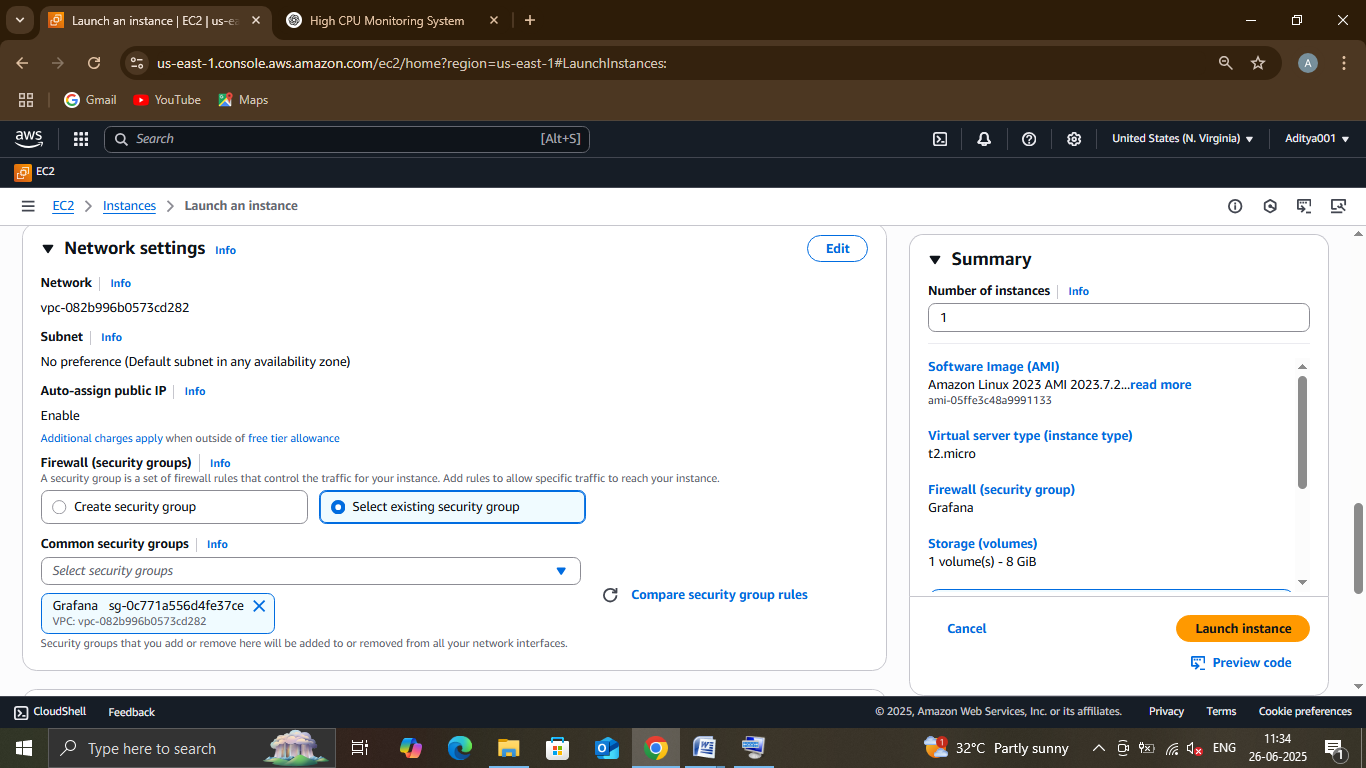
* In **Type** Select : **Custom TCP.**
* In **Prorocol** Select : **TCP**.
* In **Port** Select : **3000**.
* In **Source** Select **0.0.0.0/.0** in CIDR Blocks.

8. Click on **Create Security Group.**

**Task 2: Launch EC2 Instance**

In this task, We are going launch an EC2 in which we monitor the CPU Utilization in Grafana.

1. In AWS Console and Navigate to EC2 Services.
2. Click on **Launch Instance.**
3. Choose Amazon Linux 2 in AMI.
4. Choose t2.micro (Free tier eligible).
5. Create New Key Pair and try using .pem to avoid unnecessary connection error.
6. In **Network Setting** go to **Firewall (Security Group)** Select existing security groups : Grafana ( which we created previously).
7. Now click **Launch Instance.**



**Task 3: Install Grafana in Instance**

Grafana is a visualization tool. We’ll install it on the same EC2 to keep it simple.

1. Connect the EC2 instance and run these commands .

**# Download Grafana**

**sudo yum install -y https://dl.grafana.com/oss/release/grafana-10.4.1-1.x86\_64.rpm**

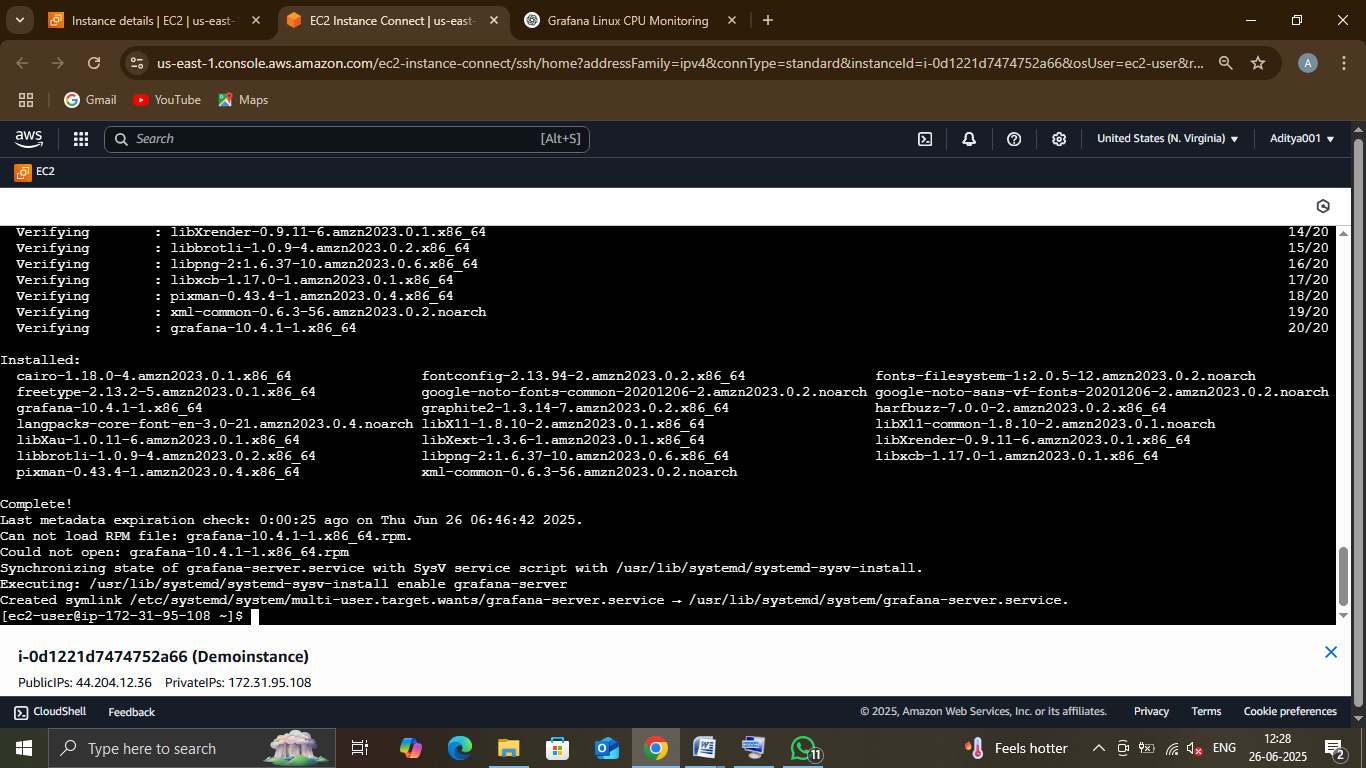
**# Install it**

**sudo yum install -y grafana-10.4.1-1.x86\_64.rpm**

**# Start Grafana**

**sudo systemctl start grafana-server**

**sudo systemctl enable grafana-server**

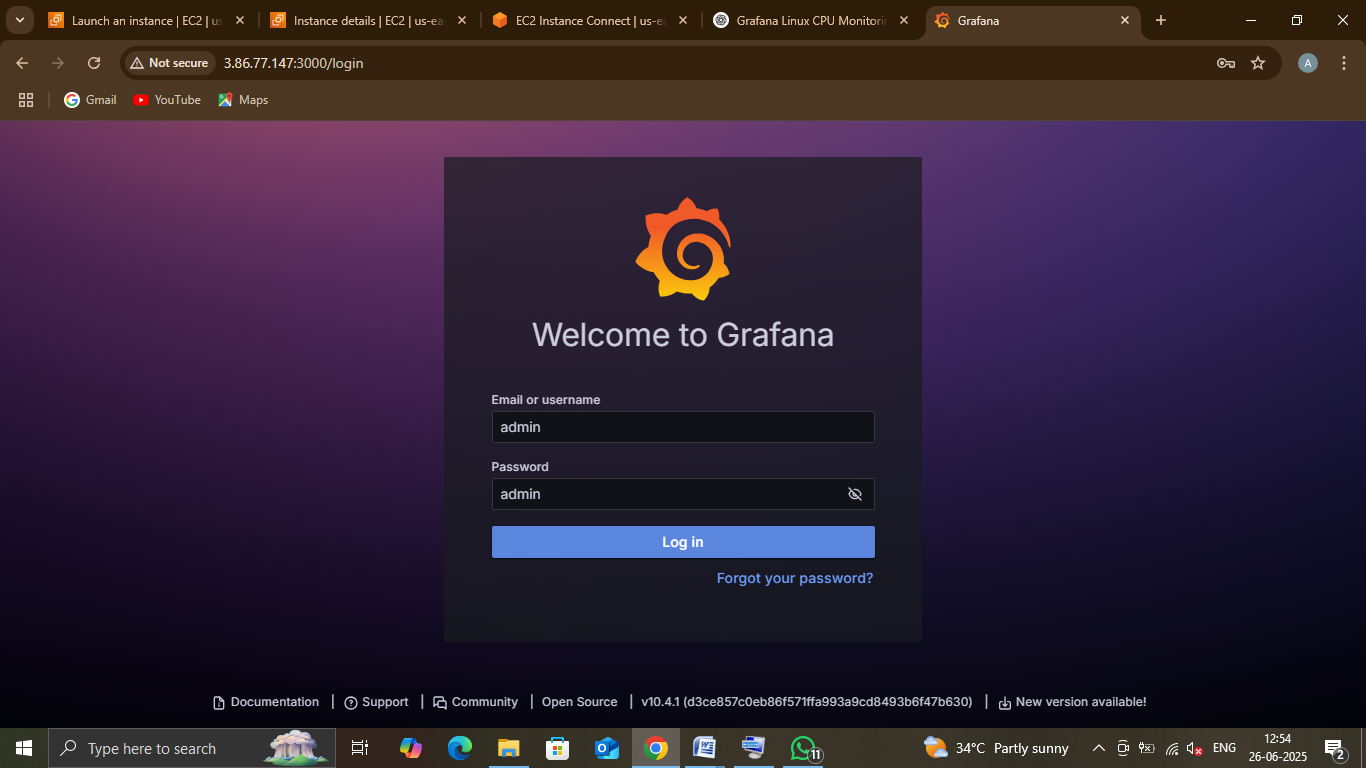
****

**Task 4: Login Grafana in web.**

We’ll now log into Grafana to configure it.

1. Open Browser and paste is link in new tab <http://your-ec2-public-ip:3000>

**Note** : Don’t forget to paste your IP Address of instance you see this .



2. Login:

* Username: admin
* Password: admin
* Then change the password.

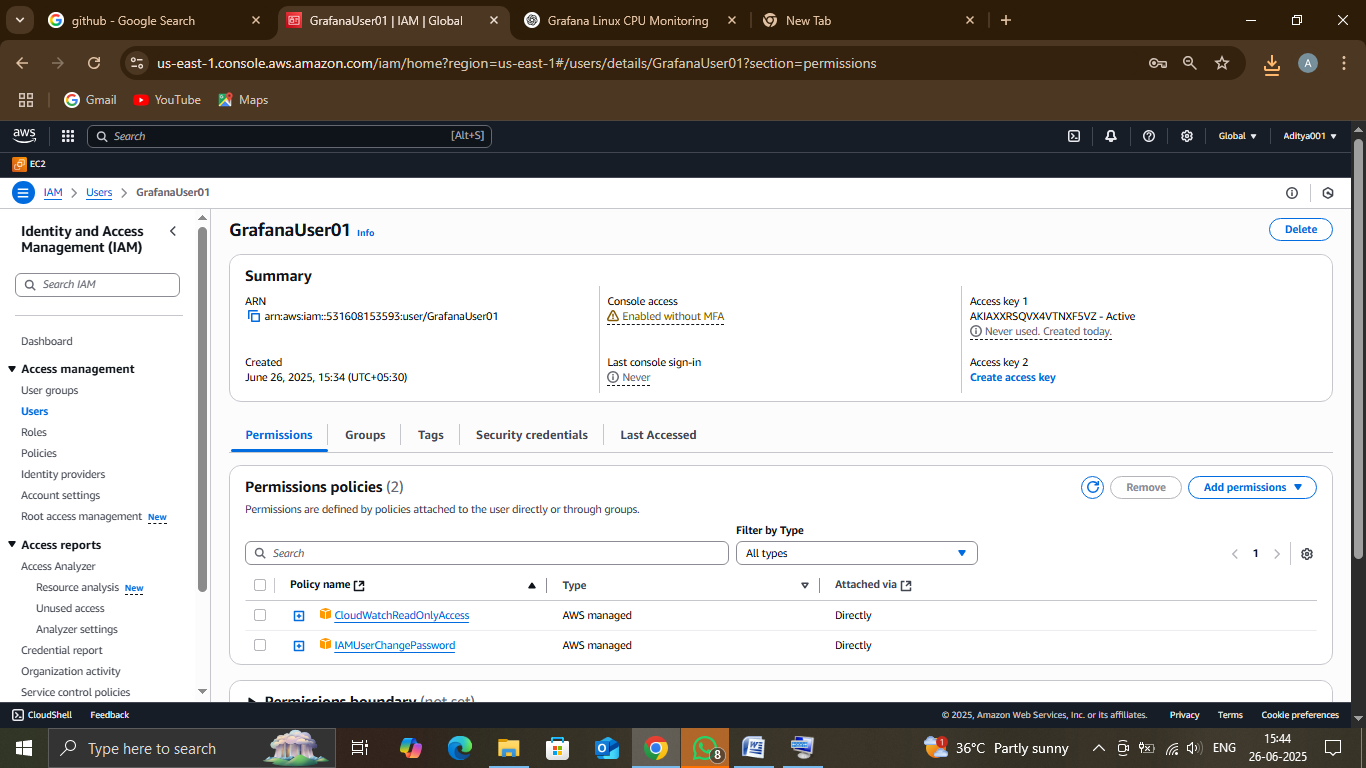
**Task 5: Create A new IAM User for Grafana.**

In this step we are going to create a new **IAM user** to connect with grafana without encountering any error.

1. Navigate to **IAM** in **Security, Identity, & Compliance.**
2. Navigate to **Users** in **Access Management** and click on **Create User.**
3. In **Step 1 :-** Enter **User Name : GrafanUser01.**

**Note-** Check the **Provide user access to the AWS Management Console** and the check the **I want to create an IAM user**  Then Create a **Custom Paswword**.

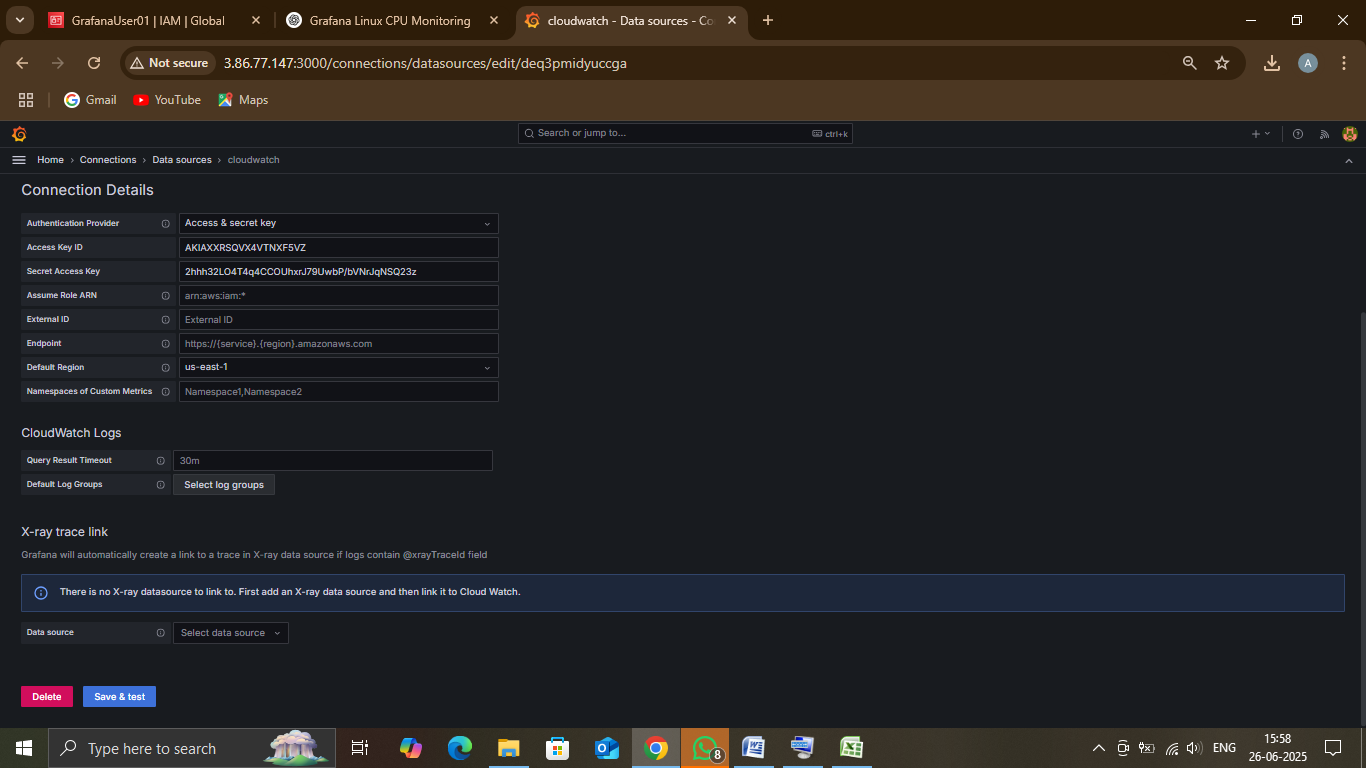
1. Now click on **Next.**
2. In Step 2 :- In **Permission Option** ,Select **Attach policies directly** and attach the **CloudWatchReadOnlyAccess** policy and click on **Next.**
3. Then review all the changers and click on **Create User.**
4. Now , We need to create a access key,So click on **GrafanaUser01**  which we create recently in **IAM dashboard.**
5. Here you find **Create Access Key** in Summary so click on that it will redirect you where you create a new access key.
6. Here you see you are in Step 1:- **Access key best practices & alternatives.**
7. Click **Application running outside AWS** in **Use Case** and click on **Next.**
8. Write **It is to use grafana in ec2 instance** in **Description tag value** and click on **Create access key.**
9. **Now ,**Download the **.csv file** for future use and click on **Done.**



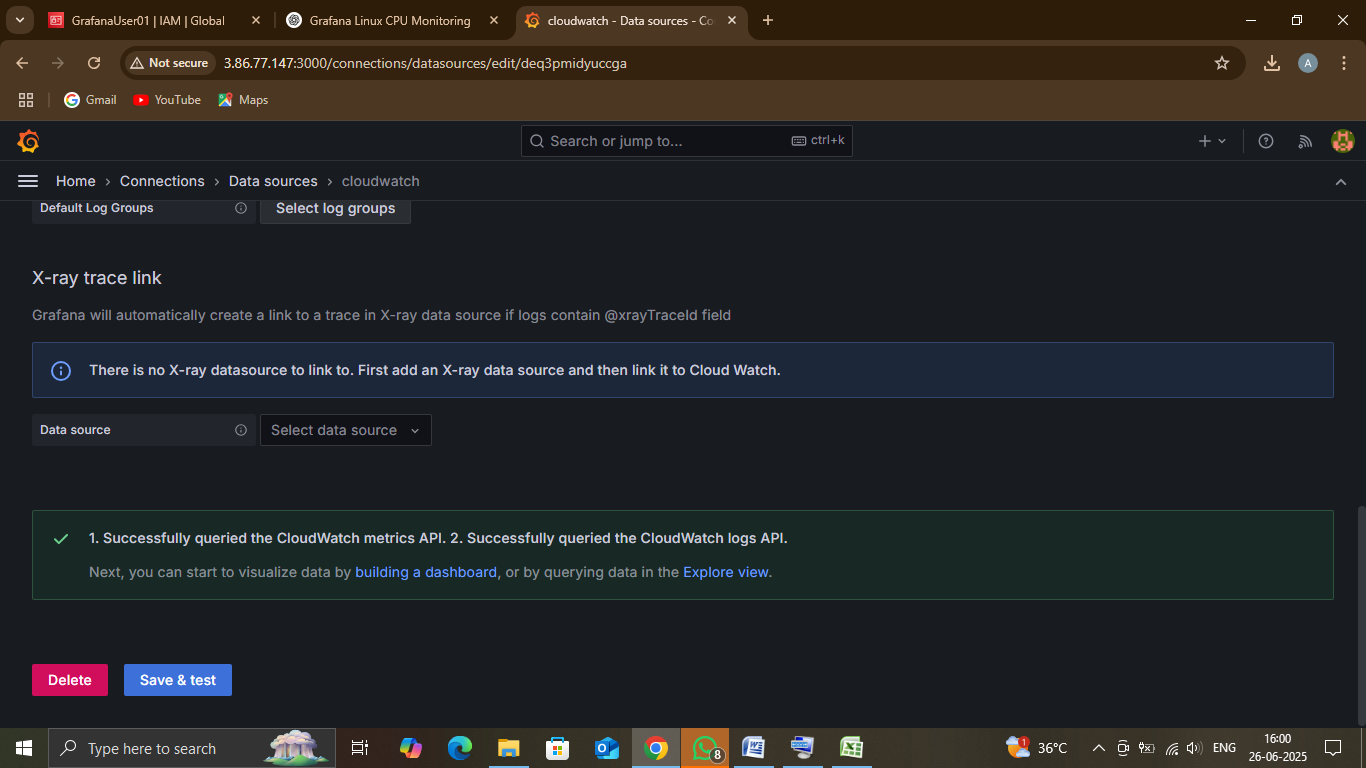
**Task 6: Connect CloudWatch to Grafana.**

In this Grafana can pull metrics from AWS CloudWatch to make graphs.

1. Open Grafana in Browser by pasting the link in new tab <http://your-ec2-public-ip:3000>
2. Search **Data source** in **Search box** in middle of the desktop
3. .In Data source Select **CloudWatch** in the given option.
4. Change **Authentication Provider** to **Access & secret key.**
5. And past the access and secret key which is in the .csv file which we download earlier.
6. Select the **us-east-1** in **Default region.**
7. Then Click on **Save and test.**



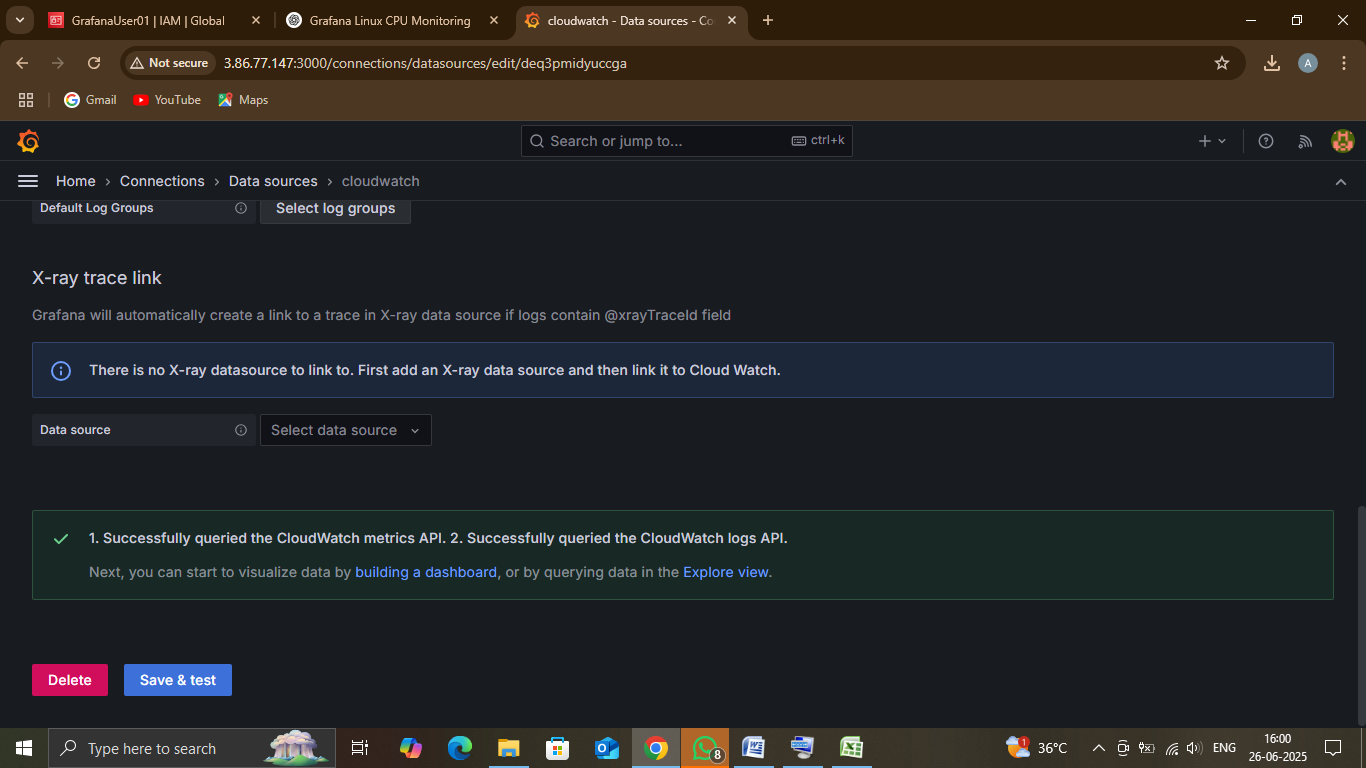
**Note -**  After Clicking on Save and test ,If you will see a popup like this that means your cloudwatch is successfully 😊 .If not then recheck the details you enter.



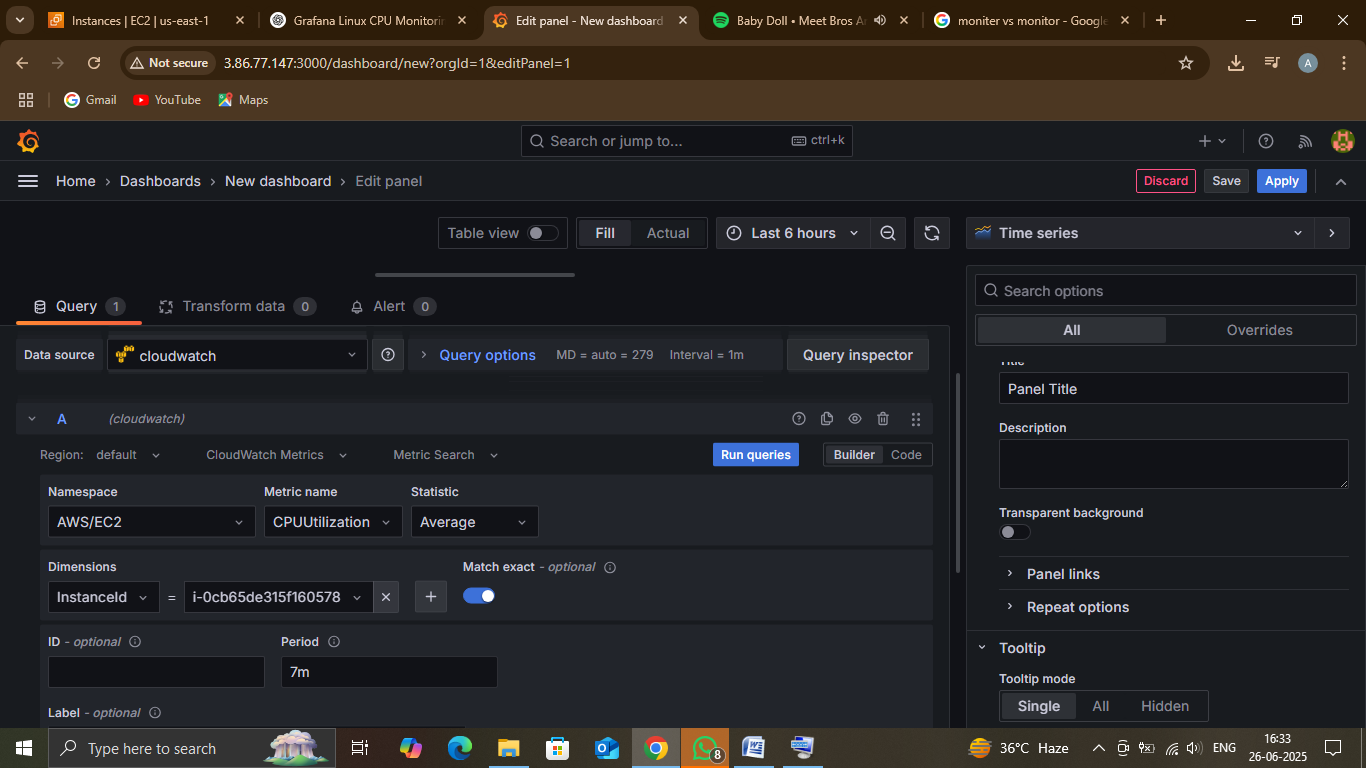
**Task 7: Create Daeshboard in Grafana.**

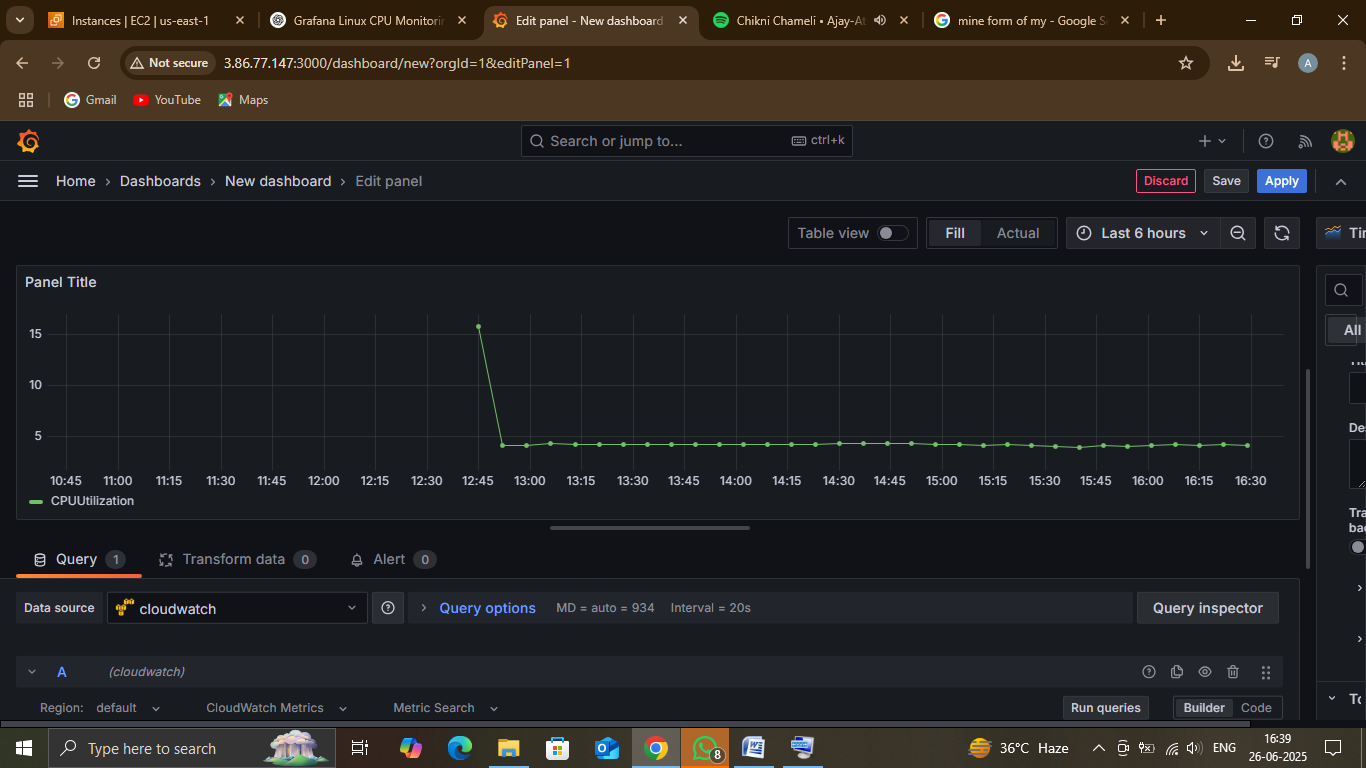
In this step , We are going to setup an Dashboard to monitor the CPU Utilization in EC2.

1. Click on **building a dashboard .**



1. It will redirect to the New Dashboard in Dashboards . Now click on **+Add Visualization.**
2. Select **CloudWatch** in **Data Source**.
3. See **Query Panel** in bottem left.
4. Enter **Data source: cloudwatch.**
5. **Namespace :** AWS/EC2**.**
6. **Metric name :** CPUUtilization.
7. **Statistic :** Average.
8. **Dimensions :** InstanceId.
9. And choose Instance id from options or paste the id.
10. In **Period :** 10m.
11. And hit **Run queries** in right side of panel. After that hit apply to save dashboard.

****

**Note :-** After the runnig the queries this a graph is show like this. Don’t worry mine was bigger because my instance is running for long time.

**Abstract of the Project**

The main goal of this project is to create a monitoring dashboard that shows the **real-time CPU usage** of Amazon EC2 instances. To achieve this, we will connect **AWS CloudWatch**, which collects and stores performance data, with **Grafana**, a powerful tool used to visualize data through interactive graphs and charts.

This project includes the following tasks:

1. **Connecting CloudWatch to Grafana**:  
   Setting up a data source in Grafana that links it to AWS CloudWatch so it can access performance metrics like CPU usage.
2. **Configuring Grafana Panels**:  
   Creating and customizing panels in Grafana to display the CPU usage of EC2 instances in a clear and easy-to-read format.
3. **Real-Time Monitoring**:  
   Ensuring that the dashboard updates automatically and shows the latest CPU performance data without delay.
4. **Performance Visibility**:  
   Making it easy for users or system admins to see how the EC2 instances are performing at any time, so they can quickly detect high usage or issues.

By the end of this project, we will have a live monitoring system that helps us track the CPU usage of EC2 instances using AWS CloudWatch and Grafana in one place.

**Summary of the Functionality to be Built**

- CloudWatch metric collection

- Grafana dashboard creation

- Real-time CPU graph

- IAM integration

- Stress testing for validation

**Reports to be Built**

1. **Report 1: Live CPU Usage Graph**

* Real-time graph showing CPU utilization of selected EC2 instances.
* Useful for immediate system health check.

1. **Report 2: Historical CPU Usage Trends**

* Visualize CPU usage over selectable time ranges (e.g., last 24 hours, 7 days).
* Helps in analyzing long-term patterns and predicting resource needs.

1. **Report 3: Peak Usage Times**

* Identifies specific hours or days when CPU usage spikes occur most often.
* Useful for workload scheduling and optimization.

1. **Report 4: Usage Threshold Alerts (Manual Setup)**

* Alert configuration when CPU usage crosses a defined threshold (e.g., > 80%).
* Sends email, SMS, or Slack notifications for quick action.

1. **Report 5: Multi-instance CPU Comparison (Optional)**

* Compare CPU performance across multiple EC2 instances in one view.
* Great for spotting underperforming or overburdened instances.

**Appendix (Source Files)**

Screenshots and JSON exports of Grafana dashboards can be added here.

Example JSON, Access Key setup steps, and IAM policy configurations can also be included.

Metrics → EC2.