Placement Empowerment Program

Cloud Computing and DevOps Centre

Monitor a Cloud-Based Application Using Prometheus: Deploy a Prometheus instance in the cloud to monitor a web app running on a cloud VM.

Name: Saravana Krishnan J Department: IT

Introduction

Monitoring is a critical aspect of maintaining application performance, reliability, and scalability in modern cloud environments. **Prometheus**, an open-source monitoring and alerting toolkit, is widely used for collecting real-time system metrics and analyzing them through its robust query language, **PromQL**.

This Proof of Concept (POC) demonstrates how to deploy **Prometheus on a cloud-based virtual machine (VM)** to monitor a web application, ensuring system observability and proactive issue detection.

Overview

Prometheus is a powerful, **time-series database** designed for monitoring cloud-native environments. It follows a **pull-based model** to scrape metrics from configured targets and provides flexible querying capabilities for data analysis. This POC focuses on deploying **Prometheus on an AWS EC2 instance** to monitor a web application.

Key Features of Prometheus

✔ **Time-series data collection** with labels for filtering and aggregation

✔ **Powerful querying capabilities** using PromQL

✔**Built-in service discovery** to detect and monitor cloud resources

✔ **Efficient storage and data compression** for high scalability

✔ **Integration with visualization tools** like Grafana

✔**Support for alerting and notifications** using Alertmanager

Objectives

The primary goals of this POC are:

1. **Deploy Prometheus** on a cloud VM (AWS EC2 instance)
2. **Install and configure Prometheus** for system monitoring
3. **Enable web access** to Prometheus through port 9090
4. **Validate the setup** by accessing Prometheus UI
5. **Understand how Prometheus collects and queries system metrics**

Importance

1. Enhances System Observability – Helps in real-time monitoring of system resources and application performance.

2. Proactive Issue Detection – Enables early detection of performance degradation and resource exhaustion.

3. Cloud & Microservices Compatibility – Integrates seamlessly with Docker, Kubernetes, and cloud environments.

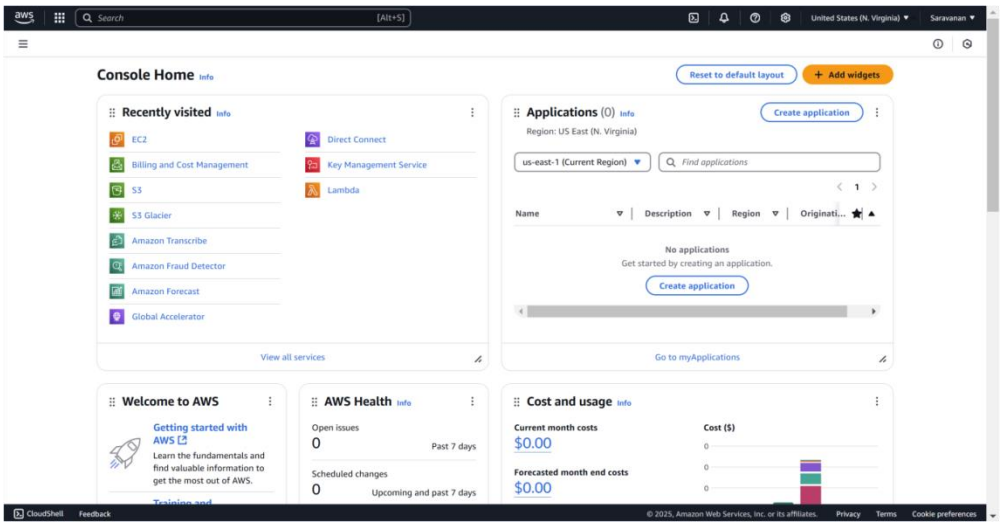
4. Simplifies Debugging & Troubleshooting – Allows detailed analysis of system metrics for issue resolution.

5. Supports Scalable Infrastructure – Works efficiently in dynamic cloud environments with auto-scaling.

Step-by-Step Overview

Step 1:

1. Go to AWS Management Console.
2. Enter your username and password to log in.

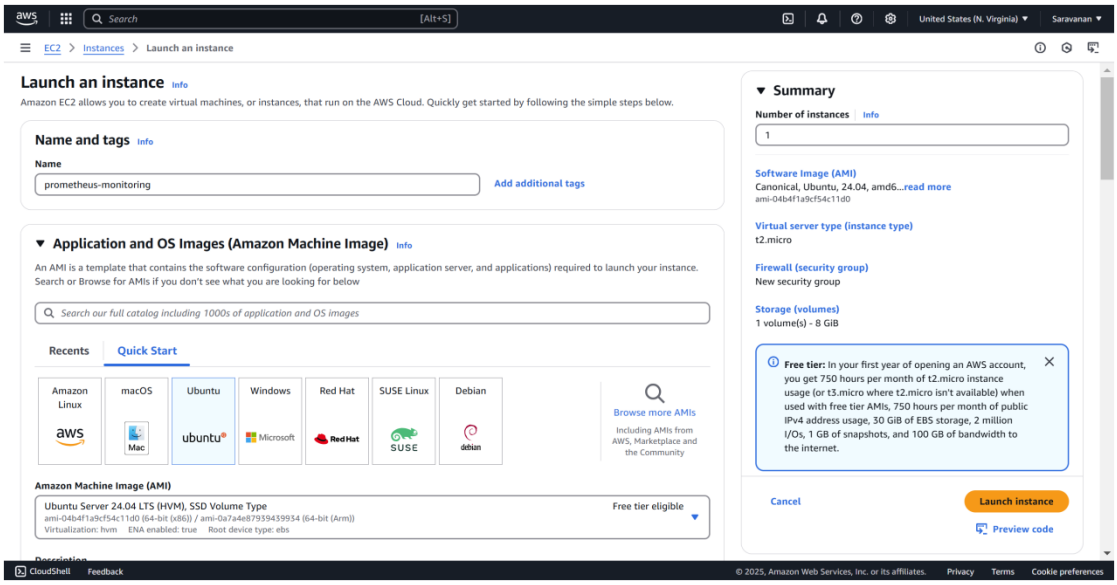


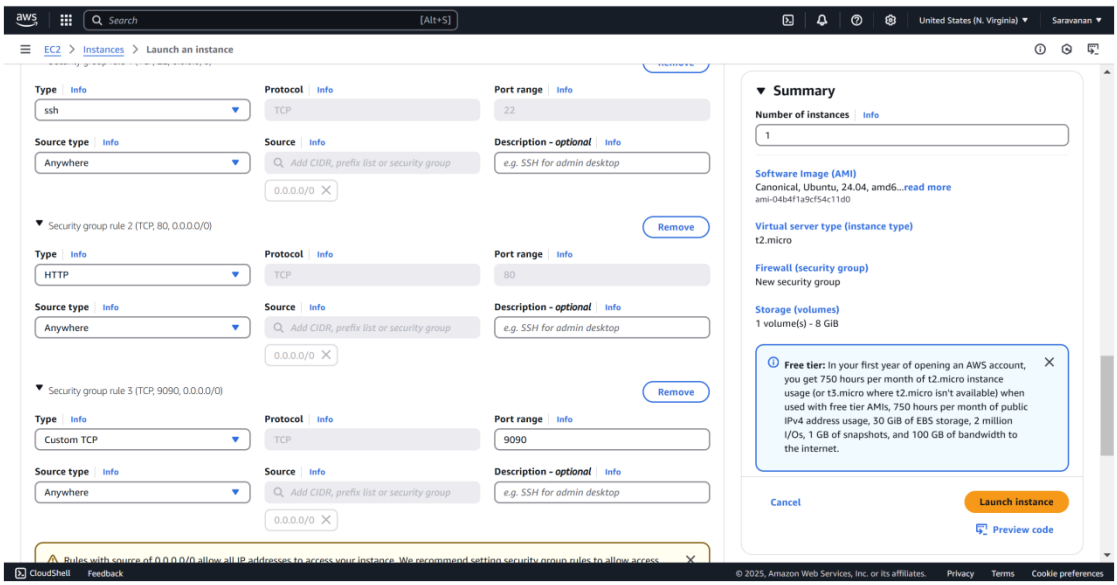
Step 2:

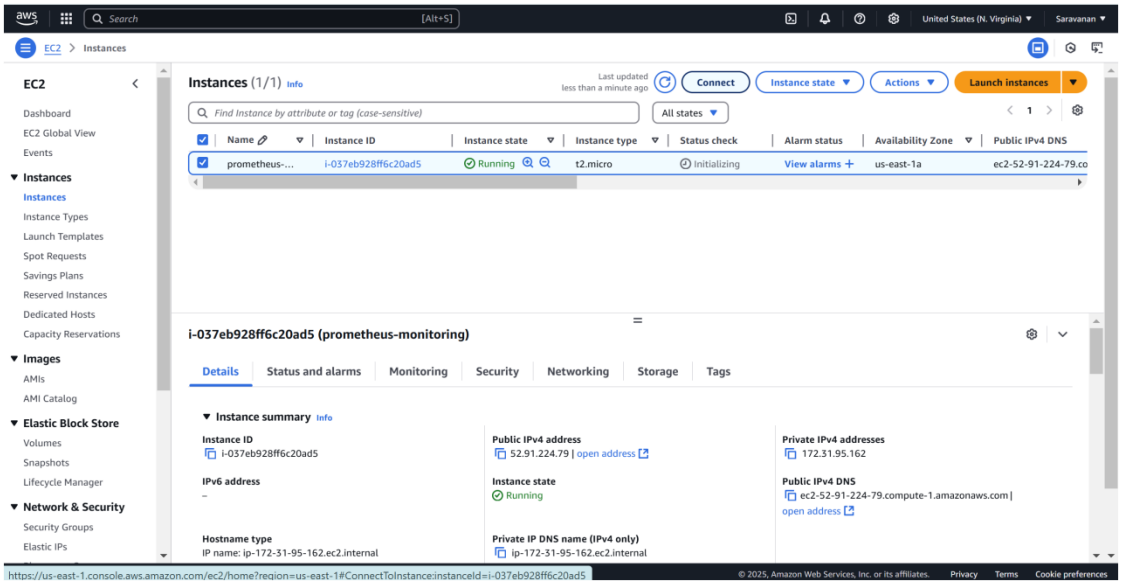
1. Navigate to **EC2 → Launch Instance**.
2. Choose an **Ubuntu** OS.
3. Configure the security group:

   Allow inbound rules for:

   - SSH (Port **22**) → Your IP
   - HTTP (Port **80**) → Anywhere
   - Prometheus (Port **9090**) → Anywhere

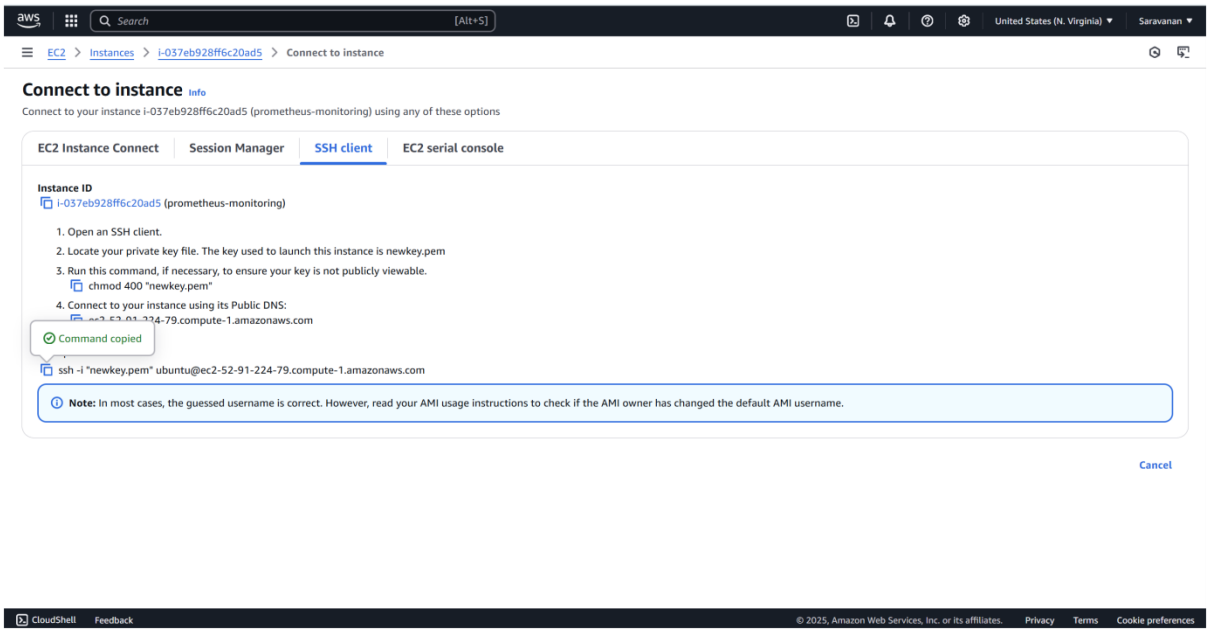

aws
Search
[Alt+S]
United States (N. Virginia)
Saravanan
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
prometheus-monitoring
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 Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-04b4f1a9cf54c11d0 (64-bit (x86)) / ami-0a7a4e87939439954 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-04b4f1a9cf54c11d0
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 of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Search
[Alt+S]
United States (N. Virginia)
Saravanan
EC2 > Instances > Launch an instance
Type Info
ssh
Protocol Info
TCP
Port range Info
22
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or security group
0.0.0.0/0
Description - optional Info
e.g. SSH for admin desktop
Security group rule 2 (TCP, 80, 0.0.0.0/0)
Remove
Type Info
HTTP
Protocol Info
TCP
Port range Info
80
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or security group
0.0.0.0/0
Description - optional Info
e.g. SSH for admin desktop
Security group rule 3 (TCP, 9090, 0.0.0.0/0)
Remove
Type Info
Custom TCP
Protocol Info
TCP
Port range Info
9090
Source type Info
Anywhere
Source Info
Add CIDR, prefix list or security group
0.0.0.0/0
Description - optional Info
e.g. SSH for admin desktop
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-04b4f1a9cf54c11d0
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 of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Search
[Alt+S]
United States (N. Virginia)
Saravanan
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
prometheus-...
i-037eb928ff6c20ad5
Running
t2.micro
Initializing
View alarms +
us-east-1a
ec2-52-91-224-79.co
i-037eb928ff6c20ad5 (prometheus-monitoring)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-037eb928ff6c20ad5
Public IPv4 address
52.91.224.79 | open address
Private IPv4 addresses
172.31.95.162
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-52-91-224-79.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-172-31-95-162.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-95-162.ec2.internal
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-037eb928ff6c20ad5
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Step 3:

Connect to the instance via SSH in Command prompt.




```
C:\Users\Hi>cd Downloads

C:\Users\Hi\Downloads>ssh -i "newkey.pem" ubuntu@ec2-52-91-224-79.compute-1.amazonaws.com
```

Step 4:

Run the following command to update the package list and upgrade existing packages:

sudo apt update && sudo apt upgrade -y

```
ubuntu@ip-172-31-95-162:~$ sudo apt update && sudo apt upgrade -y
```

Step 5:

Run the following command to create a dedicated system user for Prometheus:

sudo useradd --no-create-home --shell /bin/false prometheus

```
ubuntu@ip-172-31-95-162:~$ sudo useradd --no-create-home --shell /bin/false prometheus
```

Step 6:

Switch to the /tmp directory

cd /tmp

```
ubuntu@ip-172-31-95-162:~$ cd /tmp
```

Step 7:

Download the latest Prometheus release

curl -LO https://github.com/prometheus/prometheus/releases/download/v2.46.0/prometheus-2.46.0.linux-amd64.tar.gz

Make sure the file size is correct by running:

ls -lh prometheus-2.46.0.linux-amd64.tar.gz

```
ubuntu@ip-172-31-95-162:/tmp$ curl -LO https://github.com/prometheus/prometheus/releases/download/v2.46.0/prometheus-2.46.0.linux-amd64.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 90.4M  100 90.4M    0     0  90.0M      0  0:00:01  0:00:01 --:--:--  113M
ubuntu@ip-172-31-95-162:/tmp$ ls -lh prometheus-2.46.0.linux-amd64.tar.gz
-rw-rw-r-- 1 ubuntu ubuntu 91M Mar  2 12:28 prometheus-2.46.0.linux-amd64.tar.gz
```

Step 8:

Extract the downloaded file

tar xvf prometheus-2.46.0.linux-amd64.tar.gz

This extracts the files from the tar archive.

```
ubuntu@ip-172-31-95-162:/tmp$ tar xvf prometheus-2.46.0.linux-amd64.tar.gz
prometheus-2.46.0.linux-amd64/
prometheus-2.46.0.linux-amd64/console_libraries/
prometheus-2.46.0.linux-amd64/console_libraries/prom.lib
prometheus-2.46.0.linux-amd64/console_libraries/menu.lib
prometheus-2.46.0.linux-amd64/NOTICE
prometheus-2.46.0.linux-amd64/promtool
prometheus-2.46.0.linux-amd64/prometheus.yml
prometheus-2.46.0.linux-amd64/LICENSE
prometheus-2.46.0.linux-amd64/prometheus
prometheus-2.46.0.linux-amd64/consoles/
prometheus-2.46.0.linux-amd64/consoles/node-disk.html
prometheus-2.46.0.linux-amd64/consoles/node-cpu.html
prometheus-2.46.0.linux-amd64/consoles/prometheus.html
prometheus-2.46.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.46.0.linux-amd64/consoles/node.html
prometheus-2.46.0.linux-amd64/consoles/index.html.example
prometheus-2.46.0.linux-amd64/consoles/node-overview.html
```

Step 9:

Move Prometheus Files:

Run the following commands one by one:

1. Move Prometheus binaries (prometheus and promtool) to /usr/local/bin/

sudo mv prometheus-2.46.0.linux-amd64/prometheus /usr/local/bin/

sudo mv prometheus-2.46.0.linux-amd64/promtool /usr/local/bin/

2. Set the correct permissions

sudo chown prometheus:prometheus /usr/local/bin/prometheus

sudo chown prometheus:prometheus /usr/local/bin/promtool

3. Create directories for Prometheus configuration and data

sudo mkdir /etc/prometheus

sudo mkdir /var/lib/prometheus

4. Move configuration and console files

sudo mv prometheus-2.46.0.linux-amd64/prometheus.yml /etc/prometheus/

sudo mv prometheus-2.46.0.linux-amd64/consoles /etc/prometheus/

sudo mv prometheus-2.46.0.linux-amd64/console_libraries /etc/prometheus/

5. Set ownership for Prometheus files

sudo chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus

```
ubuntu@ip-172-31-95-162:/tmp$ sudo mv prometheus-2.46.0.linux-amd64/prometheus /usr/local/bin/
ubuntu@ip-172-31-95-162:/tmp$ sudo mv prometheus-2.46.0.linux-amd64/promtool /usr/local/bin/
ubuntu@ip-172-31-95-162:/tmp$ sudo chown prometheus:prometheus /usr/local/bin/prometheus
ubuntu@ip-172-31-95-162:/tmp$ sudo chown prometheus:prometheus /usr/local/bin/promtool
ubuntu@ip-172-31-95-162:/tmp$ sudo mkdir /etc/prometheus
ubuntu@ip-172-31-95-162:/tmp$ sudo mkdir /var/lib/prometheus
ubuntu@ip-172-31-95-162:/tmp$ sudo mv prometheus-2.46.0.linux-amd64/prometheus.yml /etc/prometheus/
ubuntu@ip-172-31-95-162:/tmp$ sudo mv prometheus-2.46.0.linux-amd64/consoles /etc/prometheus/
ubuntu@ip-172-31-95-162:/tmp$ sudo mv prometheus-2.46.0.linux-amd64/console_libraries /etc/prometheus/
ubuntu@ip-172-31-95-162:/tmp$ sudo chown -R prometheus:prometheus /etc/prometheus /var/lib/prometheus
```

Step 10:

Check if Prometheus is installed by running:

prometheus –version

```
ubuntu@ip-172-31-95-162:/tmp$ prometheus --version
prometheus, version 2.46.0 (branch: HEAD, revision: cbb69e51423565ec40f46e74f4ff2dbb3b7fb4f0)
  build user:       root@42454fc0f41e
  build date:       20230725-12:31:24
  go version:       go1.20.6
  platform:         linux/amd64
```

Step 11:

Create a new service file

1. Run the following command to open the nano text editor:

sudo nano /etc/systemd/system/prometheus.service

2. Paste the following configuration

3. Save and exit

Press **CTRL + O** and then **Enter**. Then Press **Ctrl+X** .

```
ubuntu@ip-172-31-95-162:/tmp$ sudo nano /etc/systemd/system/prometheus.service
```

```
ubuntu@ip-172-31-95-162: /tr
  GNU nano 7.2                    /etc/systemd/system/prometheus.service *
[Unit]
Description=Prometheus Monitoring System
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file=/etc/prometheus/prometheus.yml \
    --storage.tsdb.path=/var/lib/prometheus \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target

^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Step 12:

1. Reload systemd to recognize the new service

sudo systemctl daemon-reload

2. Enable Prometheus to start on boot

sudo systemctl enable Prometheus

3. Start Prometheus

sudo systemctl start Prometheus

4. Check if Prometheus is running

sudo systemctl status Prometheus

```
ubuntu@ip-172-31-95-162:/tmp$ sudo systemctl daemon-reload
ubuntu@ip-172-31-95-162:/tmp$ sudo systemctl enable prometheus
Created symlink /etc/systemd/system/multi-user.target.wants/prometheus.service → /etc/systemd/system/prometheus.service.
ubuntu@ip-172-31-95-162:/tmp$ sudo systemctl start prometheus
ubuntu@ip-172-31-95-162:/tmp$ sudo systemctl status prometheus
● prometheus.service - Prometheus Monitoring System
     Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-03-02 12:34:29 UTC; 10s ago
   Main PID: 14738 (prometheus)
      Tasks: 6 (limit: 1130)
     Memory: 15.8M (peak: 16.0M)
        CPU: 56ms
     CGroup: /system.slice/prometheus.service
             └─14738 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/var/li>

Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.272Z caller=tls_config.go:274 level=info com>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.272Z caller=tls_config.go:277 level=info com>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.273Z caller=head.go:755 level=info component>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.273Z caller=head.go:792 level=info component>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.275Z caller=main.go:1047 level=info fs_type=>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.275Z caller=main.go:1050 level=info msg="TSD>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.275Z caller=main.go:1231 level=info msg="Loa>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.280Z caller=main.go:1268 level=info msg="Com>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.280Z caller=main.go:1011 level=info msg="Ser>
Mar 02 12:34:29 ip-172-31-95-162 prometheus[14738]: ts=2025-03-02T12:34:29.280Z caller=manager.go:1009 level=info compo>
```

Step 13:

Find the Public IP of your EC2 instance

curl -s ifconfig.me

Copy this **public IP**.

```
ubuntu@ip-172-31-95-162:/tmp$ curl -s ifconfig.me
52.91.224.79ubuntu@ip-172-31-95-162:/tmp$ client_l
```

Step 14:

Open Prometheus in Your Browser

Go to:

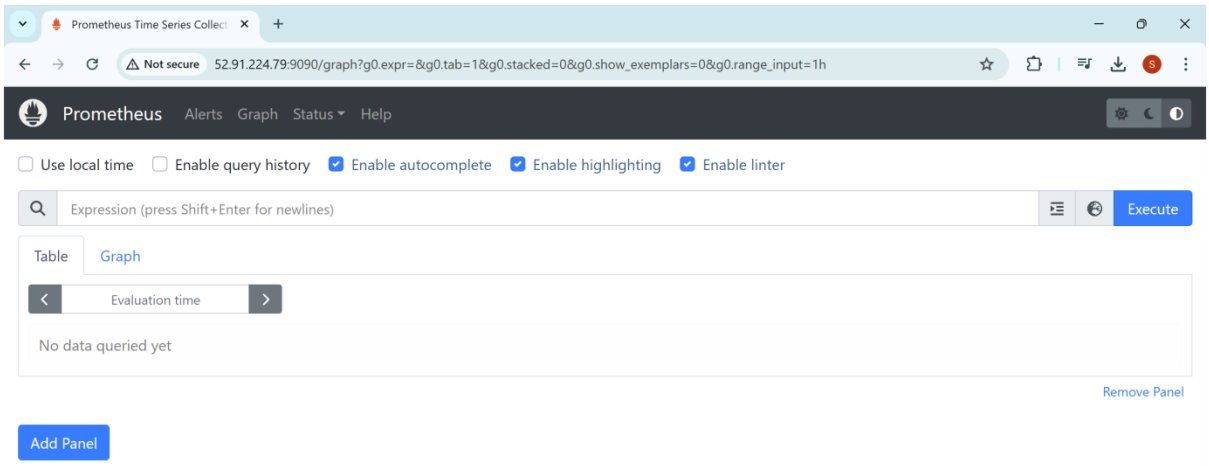
http://<your-public-ip>:9090

Replace <your-public-ip> with the EC2 instance's public IP.

3. Verify Prometheus UI

You should see the Prometheus dashboard!

Eg : http://52.91.224.79:9090



Step 14:

Set Up a Simple Web App with Prometheus Metrics

We'll create a Python web server using Flask and prometheus_client.

Install Required Packages

Run the following command on your VM:

sudo apt update

sudo apt install python3-pip -y

pip3 install flask prometheus_client

```
ubuntu@ip-172-31-95-162:~$ sudo apt update
sudo apt install python3-pip -y
pip3 install flask prometheus_client
```

Step 15:

Now, create a new Python script for your web app: (Open another command prompt and connect it to ssh and do)

nano my_web_app.py

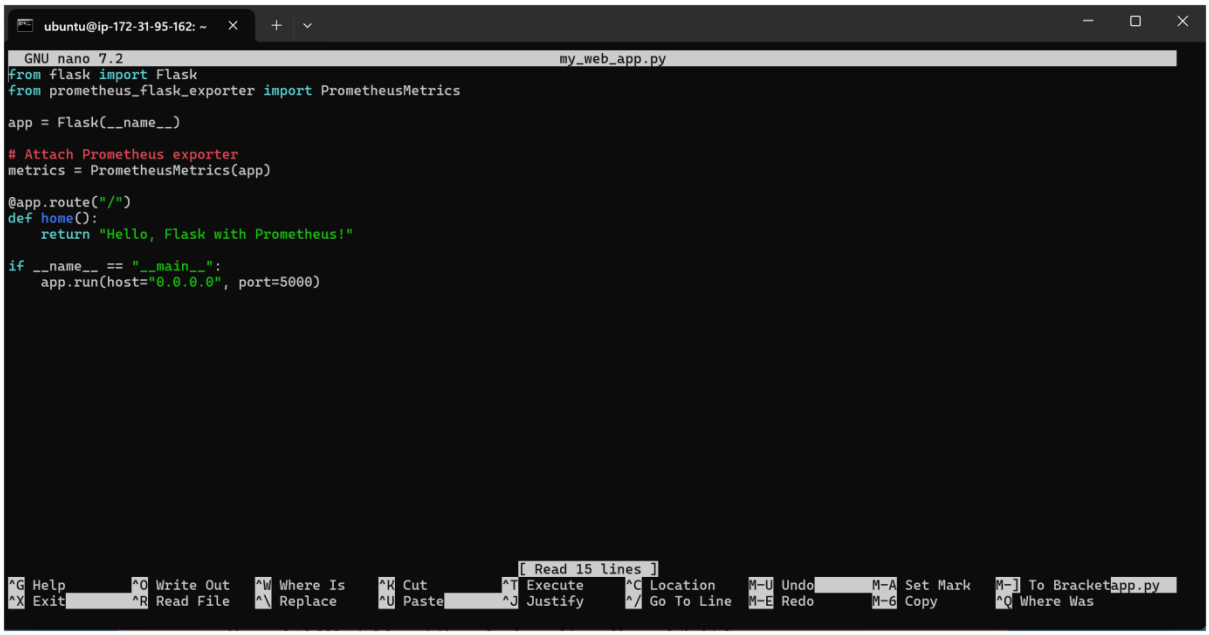
Copy and paste the following code:

This app:

- Runs a web server on **port 5000**.
- Exposes Prometheus metrics on **port 8000**.

Press **CTRL + O** and then **Enter**. Then Press **Ctrl+X** .

```
ubuntu@ip-172-31-95-162:~$ nano my_web_app.py
```



Step 16:

Install python3-venv

sudo apt update

sudo apt install python3-venv -y

Create a Virtual Environment

python3 -m venv myenv

✅ This creates a new virtual environment named myenv.

Activate the Virtual Environment

source myenv/bin/activate

✅ Your terminal should now show (myenv) at the beginning, indicating the virtual environment is active.

Install Flask and prometheus_client

Now install the required packages:

pip install flask prometheus_client

✅This installs Flask and prometheus_client inside the virtual environment

```
ubuntu@ip-172-31-95-162:~$ sudo apt update
```

```
ubuntu@ip-172-31-95-162:~$ python3 -m venv myenv
ubuntu@ip-172-31-95-162:~$ source myenv/bin/activate
(myenv) ubuntu@ip-172-31-95-162:~$ pip install flask prometheus_client
```

Step 17:

Run the web app:

python3 my_web_app.py

Your app is now running!

```
(myenv) ubuntu@ip-172-31-95-162:~$ python3 my_web_app.py
 * Serving Flask app 'my_web_app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.31.95.162:5000
Press CTRL+C to quit
```

Step 18:

Allow HTTP traffic on ports 5000 :

1. Go to **AWS Console → EC2 → Security Groups**.

2. Find the Security Group attached to your instance.

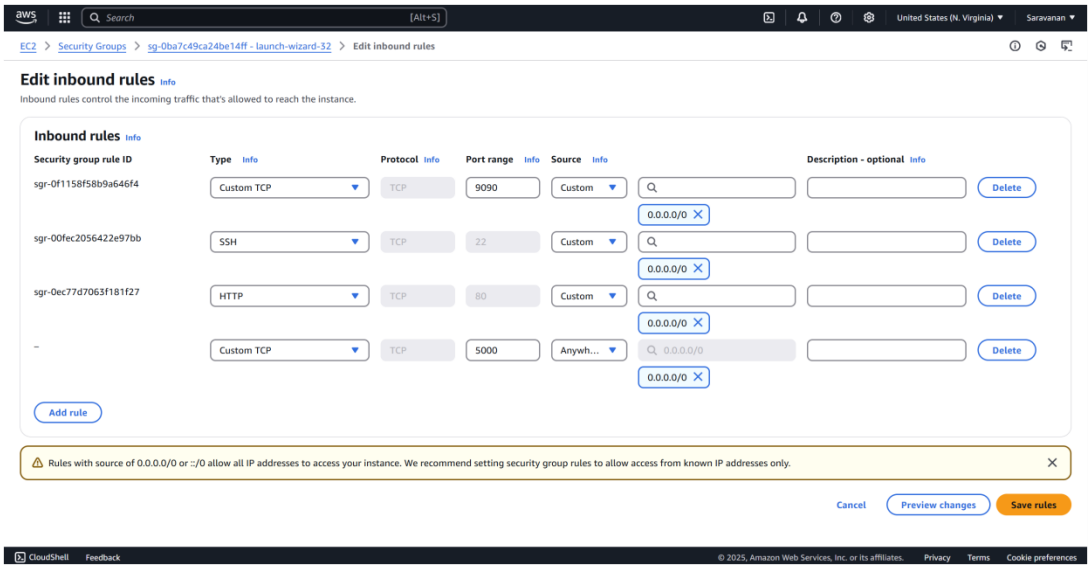
3. Click **Inbound rules → Edit inbound rules**.

4. Add a new rule:

Type: Custom TCP

Port Range: 5000

Source: 0.0.0.0/0 (or your IP for security)

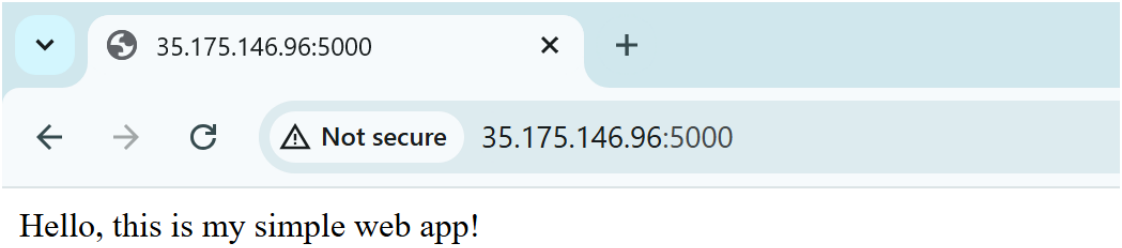


Step 19:

Test Your Flask App

Open a browser and go to:

http://<your-EC2-public-IP>:5000



Step 20:

Edit the Prometheus Configuration File

Run the following command to open the configuration file in **nano** editor:

sudo nano /etc/prometheus/prometheus.yml

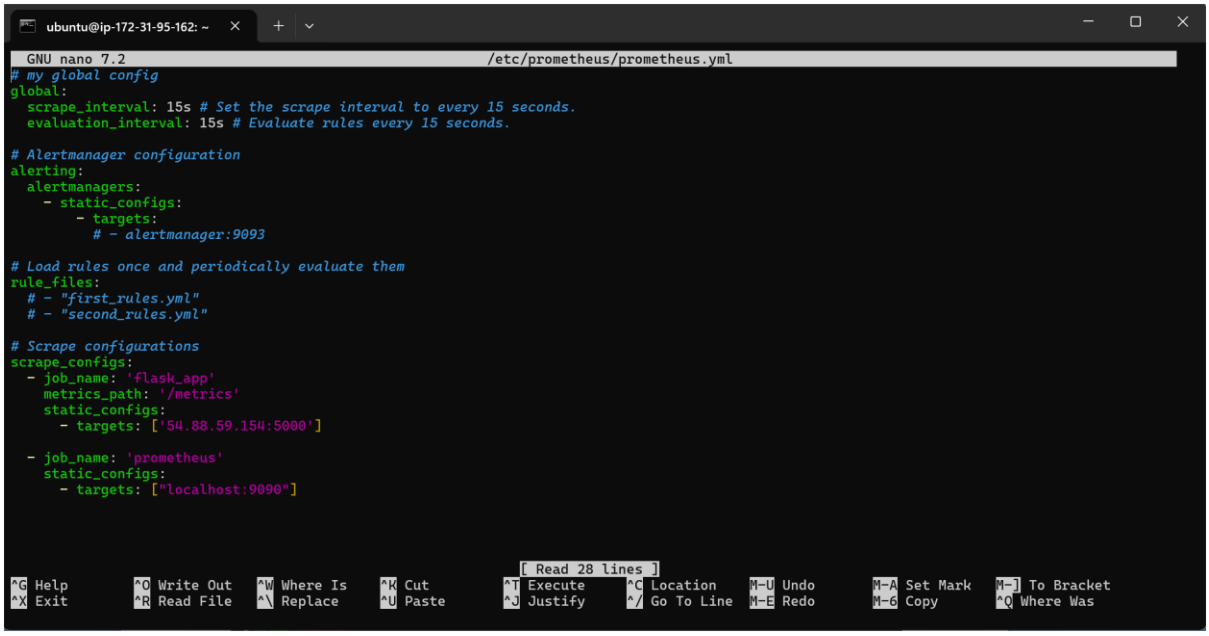
```
ubuntu@ip-172-31-95-162:~$ sudo nano /etc/prometheus/prometheus.yml
```

Step 21:

Add this scrape job at the end of the scrape_configs section:

Ensure **35.175.146.96:5000** is replaced with your **EC2 public IP**

Press **CTRL + O** and then **Enter**. Then Press **Ctrl+X** .



Step 22:

1. Activate Your Virtual Environment

source myenv/bin/activate

Make sure the prompt changes to (myenv).

2. Install Prometheus Flask Exporter

Now, install the package inside the virtual environment**:**

pip install prometheus-flask-exporter

```
ubuntu@ip-172-31-95-162:~$ source myenv/bin/activate
(myenv) ubuntu@ip-172-31-95-162:~$ pip install prometheus-flask-exporter
```

Step 23:

Test /metrics Endpoint

Run:

curl -v http://<Your IP>/metrics

```
(myenv) ubuntu@ip-172-31-95-162:~$ curl -v http://54.88.59.154:5000/metrics
```

Step 24:

Restart Prometheus:

sudo systemctl restart Prometheus

```
ubuntu@ip-172-31-95-162:~$ sudo systemctl restart prometheus
```

Step 25:

Open **Prometheus Web UI**:

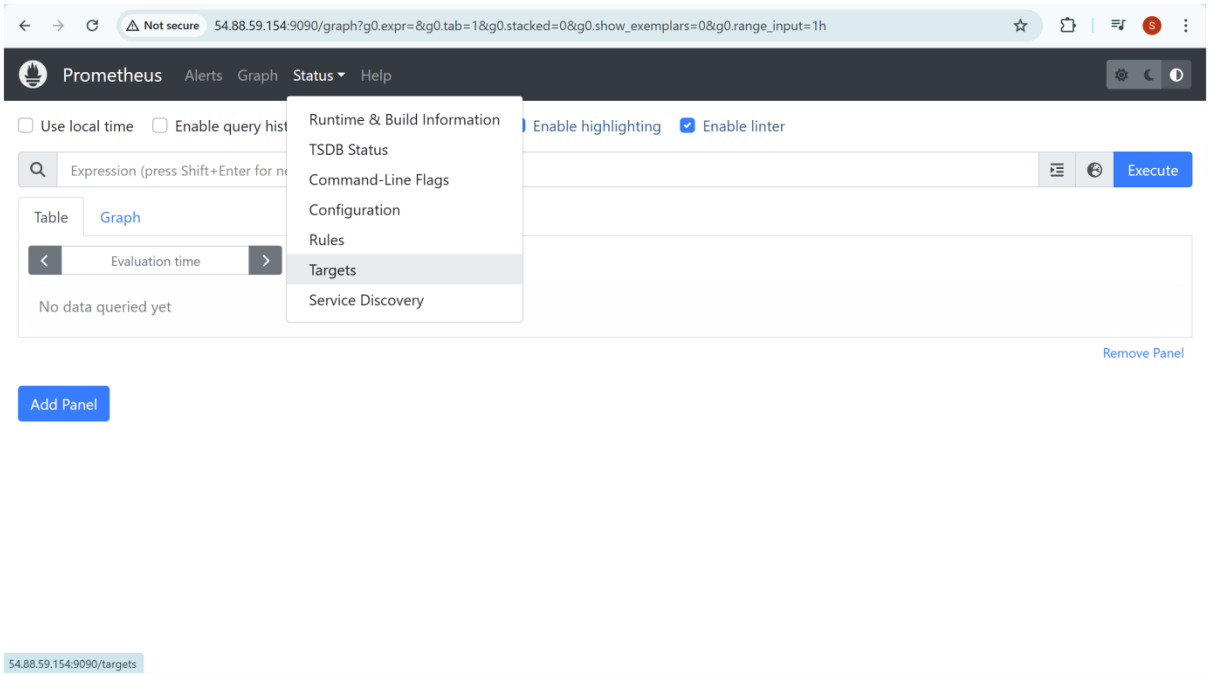
http://<YOUR_PROMETHEUS_IP>:9090

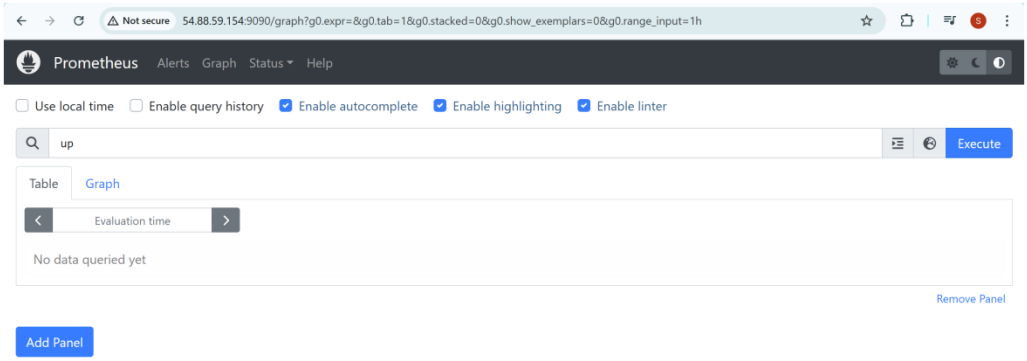
Run up in the Graph tab

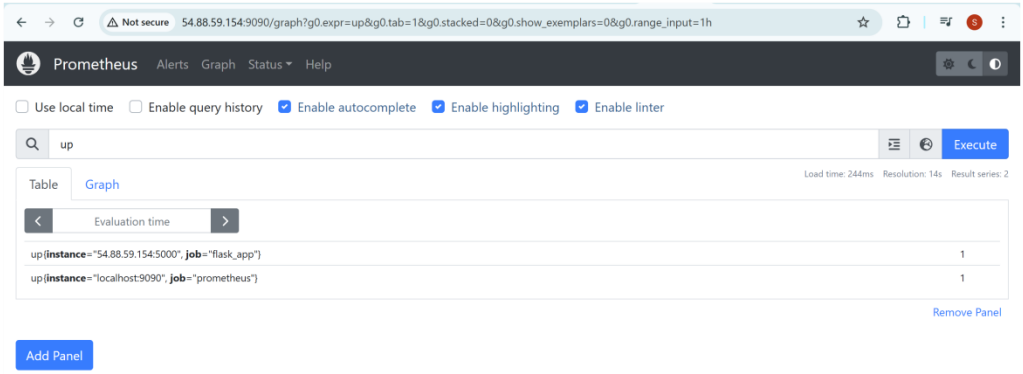
- Open **Prometheus UI → Graph** tab.
- In the query box, type: up and hit **Execute**.
- This should return 1 for each target that is up.

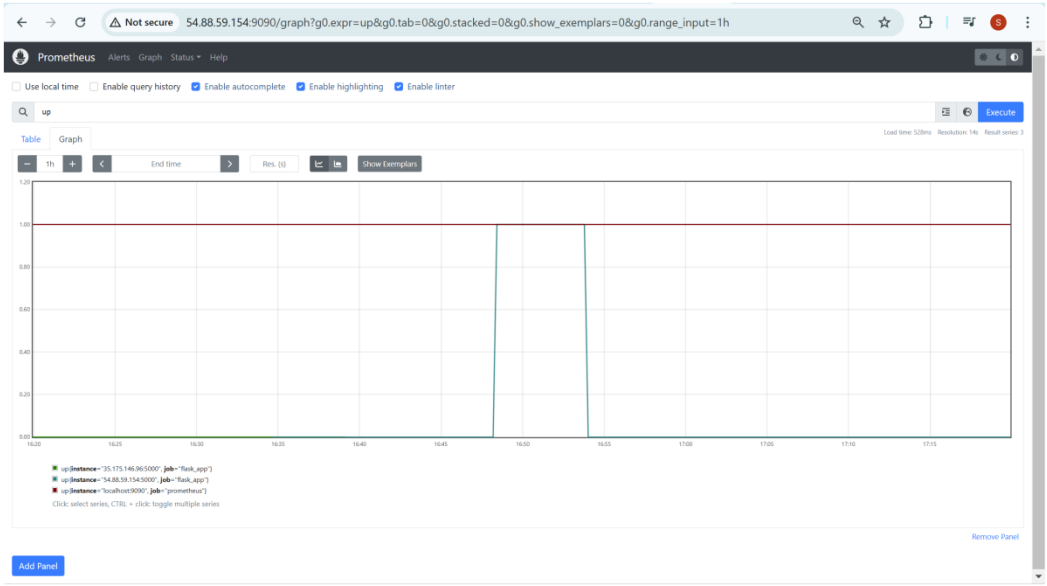
✅Check Active Targets

- Go to **Status → Targets** in Prometheus UI.
- Ensure your targets (like prometheus itself) are **UP** and not DOWN.




Not secure
54.88.59.154:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0&g0.show_exemplars=0&g0.range_input=1h
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
up
Execute
Table
Graph
Evaluation time
No data queried yet
Remove Panel
Add Panel


Not secure
54.88.59.154:9090/graph?g0.expr=up&g0.tab=1&g0.stacked=0&g0.show_exemplars=0&g0.range_input=1h
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
up
Execute
Load time: 244ms Resolution: 14s Result series: 2
Table
Graph
Evaluation time
up{instance="54.88.59.154:5000", job="flask_app"} 1
up{instance="localhost:9090", job="prometheus"} 1
Remove Panel
Add Panel


Not secure
54.88.59.154:9090/graph?g0.expr=up&g0.tab=0&g0.stacked=0&g0.show_exemplars=0&g0.range_input=1h
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
up
Execute
Table
Graph
1h
End time
Res. (s)
Show Exemplars
up{instance="35.175.146.065000", job="flask_app"}
up{instance="54.88.59.154:5000", job="flask_app"}
up{instance="localhost:9090", job="prometheus"}
Click: select series, CTRL + click: toggle multiple series
Remove Panel
Add Panel

PoC is successfully completed

Installed Prometheus on the cloud VM, added the Flask app (54.88.59.154:5000/metrics) as a target in prometheus.yml, and verified successful metric scraping in Prometheus.

Outcomes

By completing this PoC, you will:

1. **Install and Configure Prometheus** – Set up Prometheus on a cloud VM to monitor application metrics.

2. **Integrate Prometheus with a Flask Application** – Expose the Flask app's /metrics endpoint for Prometheus scraping.

3. **Define Prometheus Scrape Targets** – Modify the Prometheus configuration file to include the Flask app as a monitored target.

4. **Validate Prometheus Monitoring** – Query Prometheus to confirm successful metric collection from the Flask application.

5. **Access and Analyze Metrics via Prometheus UI** – Use the Prometheus web interface to visualize and analyze collected data.

6. **Enhance Observability and Monitoring Skills** – Gain hands-on experience in setting up application monitoring using Prometheus.