Task: Setting up Prometheus and Grafana for Monitoring and Visualization
Description:
Set up Prometheus for collecting metrics and Grafana for visualization to monitor the performance and health of your application.
1. Installation and Configuration:
Test Case 1: Verify that Prometheus and Grafana are installed and configured on their respective servers.
Expected Outcome: Both Prometheus and Grafana should be installed and running without errors.

Installation and Configuration:
Prometheus:

Installation: Install Prometheus on its dedicated server by downloading the package suitable for your operating system.

Configuration: Configure Prometheus to scrape metrics from your application and infrastructure. Modify the prometheus.yml configuration file to specify targets and scraping intervals.

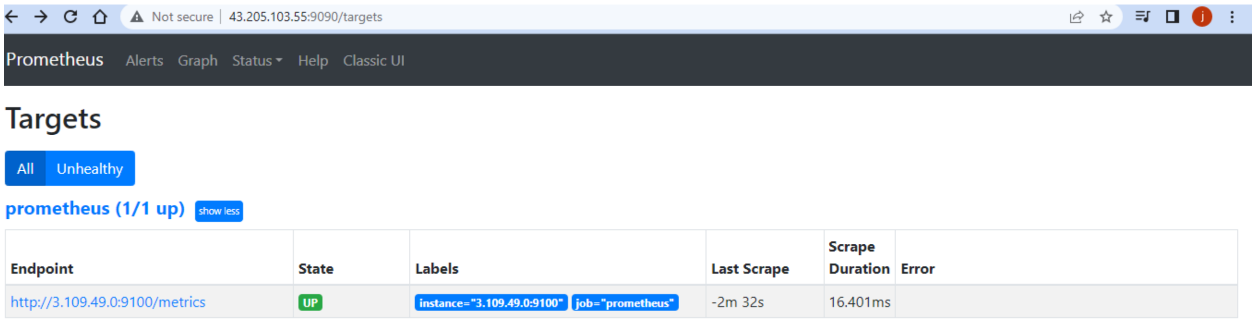
Grafana:

Installation: Install Grafana on its dedicated server by downloading the package suitable for your operating system.

Configuration: After installation, configure Grafana to connect to Prometheus as a data source. Set up data source settings in Grafana to point to your Prometheus instance.

Metrics Collection and Visualization:
Test Case 1:

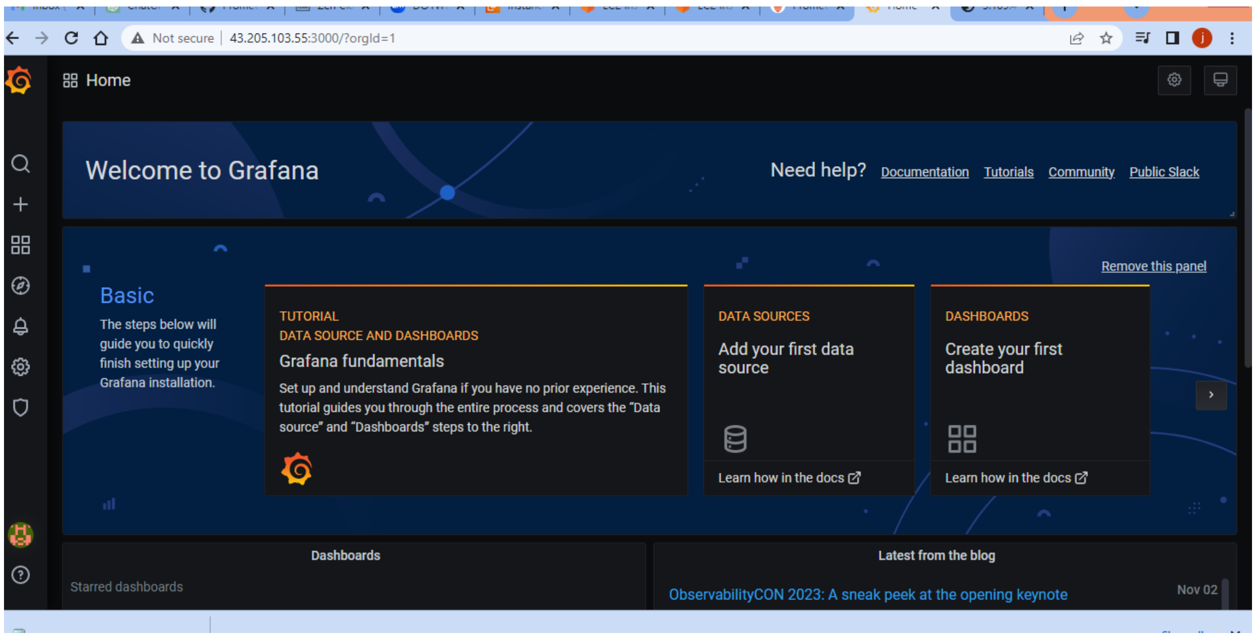
- Check Prometheus and Grafana installations by accessing their web interfaces.
- Prometheus should be reachable at http://prometheus-server-ip:9090.
- Grafana should be accessible at http://grafana-server-ip:3000.
- Both interfaces should load without errors.



Targets

prometheus (1/1 up)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://3.109.49.0:9100/metrics	UP	instance="3.109.49.0:9100" job="prometheus"	-2m 32s	16.401ms	


Not secure | 43.205.103.55:3000/?orgId=1

Home

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

DATA SOURCES
Add your first data source
Learn how in the docs

DASHBOARDS
Create your first dashboard
Learn how in the docs

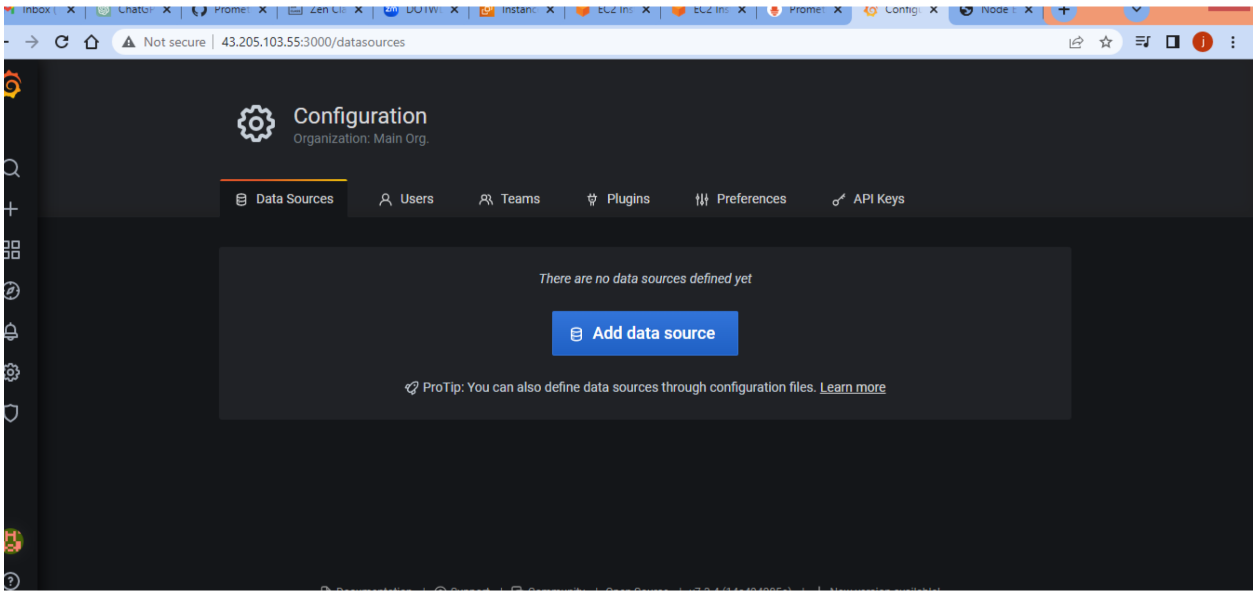
Dashboards
Starred dashboards

Latest from the blog
ObservabilityCON 2023: A sneak peek at the opening keynote Nov 02

2. Metrics Collection and Visualization:
Test Case 2: Confirm that Prometheus collects metrics from the application and infrastructure, and Grafana displays these metrics on a dashboard.
Expected Outcome: Metrics from the application and infrastructure should be visible on a Grafana dashboard, showing real-time data.

Test Case 2:

- Verify that Prometheus is collecting metrics by checking the Prometheus targets page for successful scrapes.
- Create a dashboard in Grafana and add Prometheus as a data source.
- Design panels in Grafana to visualize the collected metrics.
- The dashboard in Grafana should display real-time data from Prometheus.



43.205.103.55:3000/datasources

Configuration

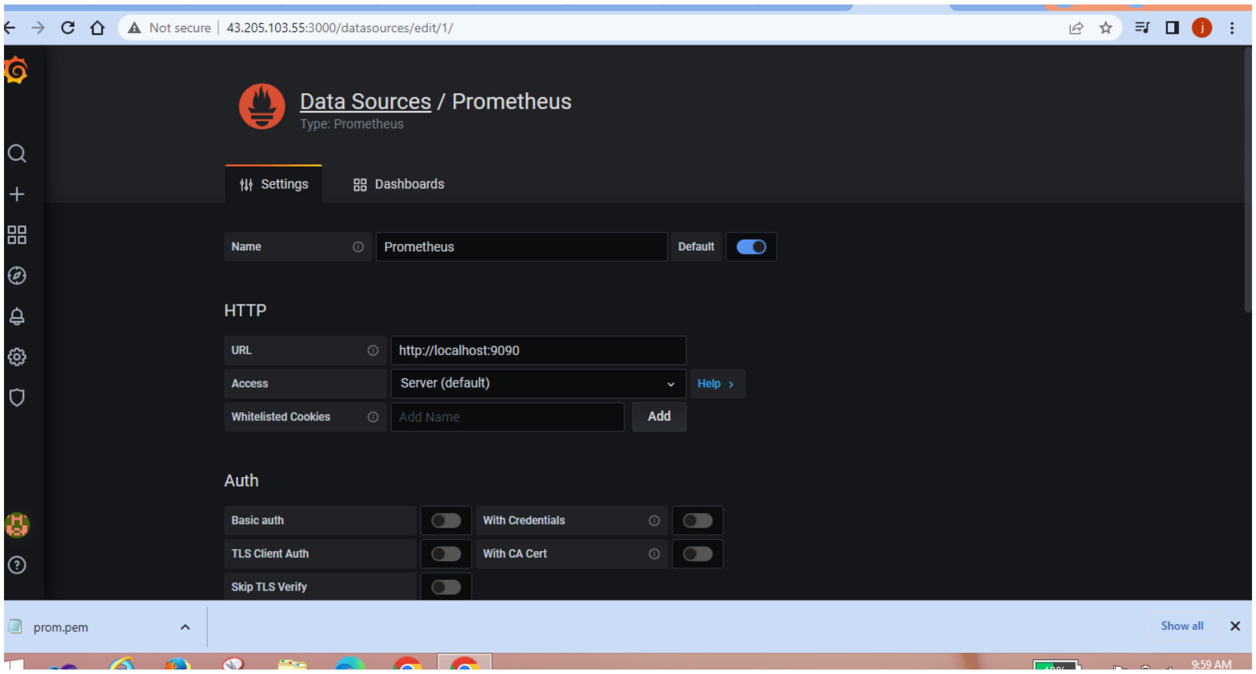
Organization: Main Org.

Data Sources | Users | Teams | Plugins | Preferences | API Keys

There are no data sources defined yet

Add data source

ProTip: You can also define data sources through configuration files. Learn more



43.205.103.55:3000/datasources/edit/1/

Data Sources / Prometheus

Type: Prometheus

Settings | Dashboards

Name	Prometheus	Default	on

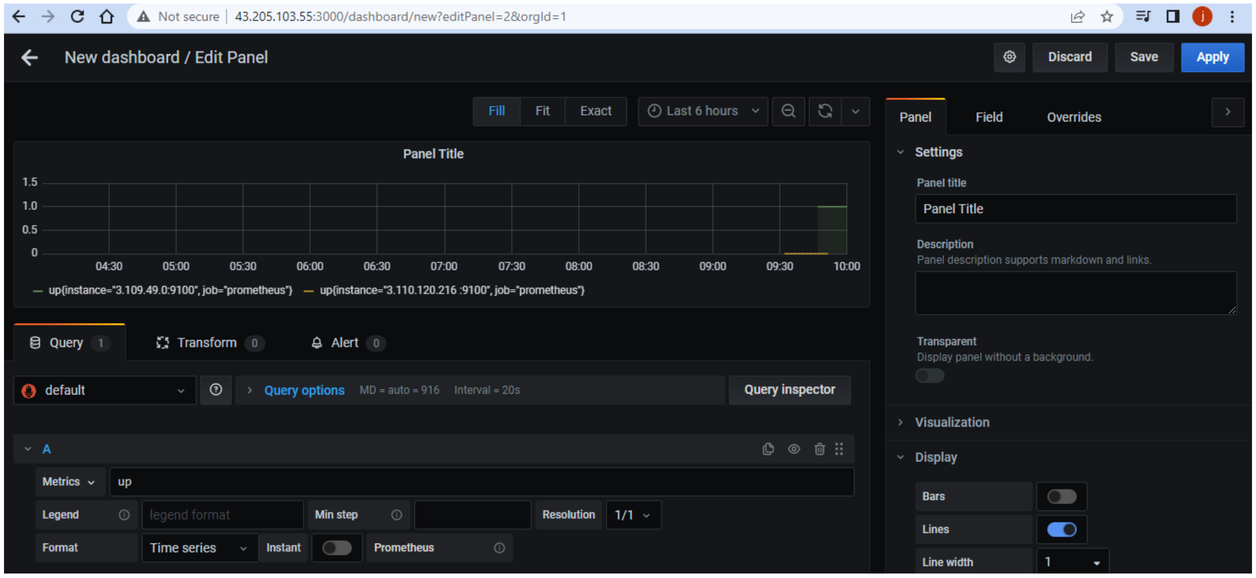
HTTP

URL	http://localhost:9090
Access	Server (default)
Whitelisted Cookies	Add Name

Auth

Basic auth	off	With Credentials	off
TLS Client Auth	off	With CA Cert	off
Skip TLS Verify	off		

prom.pem

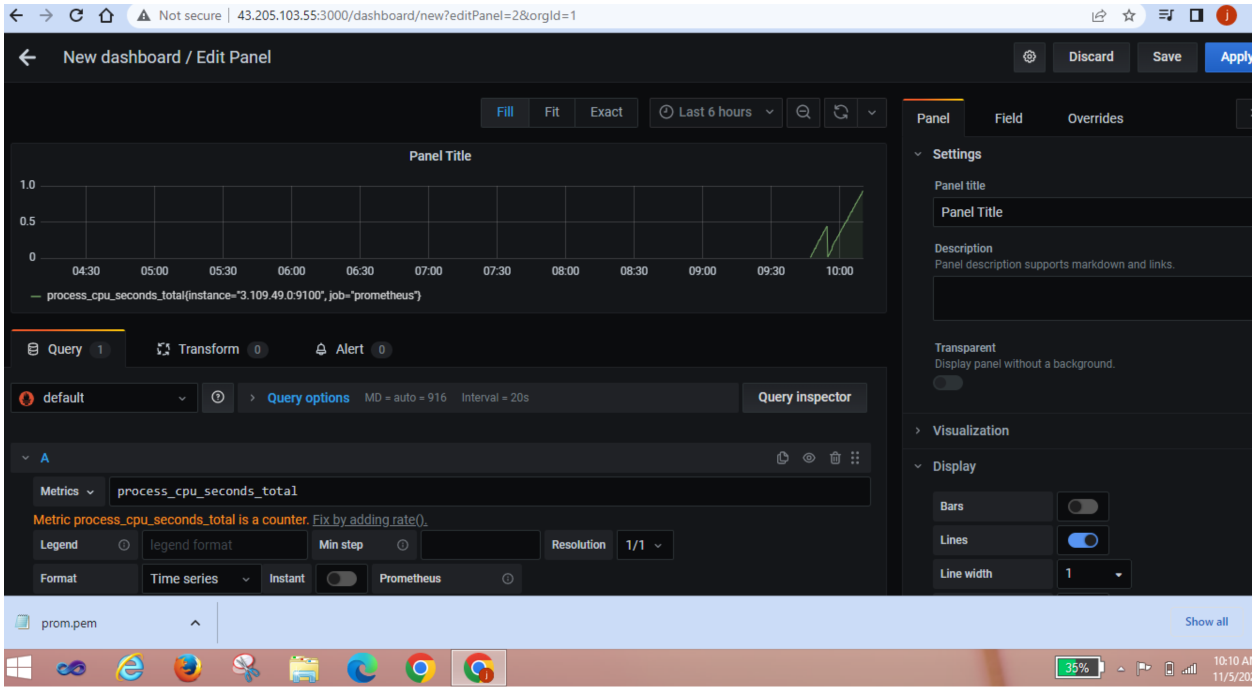


up{instance="3.109.49.0:9100", job="prometheus"} — up{instance="3.110.120.216 :9100", job="prometheus"}

Data source: default
Query options MD = auto = 916 Interval = 20s

A
Metrics: up
Legend: legend format
Min step:
Resolution: 1/1
Format: Time series
Instant: off
Prometheus

Settings
Panel title: Panel Title
Description: Panel description supports markdown and links.
Transparent: Display panel without a background.
Bars: off
Lines: on
Line width: 1



process_cpu_seconds_total{instance="3.109.49.0:9100", job="prometheus"}

Data source: default
Query options MD = auto = 916 Interval = 20s

A
Metrics: process_cpu_seconds_total
Metric process_cpu_seconds_total is a counter. Fix by adding rate().
Legend: legend format
Min step:
Resolution: 1/1
Format: Time series
Instant: off
Prometheus

Settings
Panel title: Panel Title
Description: Panel description supports markdown and links.
Transparent: Display panel without a background.
Bars: off
Lines: on
Line width: 1

3. Alerts and Notifications:
Test Case 3: Create an alerting rule in Prometheus to trigger an alert when a specific metric threshold is breached, and verify that the alerting mechanism works.
Expected Outcome: An alert should be triggered when the specified metric threshold is exceeded, and notifications (e.g., email or Slack) should be received.

Alerts and Notifications:
Test Case 3:

- Define alerting rules in Prometheus by editing the prometheus.yml configuration file. Set alert conditions and rules.
- Test the alerts by triggering a condition breach (e.g., simulate a high load or an error condition).
- Confirm that the alert triggers as expected in Prometheus.
- Configure alert notifications in Prometheus to send alerts to the desired channels (Slack, email, etc.).
- Verify that notifications are received when the alert conditions are met.