

## Docker Monitoring using promethues and Grafana

**1) Create two ubuntu machines – Docker-server ,  
Grafanamonitoring-machine**

**2) In both the machines**

**sudo su**

**apt update -y**

**3) In Grafanamachine I will install Grafana and in  
docker-server I will install docker and Prometheus**

### Grafana machine:

**wget**

<https://dl.grafana.com/enterprise/release/grafana-enterprise-8.4.4.linux-amd64.tar.gz>

**ls**

**tar -zxvf grafana-enterprise-8.4.4.linux-amd64.tar.gz**

**ls**

**cd grafana-8.4.4**

**./bin/grafana-server**

Grafana servers works on port 3000...so in security group enable port 3000

The screenshot shows the AWS EC2 Management Console with the URL <https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#ModifyInboundSecurityGroupRules:securityGroupId=sg-0d9d282c340231536>. The page displays the 'Inbound rules' section for a specific security group. The table has columns for Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There are four existing rules and one new rule being added:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0dcfc3df15646bdd7	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-09eafdf5bfcb0baff	HTTPS	TCP	443	Custom	0.0.0.0/0
sgr-0287f0810acac5bdb	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	3000	Anywh...	0.0.0.0/0

A 'Delete' button is present for each row. At the bottom left is a 'Add rule' button.

<<public ip>>:3000

The screenshot shows a web browser window with the URL <http://18.188.102.31:3000/login>. The page is titled 'Grafana' and displays a 'Welcome to Grafana' message with a logo. It contains two input fields: 'Email or username' and 'Password', both with placeholder text ('email or username' and 'password'). Below the fields are 'Log in' and 'Forgot your password?' buttons. At the bottom of the page are links for Documentation, Support, Community, Enterprise (Free & unlicensed), v8.4.4 (fcfb01fae), and New version available!.

admin

```
pass: admin
```

```
#now lets setup the docker-server
```

```
apt install docker.io -y
```

```
service docker start
```

```
docker run -it --name c01 ubuntu /bin/bash
```

```
docker run -it --name c02 ubuntu /bin/bash
```

```
wget
```

```
https://github.com/prometheus/prometheus/releases/download/v2.34.0/prometheus-2.34.0.linux-amd64.tar.gz
```

```
ls
```

```
tar zxvf prometheus-2.34.0.linux-amd64.tar.gz
```

```
ls
```

```
cd prometheus-2.34.0.linux-amd64
```

```
ls
```

```
##WE NEED TO TELL DOCKER AS WELL THAT PROMETHEUS WOULD BE TRACKING  
DOCKER VIA PORT 9232 port
```

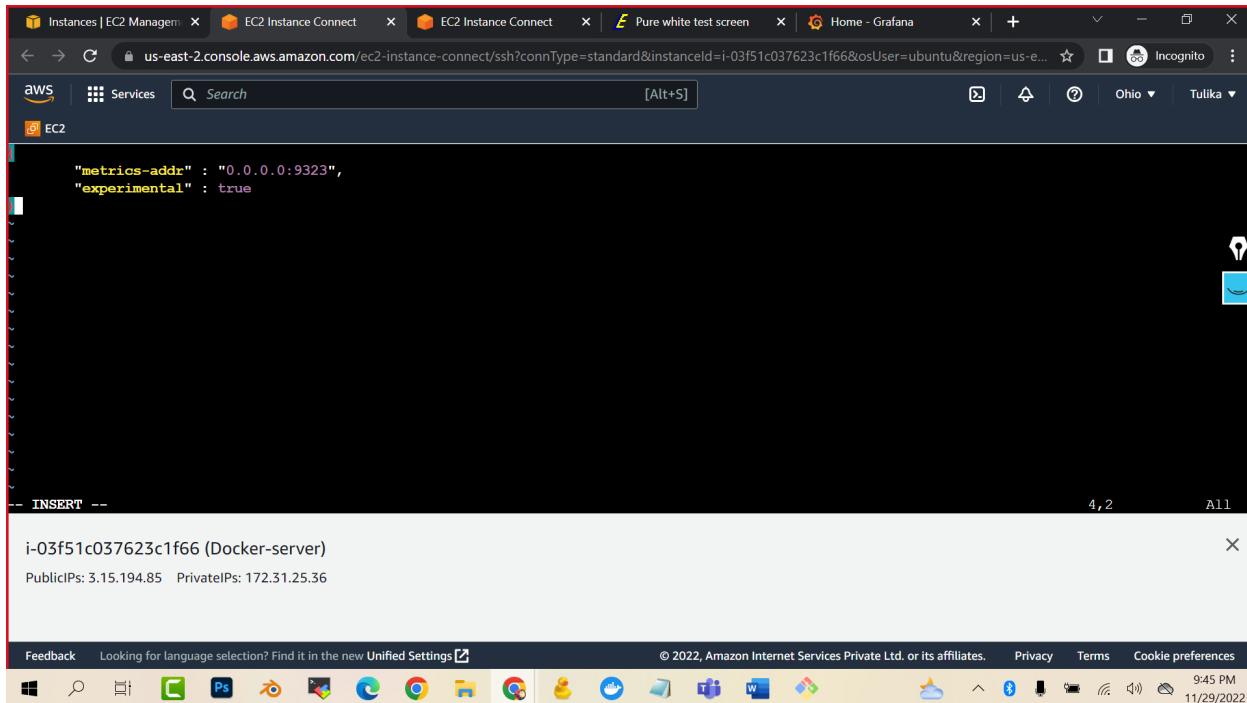
```
vi /etc/docker/daemon.json
```

```
press I to insert
```

```
{
```

```
"metrics-addr" : "0.0.0.0:9323",
"experimental" : true
}
```

To come out of this file press escape and :wq



service docker restart

```
cd prometheus-2.34.0.linux-amd64
```

```
ls
```

(you will see Prometheus.yml file...it contains all the configuration of Prometheus.  
Here we have to tell that metrics would be coming from port 9323)

Before that,

go to docker-server machine and enable the port 9323 and port 9090

The screenshot shows the AWS EC2 Management Console with several tabs open. The main content area displays a list of inbound security group rules for a specific security group (sg-0f0183d8b287ec1fe). The rules are as follows:

- Protocol: HTTPS, Port: 443, Source: 0.0.0.0/0
- Protocol: SSH, Port: 22, Source: 0.0.0.0/0
- Protocol: Custom TCP, Port: 3000, Source: 0.0.0.0/0
- Protocol: HTTP, Port: 80, Source: 0.0.0.0/0
- Protocol: Custom TCP, Port: 9090, Source: Anywhere, Destination: 0.0.0.0/0
- Protocol: Custom TCP, Port: 9323, Source: Anywhere, Destination: 0.0.0.0/0

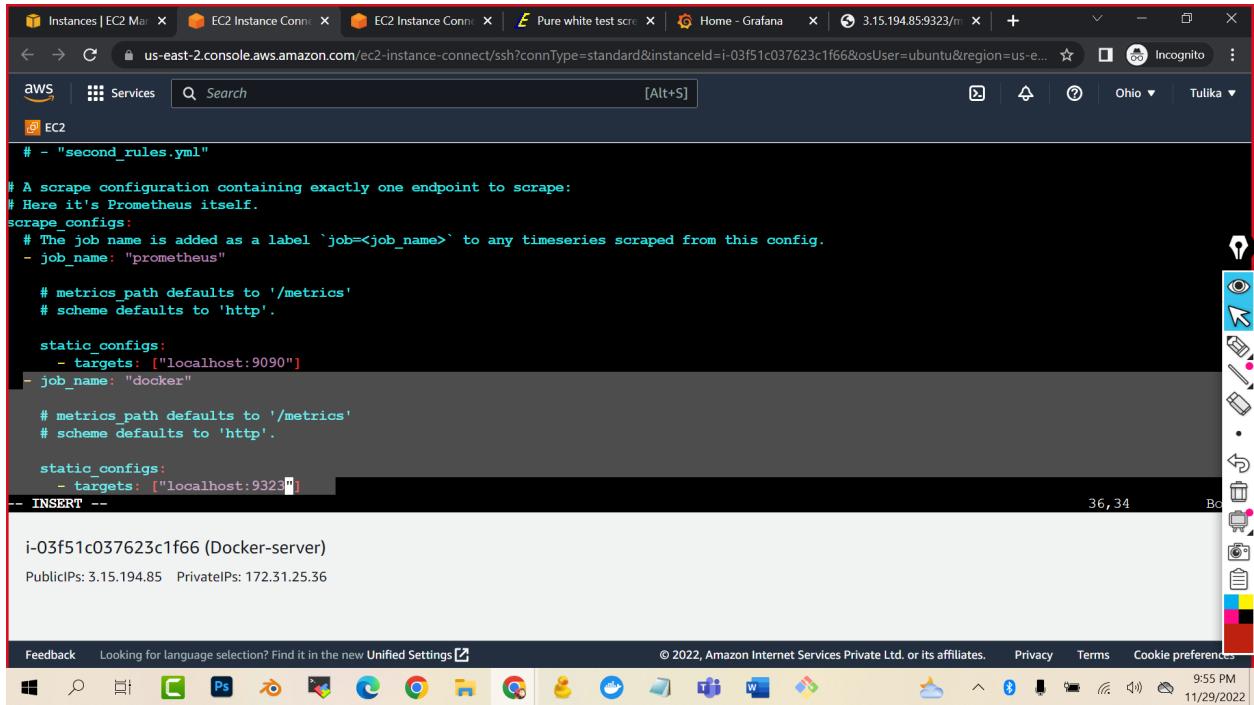
At the bottom left, there is a button labeled "Add rule". The browser's address bar shows the URL: us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#ModifyInboundSecurityGroupRules:securityGroupId=sg-0f0183d8b287ec1fe. The status bar at the bottom right indicates the time as 9:51 PM and the date as 11/29/2022.

Copy the ip of machine and <http://3.15.194.85:9323/metrics>

You should see the docker stats

The screenshot shows a web browser window displaying Docker metrics at the URL <http://3.15.194.85:9323/metrics>. The page content is a large block of text representing a Prometheus metrics endpoint for Docker daemon statistics. The text includes various metric definitions and their values, such as build failed counts, triggered builds, and histogram data for actions like changes and commit. The browser's address bar shows the URL and the status bar at the bottom right indicates the time as 9:53 PM and the date as 11/29/2022.

```
vi prometheus.yml
```



```
# - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "docker"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9323"]
-- INSERT --
i-03f51c037623c1f66 (Docker-server)
PublicIPs: 3.15.194.85 PrivateIPs: 172.31.25.36
```

- job\_name: "docker"

```
# metrics_path defaults to '/metrics'
# scheme defaults to 'http'.
```

```
static_configs:
```

```
  - targets: ["localhost:9323"]
```

(just take care of one thing that – job\_name should be exactly the job\_name of the Prometheus)

```
# - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "docker"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9323"]
-- INSERT --
```

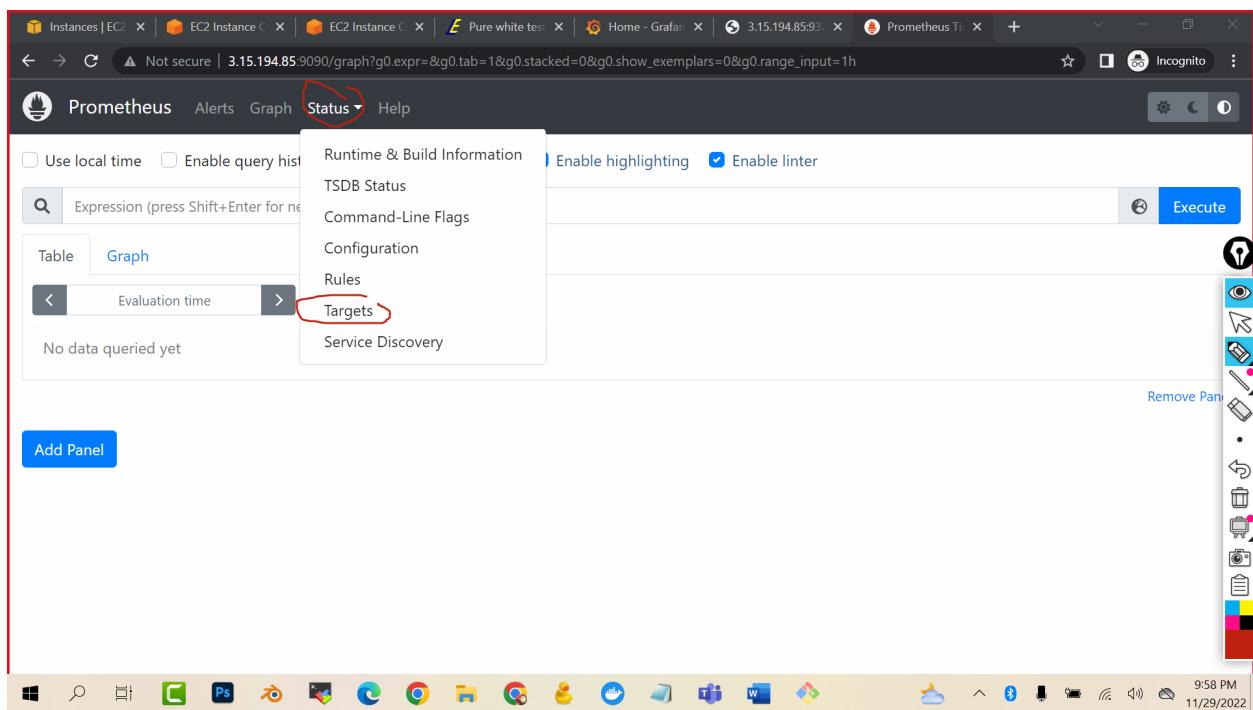
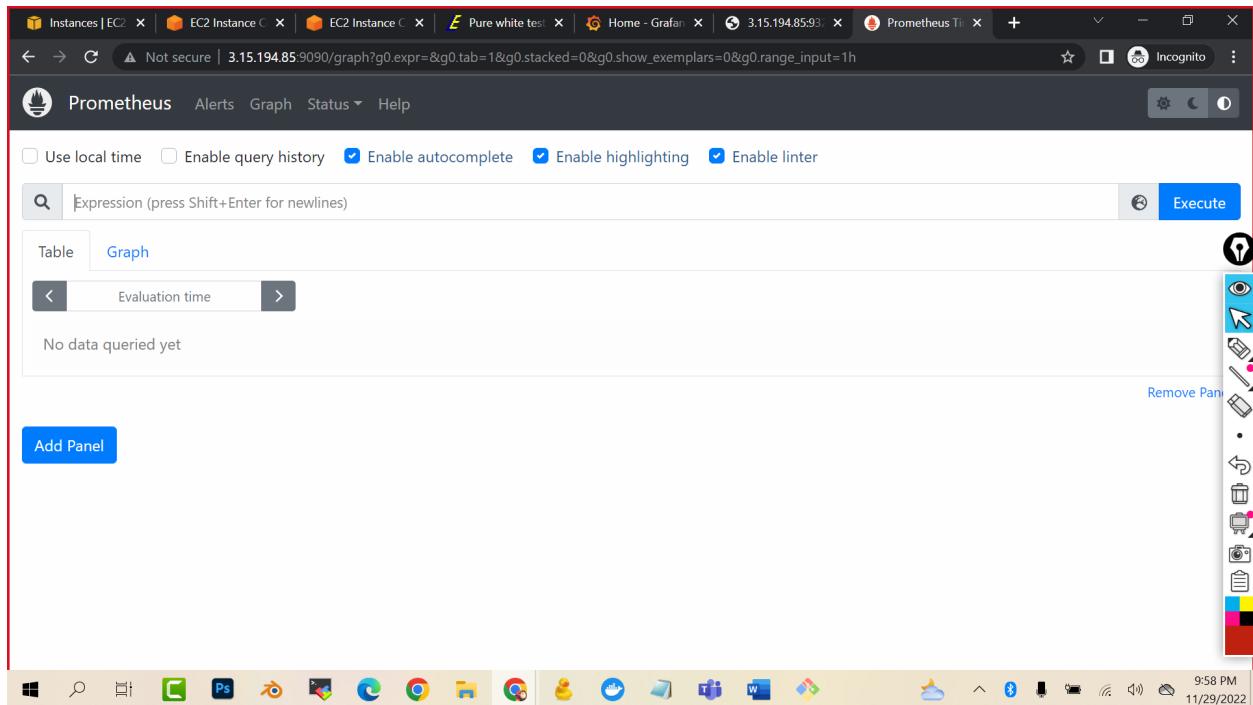
i-03f51c037623c1f66 (Docker-server)  
PublicIPs: 3.15.194.85 PrivateIPs: 172.31.25.36

./prometheus (this will start the prometheus)

```
prometheus-2.34.0.linux-amd64/prometheus
prometheus-2.34.0.linux-amd64/pronotool
root@ip-172-31-25-36:/home/ubuntu# ls
prometheus-2.34.0.linux-amd64  prometheus-2.34.0.linux-amd64.tar.gz
root@ip-172-31-25-36:/home/ubuntu# vi /etc/docker/daemon.json
root@ip-172-31-25-36:/home/ubuntu# ^C
root@ip-172-31-25-36:/home/ubuntu# vi /etc/docker/daemon.json
root@ip-172-31-25-36:/home/ubuntu# service docker restart
root@ip-172-31-25-36:/home/ubuntu# cd prometheus-2.34.0.linux-amd64
root@ip-172-31-25-36:/home/ubuntu/prometheus-2.34.0.linux-amd64# ls
LICENSE NOTICE console_libraries consoles prometheus prometheus.yml promtool
root@ip-172-31-25-36:/home/ubuntu/prometheus-2.34.0.linux-amd64# vi prometheus.yml
root@ip-172-31-25-36:/home/ubuntu/prometheus-2.34.0.linux-amd64# ./prometheus
ts=2022-11-29T16:27:13.638Z caller=main.go:479 level=info msg="No time or size retention was set so using the default time retention" duration=1ms
ts=2022-11-29T16:27:13.638Z caller=main.go:516 level=info msg="Starting Prometheus" version="(version=2.34.0, branch=HEAD, revision=88111fec4333094a6fb2680c71fffc427275)"
ts=2022-11-29T16:27:13.638Z caller=main.go:521 level=info build_context="(go=golang1.17.8, user=root@121ad7ea5487, date=20220315-15:18:00)"
ts=2022-11-29T16:27:13.638Z caller=main.go:522 level=info host_details="(Linux 5.15.0-1019-aws #23-Ubuntu SMP Wed Aug 17 18:33:13 UTC 2022 x86_64 ip-172-31-25-36 (none))"
ts=2022-11-29T16:27:13.639Z caller=main.go:523 level=info fd_limits="(soft=1024, hard=1048576)"
ts=2022-11-29T16:27:13.639Z caller=main.go:524 level=info vm_limits="(soft=unlimited, hard=unlimited)"

i-03f51c037623c1f66 (Docker-server)
PublicIPs: 3.15.194.85 PrivateIPs: 172.31.25.36
```

Now open public ip:9090 to check if Prometheus is opening or not



Targets

All Unhealthy Collapse All

Filter by endpoint or labels

**docker (1/1 up)** show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9323/metrics	UP	instance="localhost:9323" job="docker"	6.846s ago	3.201ms	

**prometheus (1/1 up)** show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	2.914s ago	4.971ms	

3.15.194.85:9090/targets#pool-docker

Now lets go to Grafana and we will give source as Prometheus to Grafana so that it can create beautiful graphs out of it

General / Home

Welcome to Grafana

Need help? Documentation Tutorials Community Public Slack

Remove this panel

Basic

The steps below will guide you to quickly familiarize yourself with your new Grafana instance.

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

DASHBOARDS

Create your first dashboard

18.188.102.31:3000/datasources

## (select data sources)

The screenshot shows the Grafana Configuration interface. The title bar indicates the URL is 18.188.102.31:3000/datasources. The main content area displays a message: "No data sources defined". Below this is a prominent blue button labeled "Add data source" with a white checkmark icon. A yellow arrow points from the text "Select data source" in the previous step to this button. To the right of the main content is a vertical toolbar with various icons for managing data sources. At the bottom of the screen, the Windows taskbar is visible, showing icons for various applications like File Explorer, Microsoft Edge, and the Start button.

The screenshot shows the "Add data source" configuration dialog. The title bar indicates the URL is 18.188.102.31:3000/datasources/new. The main content area is titled "Add data source" and "Choose a data source type". It features a search bar labeled "Filter by name or type" and a "Cancel" button. Below the search bar, there is a section titled "Time series databases" containing four entries: "Prometheus" (selected), "Graphite", "OpenTSDB", and "InfluxDB". Each entry includes a small icon, a name, a description, and a "Core" label. A yellow arrow points from the text "Select data source" in the previous step to the "Prometheus" entry. To the right of the main content is a vertical toolbar with various icons for managing data sources. At the bottom of the screen, the Windows taskbar is visible, showing icons for various applications like File Explorer, Microsoft Edge, and the Start button.

Instances | EC2 | EC2 Instance | EC2 Instance | Pure white test | Prometheus: S | 3.15.194.85:93 | Prometheus Test | + | Not secure | 18.188.102.31:3000/datasources/edit/p5i9uMKVk | Incognito

Data Sources / Prometheus  
Type: Prometheus

Settings Dashboards

Name: Prometheus Default:

HTTP

URL: http://localhost:9090 Access: Server (default) Help

Allowed cookies: New tag (enter key to add)

Timeout: Timeout in seconds

Auth

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

10:00 PM 11/29/2022

<http://<ip address>:9090/>

Instances | EC2 | EC2 Instance | EC2 Instance | Pure white test | Prometheus: S | 3.15.194.85:93 | Prometheus Test | + | Not secure | 18.188.102.31:3000/datasources/edit/p5i9uMKVk | Incognito

Data Sources / Prometheus  
Type: Prometheus

Settings Dashboards

Name: Prometheus Default:

HTTP

URL: http://3.15.194.85:9090/ Access: Server (default) Help

Allowed cookies: New tag (enter key to add)

Timeout: Timeout in seconds

Auth

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

10:02 PM 11/29/2022

The screenshot shows the Prometheus UI for editing a data source. The URL is <http://18.188.102.31:3000/datasources/edit/p5i9uMKVk>. The page displays configuration for an Alertmanager data source, including:

- Scrape interval: 15s
- Query timeout: 60s
- HTTP Method: POST

In the Misc section, there is a toggle for "Disable metrics lookup" and a field for "Custom query parameters" with an example: `max_source_resolution=5m&timeout=10`.

The Exemplars section contains a "+ Add" button.

At the bottom, there are three buttons: "Back", "Explore", and "Delete" (highlighted with a yellow circle), followed by a large blue "Save & test" button.

The status bar at the bottom right shows the time as 10:01 PM and the date as 11/29/2022.

The screenshot shows the same Prometheus UI page after saving and testing the configuration. The "Save & test" button has been clicked, and the result is displayed below it:

**Data source is working**

The status bar at the bottom right shows the time as 10:02 PM and the date as 11/29/2022.

The screenshot shows the Grafana home page. On the left, there's a sidebar with various icons. The main area has a dark header with tabs like "Instances | EC2", "EC2 Instance C", "EC2 Instance C", "Pure white test", "Home - Grafana", "3.15.194.85:93", and "Prometheus Test". Below the header, it says "Welcome to Grafana" and "Need help? Documentation Tutorials Community Public Slack".

**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.

**COMPLETE**

Add your first data source

**DASHBOARDS**

Create your first dashboard

Learn how in the docs [Learn how in the docs](#)

Learn how in the docs [Learn how in the docs](#)

A yellow circle highlights the "Create your first dashboard" link in the "DASHBOARDS" section.

Below the main content, there are sections for "Dashboards" and "Latest from the blog". The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 10:03 PM on 11/29/2022.

The screenshot shows the "New dashboard" creation interface. The top bar includes tabs for "Instances | EC2", "EC2 Instance C", "EC2 Instance C", "Pure white test", "New dashboard", "3.15.194.85:93", and "Prometheus Test". The main area has a dark header with tabs like "Last 6 hours", "Search", and "Edit".

**New dashboard**

**Add panel**

Add a new panel

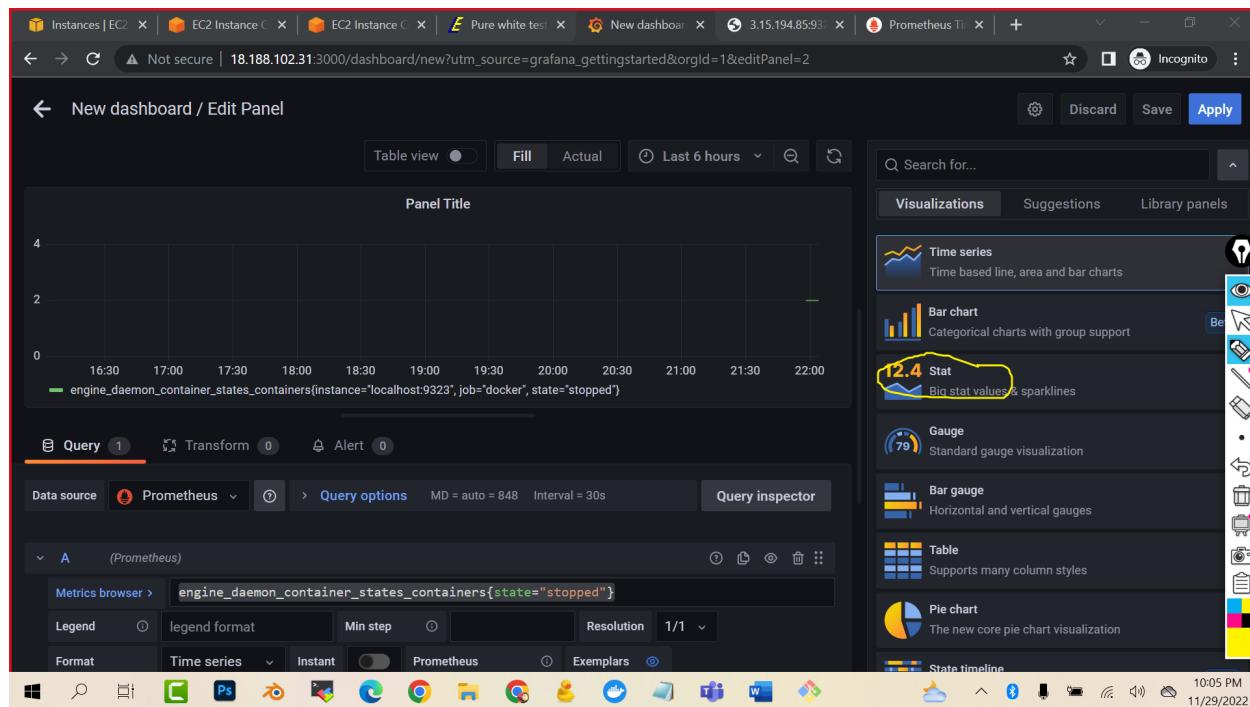
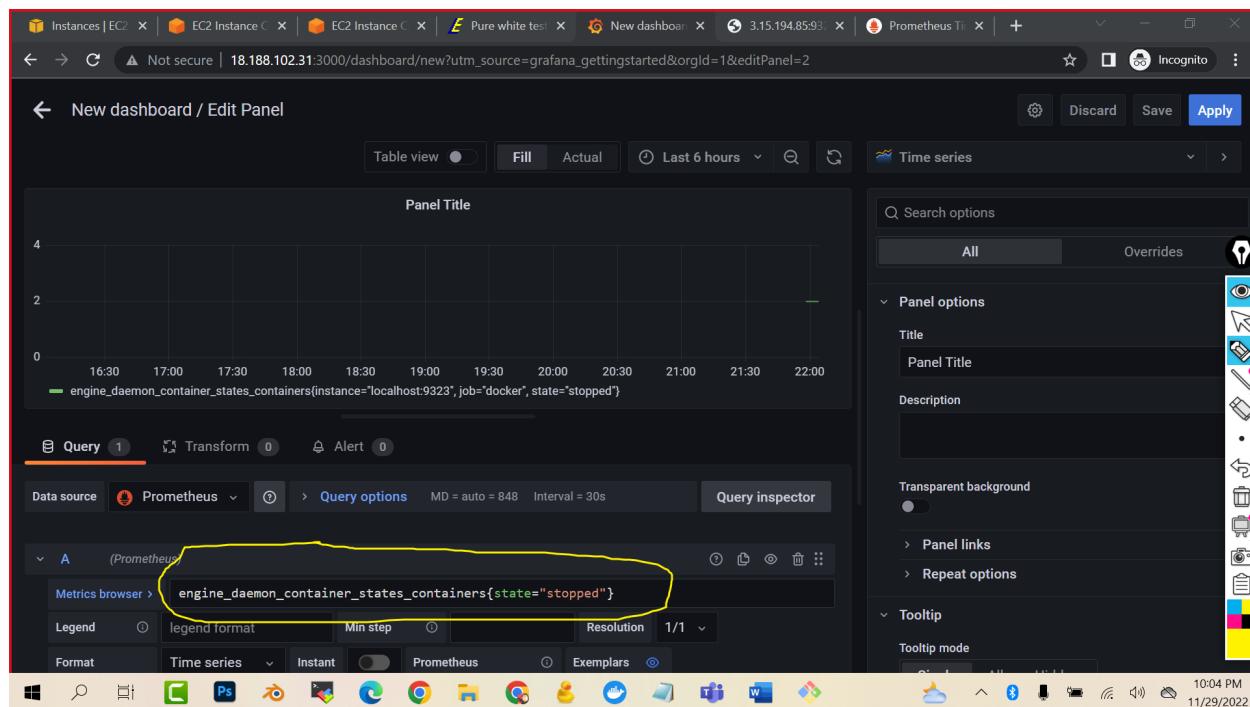
Add a new row

Add a panel from the panel library

A yellow circle highlights the "Add a new panel" button in the "Add panel" section.

The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 10:03 PM on 11/29/2022.

```
engine_daemon_container_states_containers{state="stopped"}
```



```
engine_daemon_container_states_containers{state="running"}
```

