

SERVER MONITORING USING PROMETHEUS AND GRAFANA


Project By:-

SAURABH FEGADE (D17C - 14)


RAJAT GUPTA (D17C - 16)

JAYA HINDUJA (D17C - 19)

1. Abstract

- Server Monitoring is a process to monitor server's system resources like CPU Usage, Memory Consumption, I/O, Network, Disk Usage, Process etc.
 - Server Monitoring helps understanding server's system resource usage which can help you better your capacity planning and provide a better end-user experience.
 - Server Monitoring provides you with data relating to your operating system and when used in conjunction with other monitoring data from the application you get a true glimpse into the working of your system.
- 

2. Introduction

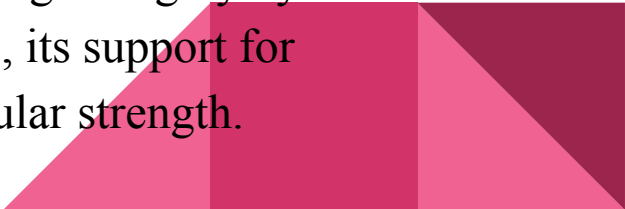
- Prometheus is an open source monitoring system and time series database. It addresses many aspects of monitoring such as the generation and collection of metrics, graphing the resulting data on dashboards, and alerting on anomalies.
 - It offers a variety of components that can run separately but are used in combination.
 - Docker provides a way for you to encapsulate server processes using Linux containers (or other encapsulation technologies) so that they are more easily managed and isolated from each other. Each Docker container is a local instance of a Docker image.
 - In this project we will show you how to install three key components for using Prometheus on Docker.
- 


3. Components

- Three Key Components are:
 - A Prometheus server to collect metrics and query them.
 - A Node Exporter to export system metrics in a Prometheus-compatible format.
 - Grafana, a web-based graphical dashboard builder that supports Prometheus among other backends.
- There are many more components in the Prometheus ecosystem, but these three provide a good starting point for using Prometheus.



Prometheus

- Prometheus scrapes metrics from instrumented jobs, either directly or via an intermediary push gateway for short-lived jobs. It stores all scraped samples locally and runs rules over this data to either record new time series from existing data or generate alerts. Grafana or other API consumers can be used to visualize the collected data.
 - Prometheus works well for recording any purely numeric time series.
 - It fits both machine-centric monitoring as well as monitoring of highly dynamic service-oriented architectures. In a world of microservices, its support for multi-dimensional data collection and querying is a particular strength.
- 

- Prometheus is designed for reliability, to be the system you go to during an outage to allow you to quickly diagnose problems.
 - Each Prometheus server is standalone, not depending on network storage or other remote services.
 - Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored.
 - Create, explore, and share dashboards with your team and foster a data driven culture.
- 

4. Implementation

Step 1:- We need the following:

- an Ubuntu 16.04 droplet on DigitalOcean;
- a non-root user with sudo access; and
- Docker installed on the server droplet.

Step 2:- Initial Server Setup. Creating the non-root user and giving it sudo access.

Step 3:- Installing Docker



Step 4:- Installing Prometheus

```
$ nano ~/prometheus.yml
```

Add the following contents to the file:

```
# A scrape configuration scraping a Node Exporter and the Prometheus server  
# itself.
```

```
scrape_configs:
```

```
  # Scrape Prometheus itself every 5 seconds.
```

```
  - job_name: 'prometheus'
```

```
    scrape_interval: 5s
```

```
    static_configs:
```

```
      - targets: ['localhost:9090']
```

```
  # Scrape the Node Exporter every 5 seconds.
```

```
  - job_name: 'node'
```

```
    scrape_interval: 5s
```

```
    static_configs:
```

```
      - targets: [45.55.147.132:9100']
```



Start the Prometheus Docker container with the external configuration file:

```
$ sudo docker run -d -p 9090:9090 -v
```

Step 5: Setting up Node Exporter

To start the Node Exporter on port 9100 using Docker:

```
$ sudo docker run -d -p 9100:9100 -v
```

Step 6: Setting up Grafana

```
$ sudo docker run -d -p 3000:3000 -e  
"GF_SECURITY_ADMIN_PASSWORD=admin" -v  
~/grafana_db:/var/lib/grafana grafana/grafana
```

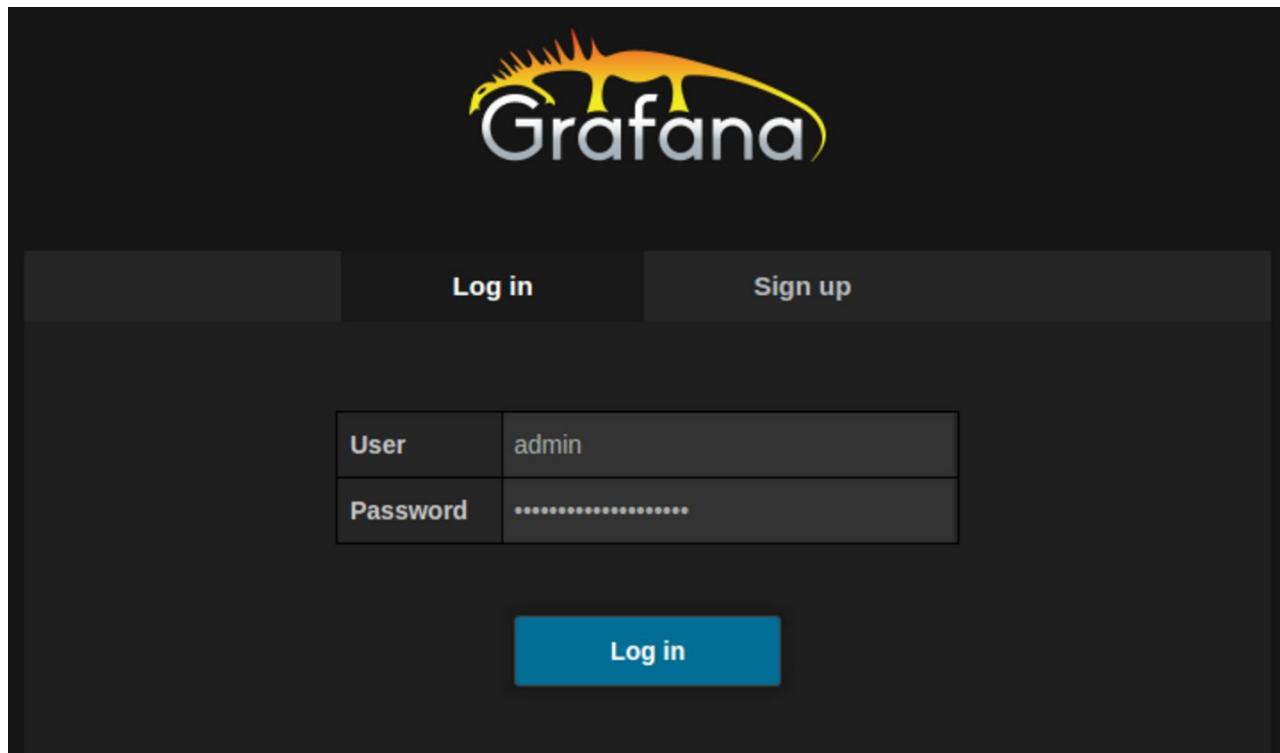
Step 7: Checking System Status

One can check all running docker containers using the following command:

```
$ sudo docker ps
```



5. Screenshots



The screenshot shows the Grafana login interface. At the top center is the Grafana logo, which consists of a yellow and orange dinosaur-like creature above the word "Grafana" in a white sans-serif font. Below the logo is a horizontal bar with three buttons: a dark grey button on the left, a "Log in" button in the center, and a "Sign up" button on the right. Underneath this bar is a login form with two rows. The first row has a label "User" and a text input field containing the text "admin". The second row has a label "Password" and a text input field filled with 12 dots. Below the form is a large teal button with the text "Log in".

Grafana

Log in Sign up

User	admin
Password

Log in



Data Sources

Edit data source

Config

Dashboards

Name

Prometheus



Default



Type

Prometheus



Http settings

Url

http://45.55.147.132:9090



Access

direct



Http Auth

Basic Auth

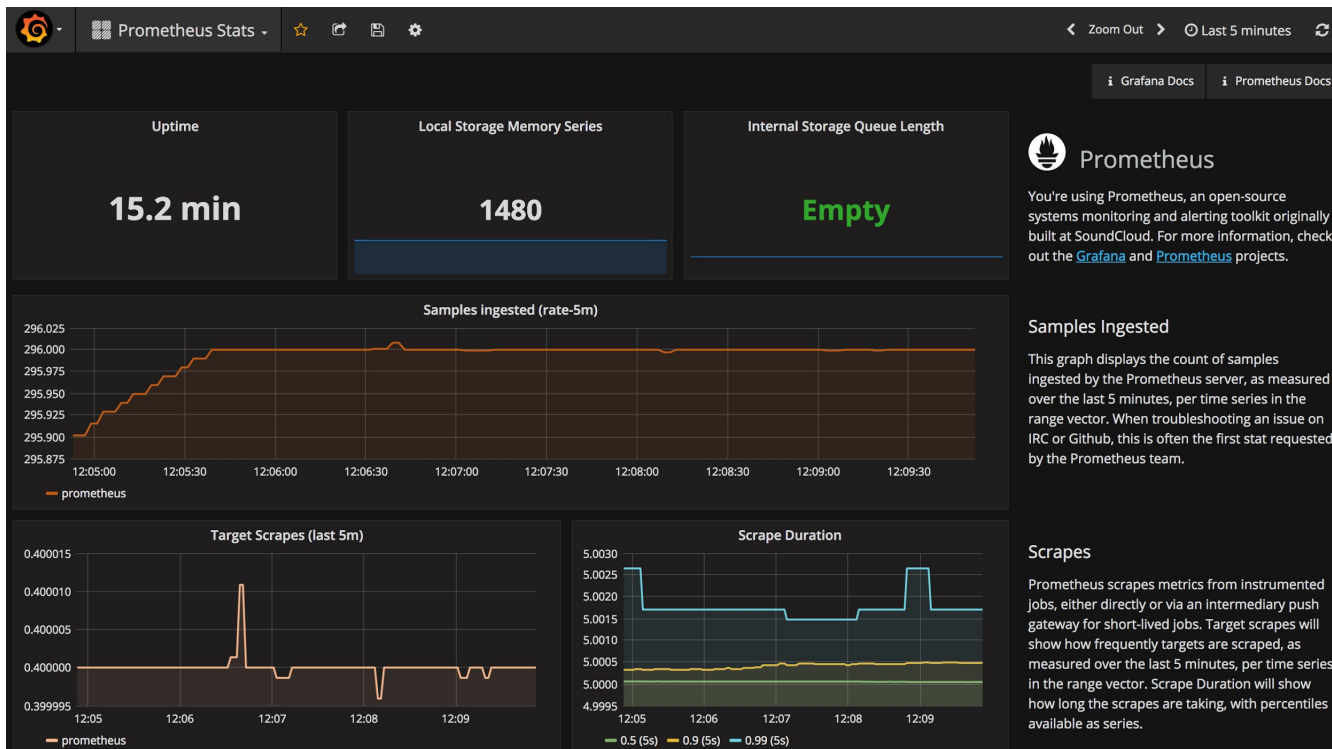


With Credentials



Success

Data source is working

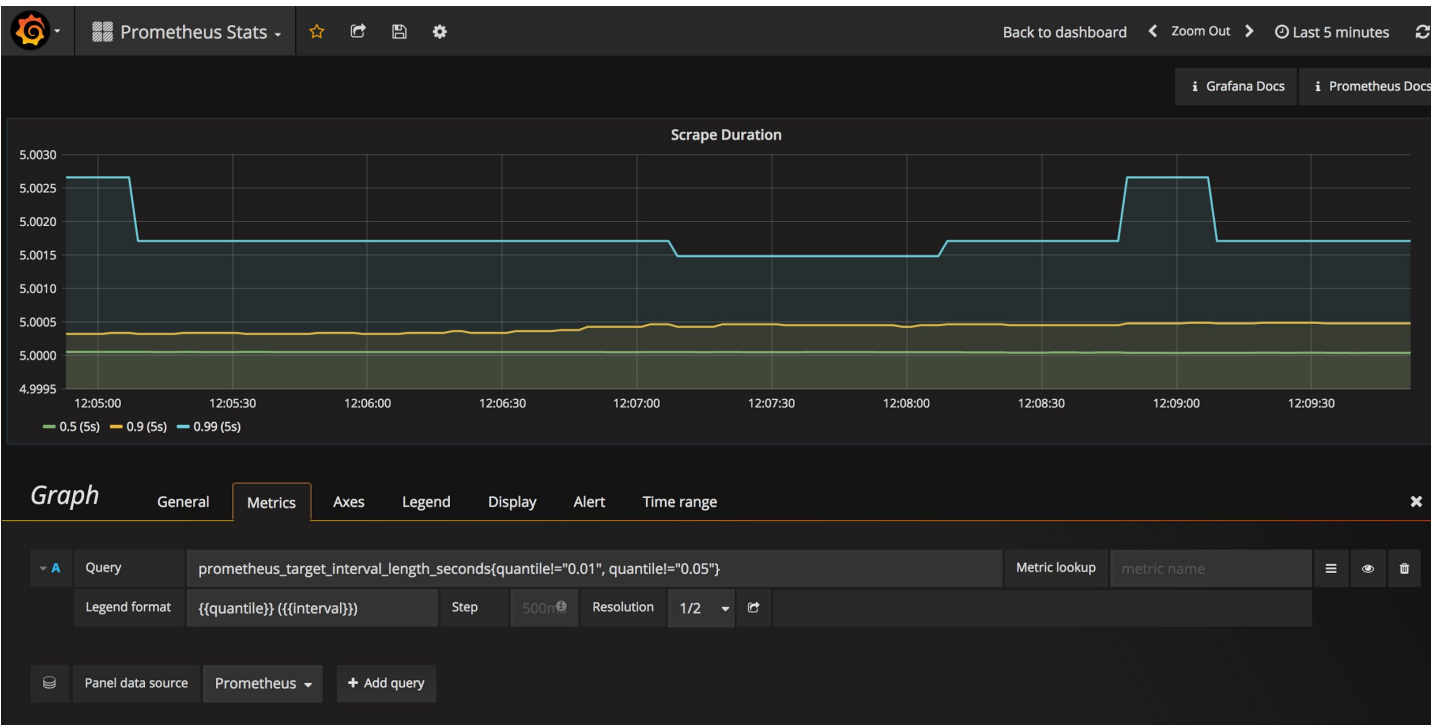


Samples Ingested


This graph displays the count of samples ingested by the Prometheus server, as measured over the last 5 minutes, per time series in the range vector. When troubleshooting an issue on IRC or Github, this is often the first stat requested by the Prometheus team.

Scrapes

Prometheus scrapes metrics from instrumented jobs, either directly or via an intermediary push gateway for short-lived jobs. Target scrapes will show how frequently targets are scraped, as measured over the last 5 minutes, per time series in the range vector. Scrape Duration will show how long the scrapes are taking, with percentiles available as series.



6. References

1. <https://www.digitalocean.com/community/tutorials/how-to-install-prometheus-using-docker-on-ubuntu-14-04>
 2. <https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04>
 3. <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-14-04>
 4. <https://prometheus.io/>
 5. <https://grafana.com/>
- 



Thank You