

Manual técnico sobre la instalación y configuración de Prometheus y Grafana

Por el grupo 5

Integrantes

201700556 -Juan Carlos Juarez Barneond

201602744 - David Omar Enriquez Reyes

201503608 - Juan Luis Robles Molina

Instalación y configuración Grafana:

Instalación de la versión OSS:

Instalamos los siguientes requerimientos

```
sudo apt-get install -y apt-transport-https
sudo apt-get install -y software-properties-common wget
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

Agregamos la dirección del estable para obtener actualizaciones

```
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list
```

Actualizamos e instalamos Grafana

```
sudo apt-get update
sudo apt-get install grafana
```

Agregar el servicio y verificamos que se haya iniciado

```
sudo systemctl daemon-reload
sudo systemctl start grafana-server
sudo systemctl status grafana-server
```

Le indicamos que se inicie cuando inicie la computadora

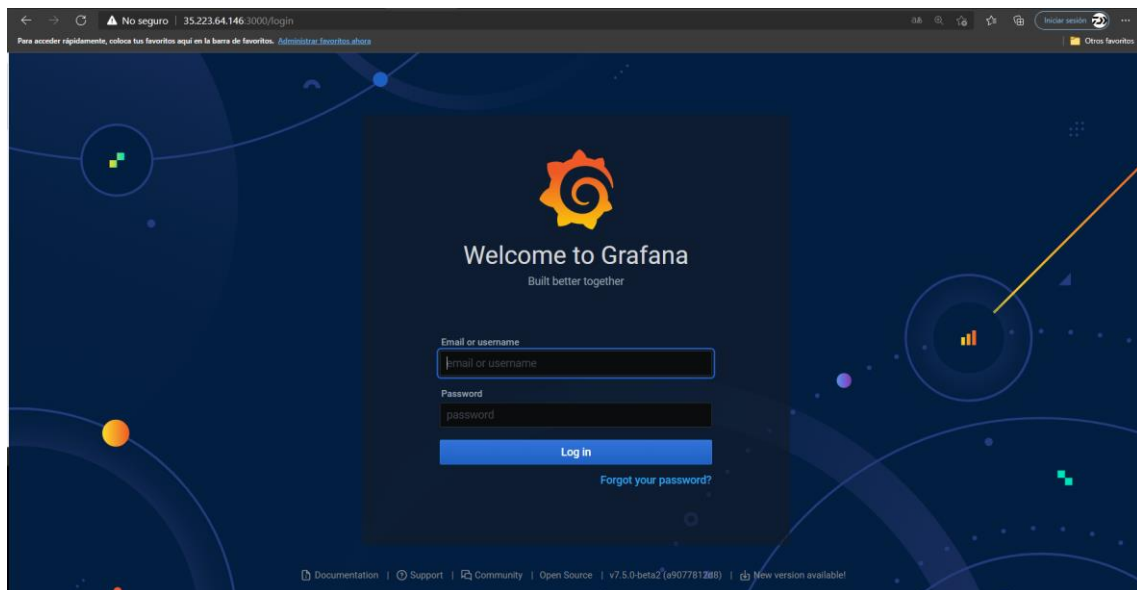
```
sudo systemctl enable grafana-server.service
```

Iniciamos el servicio

```
sudo service grafana-server start
sudo service grafana-server status
```

y si sale “activo” de estado, entonces podemos acceder a la interfaz grafica en el IP:3000

Ahora entras a la interfaz



La primera vez el usuario y password es “admin” y “admin”.

Luego te topas con un escritorio vacío, para utilizar las gráficas necesitamos información, aquí es donde entra prometheus

Configuración de Prometheus

Crearemos un par de usuarios, para mantener aislada la interacción del usuario que utilizamos, es decir, ni el usuario prometheus podrá acceder a node_exporter, ni node_exporter a prometheus, ni ningún otro a esos servicios, pero los servicios si podrán acceder entre si.

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

creamos las carpetas que van a contener lo que vamos a utilizar

- `sudo mkdir /etc/prometheus`
- `sudo mkdir /var/lib/prometheus`

Le damos permiso a cada usuario sobre su carpeta

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

Descargamos Prometheus, tener cuidado con la versión que elegimos, las versiones están disponibles en su página principal, la obtenemos dando clic derecho y copiar enlace, en el proyecto se utilizó la versión 2.26 de prometheus

- `cd ~`
- `curl -LO https://github.com/prometheus/prometheus/releases/download/v2.0.0/prometheus-2.0.0.linux-amd64.tar.gz`

Extraemos el contenido

- `tar xvf prometheus-2.0.0.linux-amd64.tar.gz`

Copiamos el contenido a las carpetas contenedoras que creamos antes

- `sudo cp prometheus-2.0.0.linux-amd64/prometheus /usr/local/bin/`
- `sudo cp prometheus-2.0.0.linux-amd64/promtool /usr/local/bin/`

le agregamos permisos al usuario prometheus sobre los nuevos archivos de las carpetas que ya habíamos creado

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

Copiamos el resto del contenido del tar

- `sudo cp -r prometheus-2.0.0.linux-amd64/consales /etc/prometheus`
- `sudo cp -r prometheus-2.0.0.linux-amd64/console_libraries /etc/prometheus`

Damos permisos de nuevo

- `sudo chown -R prometheus:prometheus /etc/prometheus/consales`
- `sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries`

Configuraremos prometheus, aquí usamos el editor de texto nano

- `sudo nano /etc/prometheus/prometheus.yml`

Agregaremos la siguiente configuración:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9090']
```

Manejamos 2 scrape Interval, el global que aplicara para actualizar todas las configuraciones y en este caso especificamos que del servicio de scrape_info en especifico queremos cada 5 segundos puesto que transmite información importante.

Agregamos de dueño a prometheus

- `sudo chown prometheus:prometheus /etc/prometheus/prometheus.yml`

Con esto comprobamos si funciona

- `sudo -u prometheus /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`

Ahora usaremos otro archivo para especificar la exportación de nuestros datos.

Cuidado con el indentado.

```
[Unit]
Description=Prometheus
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

Ahora reiniciamos los servicios

- `sudo systemctl daemon-reload`
- `sudo systemctl start prometheus`
- `sudo systemctl status prometheus`

Si todo esta bien deberá salir algo así

```
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; disabled;
 vendor preset: enabled)
   Active: active (running) since Fri 2017-07-21 11:40:40 UTC; 3s ago
 Main PID: 2104 (prometheus)
    Tasks: 7
   Memory: 13.8M
      CPU: 470ms
   CGroup: /system.slice/prometheus.service
...

```

Habilitamos prometheus para que sea un servicio al iniciar

- `sudo systemctl enable prometheus`

Configurando Node Exporter

En conjunto con prometheus nos permite obtener información del hardware del servidor

Nos movemos, descargamos

- `cd ~`
- `curl -LO https://github.com/prometheus/node_exporter/releases/download/v0.15.1/node_exporter-0.15.1.linux-amd64.tar.gz`

extraemos

- `tar xvf node_exporter-0.15.1.linux-amd64.tar.gz`

Haremos lo mismo que con prometheus

- `sudo cp node_exporter-0.15.1.linux-amd64/node_exporter /usr/local/bin`
- `sudo chown node_exporter:node_exporter /usr/local/bin/node_exporter`

Vamos a configurar la exportación de node exporter

- `sudo nano /etc/systemd/system/node_exporter.service`

agregamos la siguiente configuración

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target
```

Podemos especificar que servicios queremos de forma especifica en la línea de ExecStart de la siguiente forma

```
ExecStart=/usr/local/bin/node_exporter --collectors.enabled
meminfo,loadavg,filesystem
```

Esto cambio en la version 0.16 en adelante, ahora es `--collectors.meminfo --collectors.loadavg`

Guardamos, reiniciamos

- `sudo systemctl daemon-reload`
- `sudo systemctl start node_exporter`
- `sudo systemctl status node_exporter`

Que inicie cuando iniciamos el servidor

- `sudo systemctl enable node_exporter`

Configuramos un poco el prometheus.yml

- `sudo nano /etc/prometheus/prometheus.yml`

Ahora quedaría así

```
global:
  scrape_interval: 15s

scrape_configs:
```

```

- job_name: 'prometheus'
  scrape_interval: 5s
  static_configs:
    - targets: ['localhost:9090']
- job_name: 'node_exporter'
  scrape_interval: 5s
  static_configs:
    - targets: ['localhost:9100']

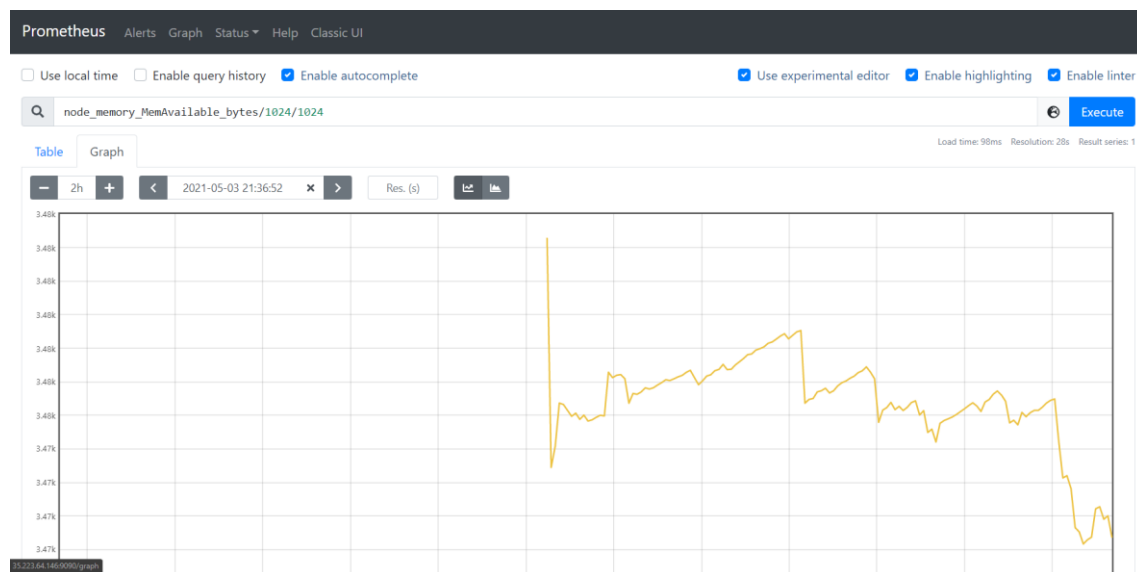
```

Reiniciamos Prometheus

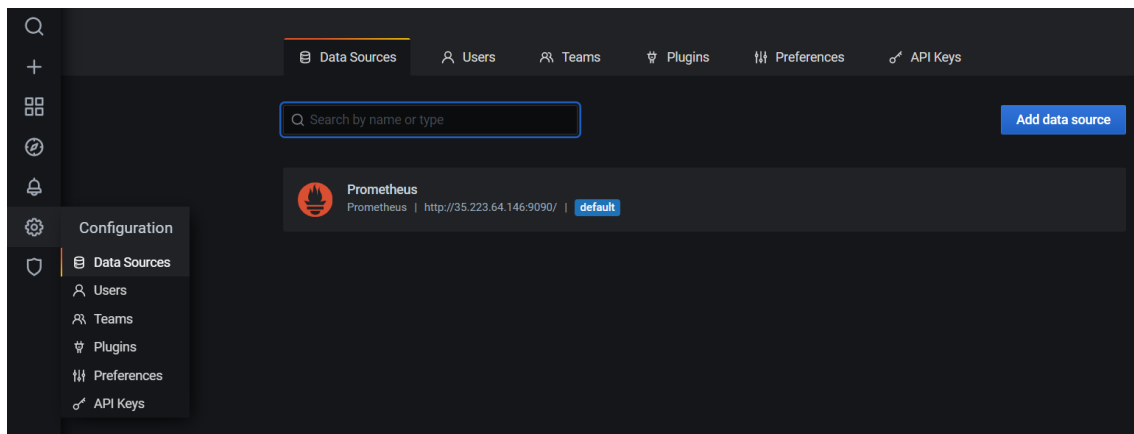
- `sudo systemctl restart prometheus`
- `sudo systemctl status prometheus`

ahora, podemos entrar a la consola de pruebas de prometheus en el puerto que hayamos especificado o por defecto en el :9090

En la consola podremos hacer pruebas y guiarnos para facilitarnos el preview de las consultas y así poder definir las en grafana, lo que sucede es que allá toma más tiempo ver el resultado y no contamos con el autocompletado de las consultas

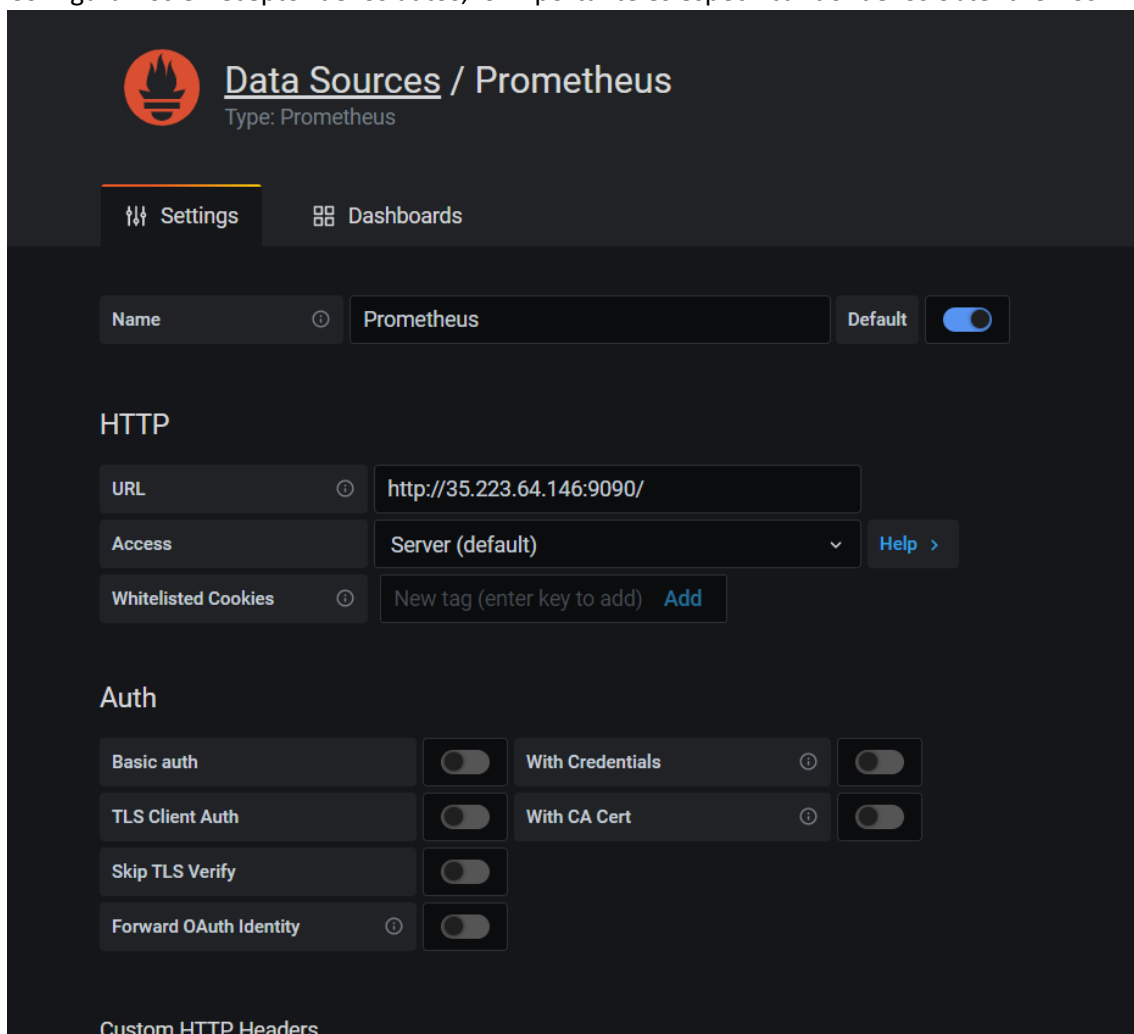


Ahora podemos dirigirnos a la ip en el Puerto 3000 para agregar prometheus como fuente de datos en grafana



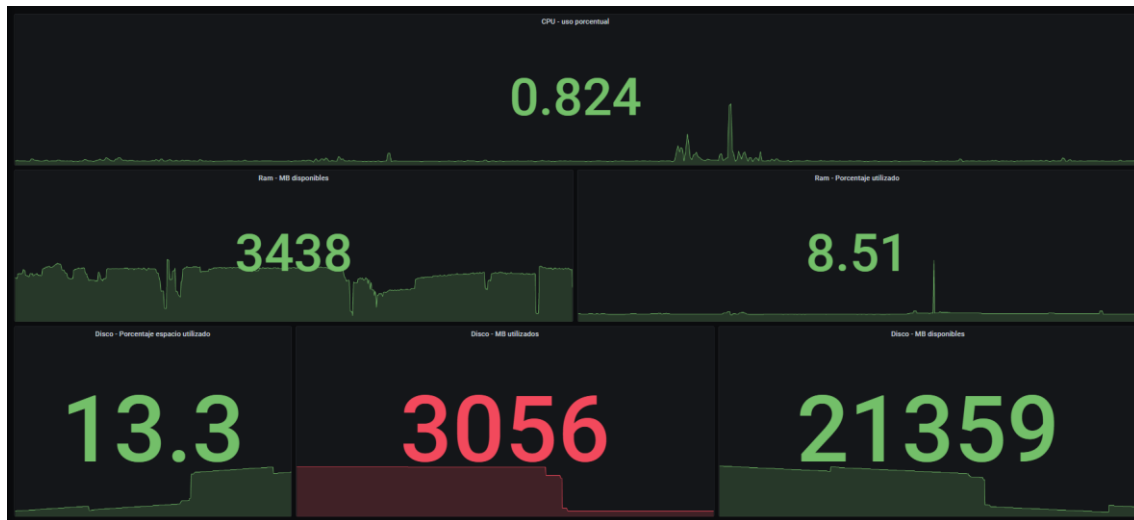
Vamos a data sources y configuramos prometheus

Configuramos el receptor de los datos, lo importante es especificar donde los obtendremos



Ahora, puedes formar tu dashboard poco a poco, es bastante sencillo e intuitivo

Y nos quedo así




O pueden dejar que los profesionales hagan su trabajo e importar una ya propuesta por usuarios más experimentados de las que están publicadas en la página de grafana

[Node Exporter Full dashboard for Grafana | Grafana Labs](#)

Copiamos el número que se indica aquí

Grafana Labs Grafana Products Open Source Learn Downloads [Contact us](#) Login

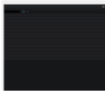

All dashboards » [Node Exporter Full](#)



Node Exporter Full by rtraile
DASHBOARD
Prometheus node exporter sample dashboard
Last updated: 3 months ago
[Start with Grafana Cloud and the new FREE tier. Includes 10K series Prometheus or Graphite Metrics and 50gb Loki Logs](#)

Downloads: 7984306
Reviews: 42
★★★★★
[Add your review!](#)

Overview Revisions Reviews



Nearly all default values exported by Prometheus node exporter graphed.
Only requires the default job_name: node, add as many targets as you need in '/etc/prometheus/prometheus.yml'.

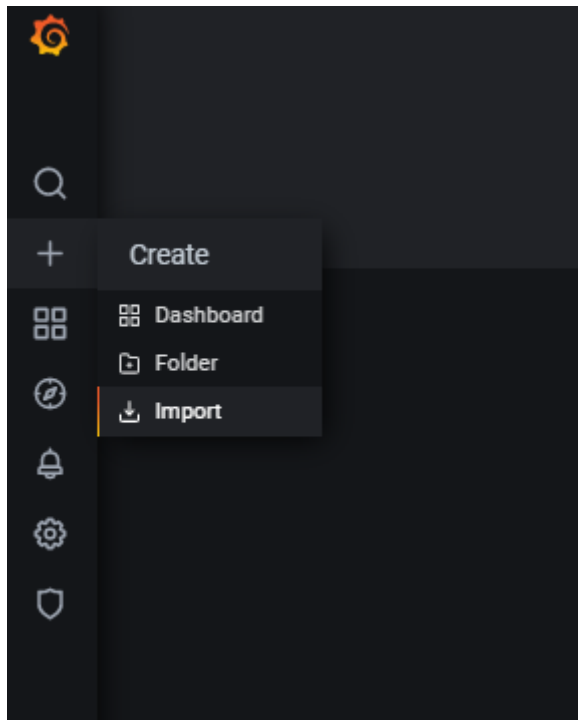
Get this dashboard:

1860


[Copy ID to Clipboard](#)

[Download JSON](#)
[How do I import this dashboard?](#)

Vamos a importar



Agregamos el numero que copiamos



Import

Import dashboard from file or Grafana.com

Upload JSON file

Import via grafana.com

1860

Load

Import via panel json

Load

Y le damos en Load, luego guardamos y ya tenemos un tema importado.

