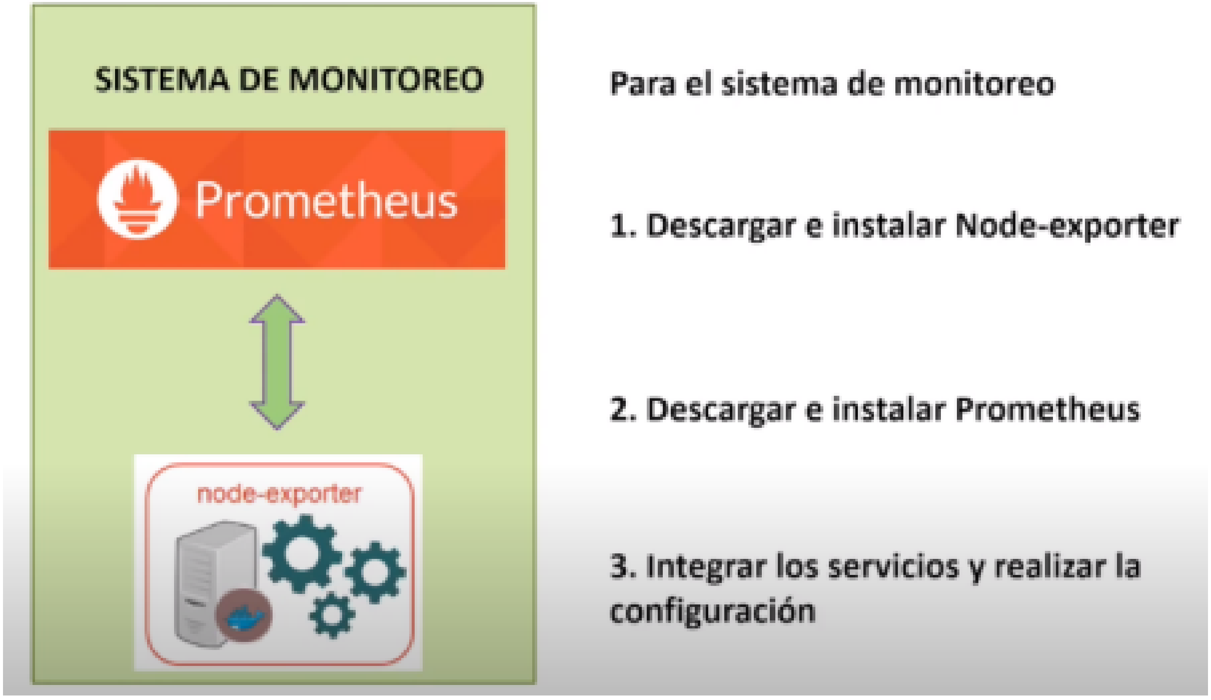
Módulo de Herramientas de monitoreo

Práctico 2

Vamos a instalar la suite de prometheus con node-exporter para obtener las métricas y Grafana para visualizarlas

Instalación del sistema de monitoreo



Para el sistema de monitoreo

1. Descargar e instalar Node-exporter
2. Descargar e instalar Prometheus
3. Integrar los servicios y realizar la configuración

Instalación de node-exporter

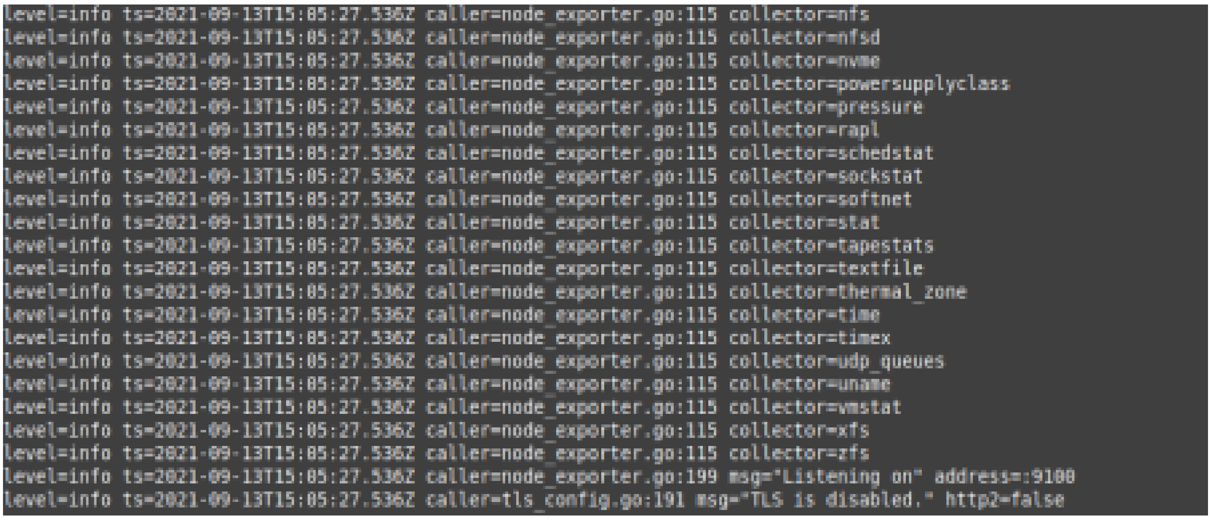
vamos a https://prometheus.io/download/

Lo descargamos

```
> wget
https://github.com/prometheus/node_exporter/releases/download/v1.2.2/node_e
xporter-1.2. 2.linux-amd64.tar.gz
```

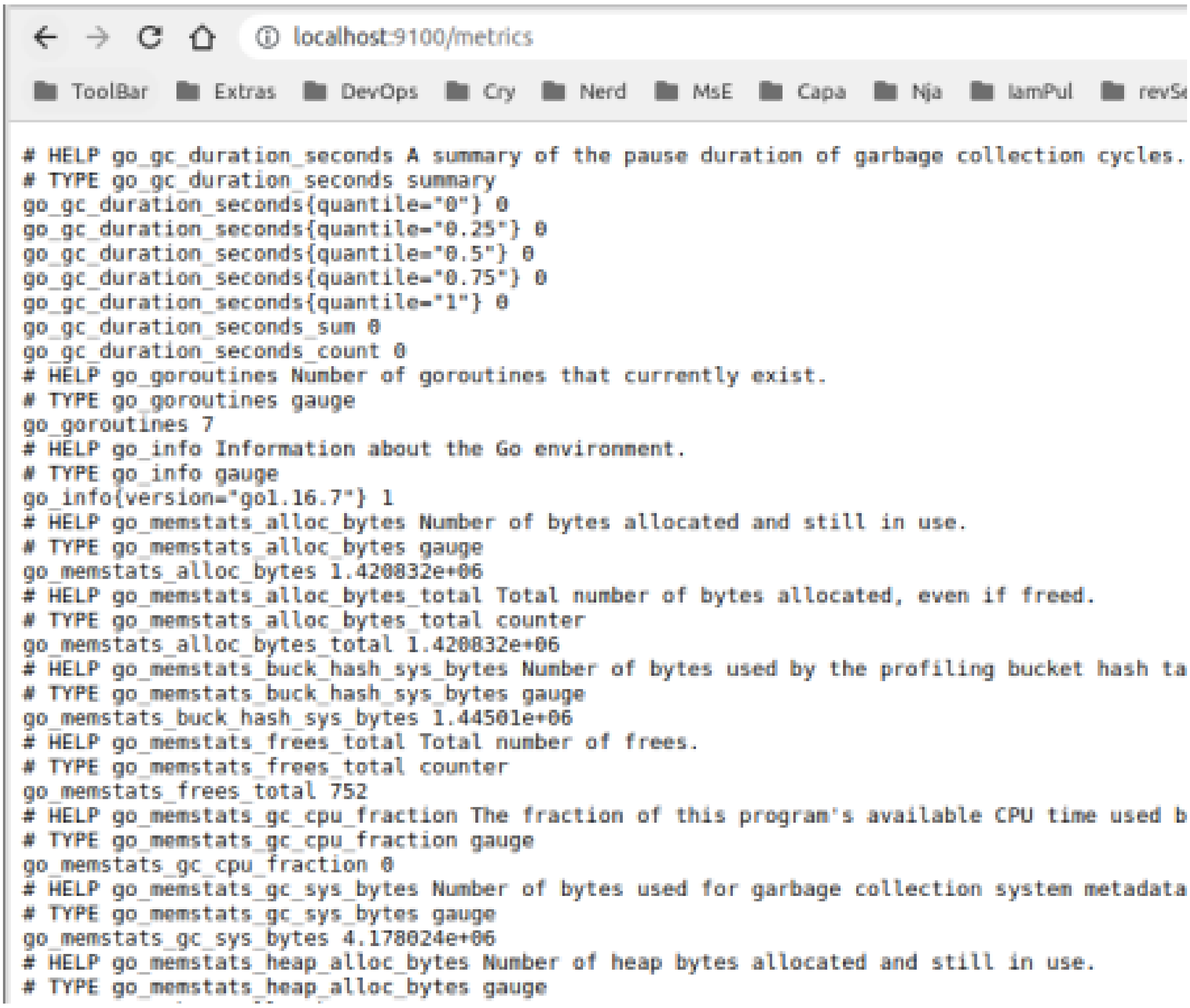
-levantamos el servicio

```
> cd node_exporter-1.2.2.linux-amd64
./node_exporter
```



```
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=nfs
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=nfsd
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=nvme
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=powersupplyclass
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=pressure
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=rapl
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=schedstat
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=sockstat
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=softnet
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=stat
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=tapestats
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=textfile
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=thermal_zone
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=time
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=timex
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=udp_queues
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=uname
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=vmstat
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=xfs
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:115 collector=zfs
level=info ts=2021-09-13T15:05:27.536Z caller=node_exporter.go:199 msg="Listening on" address=:9100
level=info ts=2021-09-13T15:05:27.536Z caller=tls_config.go:191 msg="TLS is disabled." http2=false
```

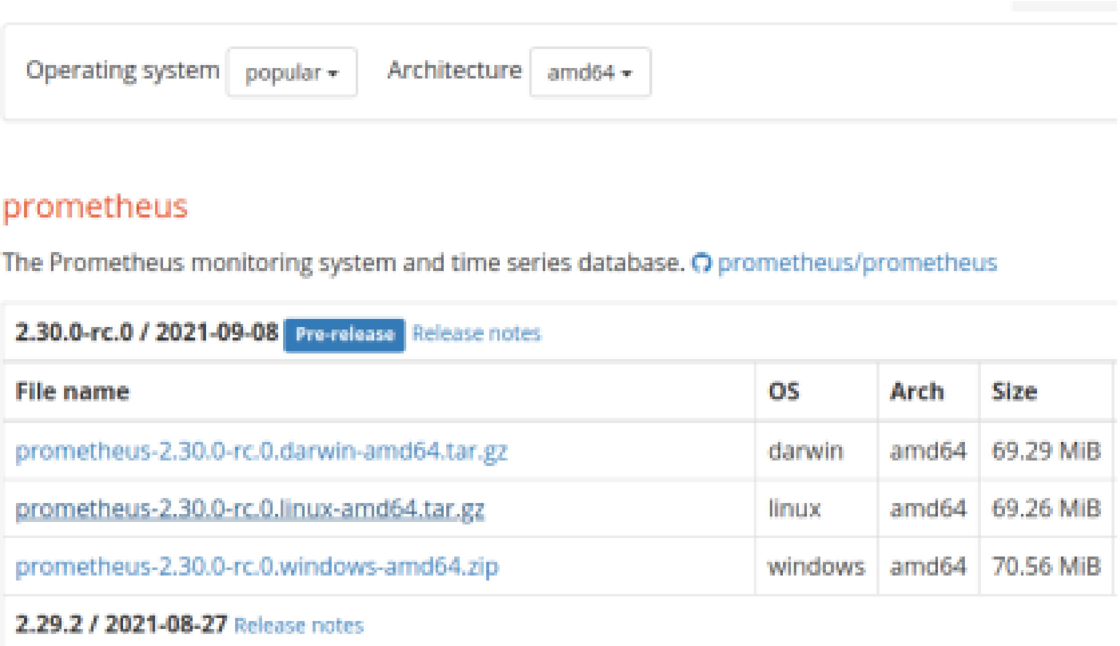
Consultamos el 9100 en el navegador



```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 7
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.16.7"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.420832e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.420832e+06
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash ta
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.44501e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 752
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used b
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 0
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 4.178024e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
```

Instalación de Prometheus

vamos a https://prometheus.io/download/



prometheus

The Prometheus monitoring system and time series database. prometheus/prometheus

2.30.0-rc.0 / 2021-09-08 Pre-release Release notes

File name	OS	Arch	Size
prometheus-2.30.0-rc.0.darwin-amd64.tar.gz	darwin	amd64	69.29 MiB
prometheus-2.30.0-rc.0.linux-amd64.tar.gz	linux	amd64	69.26 MiB
prometheus-2.30.0-rc.0.windows-amd64.zip	windows	amd64	70.56 MiB

2.29.2 / 2021-08-27 Release notes

Lo descargamos

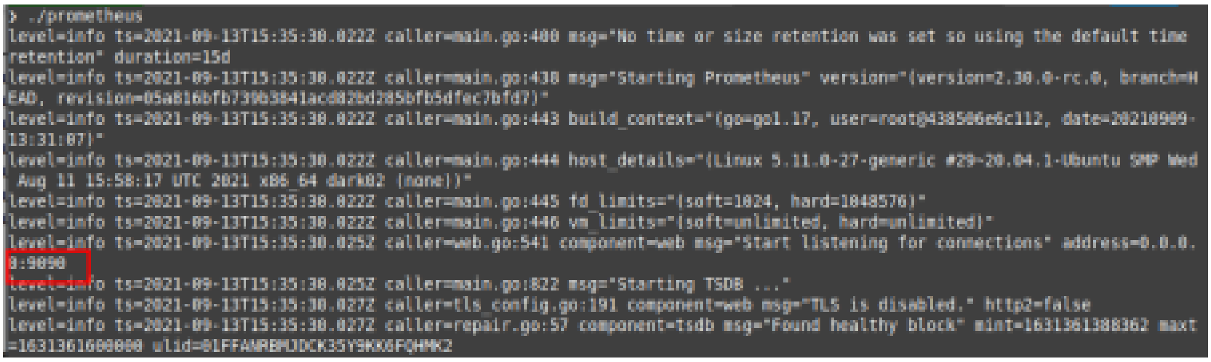
❯ wget https://github.com/prometheus/prometheus/releases/download/v2.30.0-rc.0/prometheus-2.30 .0-rc.0.linux-amd64.tar.gz

Descomprimimos

tar xvf prometheus-2.30.0-rc.0.linux-amd64.tar.gz

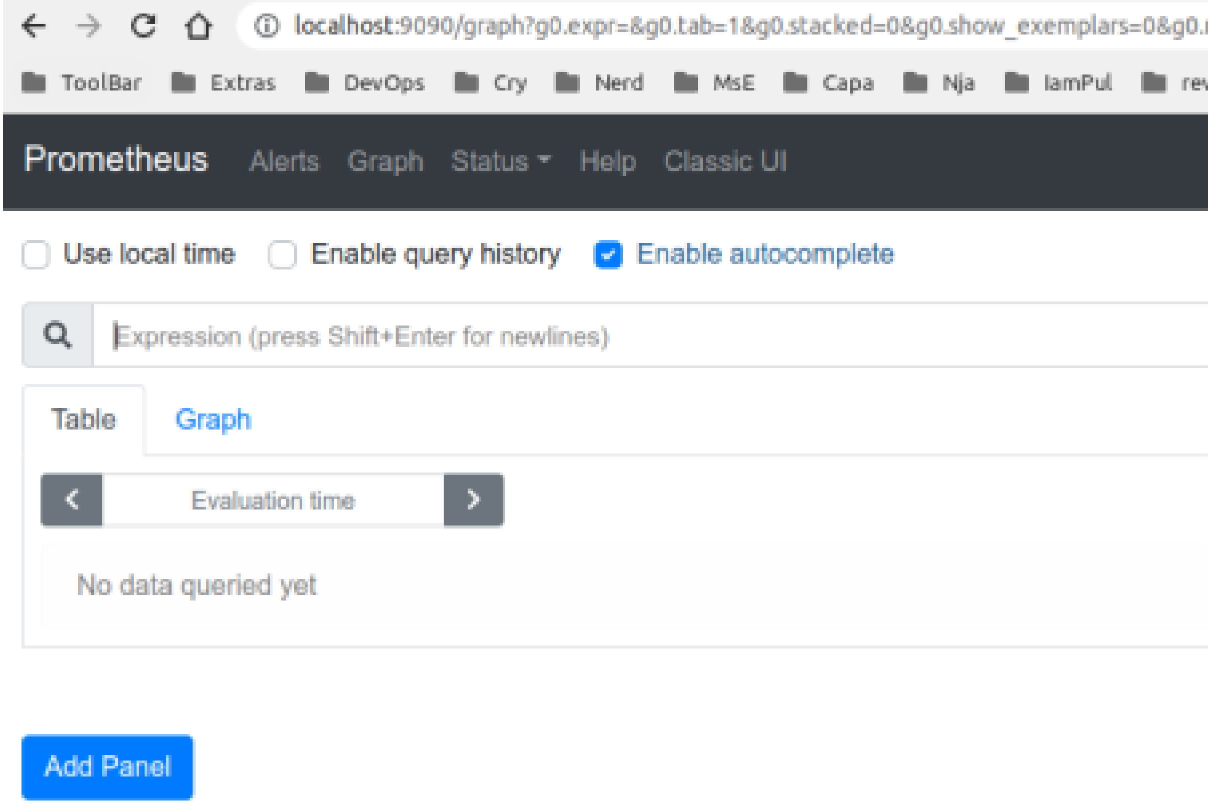
Levantamos el servicio

❯ ./prometheus



```
❯ ./prometheus
level=info ts=2021-09-13T15:35:30.022Z caller=main.go:400 msg="No time or size retention was set so using the default time
retention" duration=15d
level=info ts=2021-09-13T15:35:30.022Z caller=main.go:438 msg="Starting Prometheus" version="(version=2.30.0-rc.0, branch=H
EAD, revision=05a816bfb739b3841acd82bd285bfb5dfec7bfd7)"
level=info ts=2021-09-13T15:35:30.022Z caller=main.go:443 build_context="(go=go1.17, user=root@438506e6c112, date=20210909-
13:31:07)"
level=info ts=2021-09-13T15:35:30.022Z caller=main.go:444 host_details="(Linux 5.11.0-27-generic #29~20.04.1-Ubuntu SMP Wed
 Aug 11 15:58:17 UTC 2021 x86_64 dark02 (none))"
level=info ts=2021-09-13T15:35:30.022Z caller=main.go:445 fd_limits="(soft=1024, hard=1048576)"
level=info ts=2021-09-13T15:35:30.022Z caller=main.go:446 vm_limits="(soft=unlimited, hard=unlimited)"
level=info ts=2021-09-13T15:35:30.025Z caller=web.go:541 component=web msg="Start listening for connections" address=0.0.0.
0:9090
level=info ts=2021-09-13T15:35:30.025Z caller=main.go:822 msg="Starting TSDB ..."
level=info ts=2021-09-13T15:35:30.027Z caller=tls_config.go:191 component=web msg="TLS is disabled." http2=false
level=info ts=2021-09-13T15:35:30.027Z caller=repair.go:57 component=tsdb msg="Found healthy block" mint=1631361388362 maxt
=1631361600000 ulid=01FFANRBMJDCK35Y9KK6FQHMK2
```

Vamos al 9090

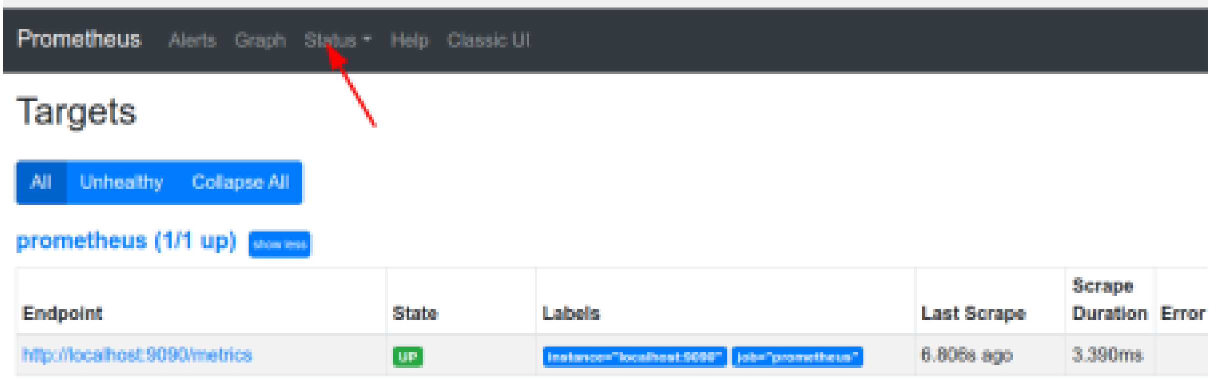


□ Use local time □ Enable query history ☑ Enable autocomplete

Expression (press Shift+Enter for newlines)

No data queried yet

Revisamos el status del servicio para sus targets



Targets

prometheus (1/1 up)

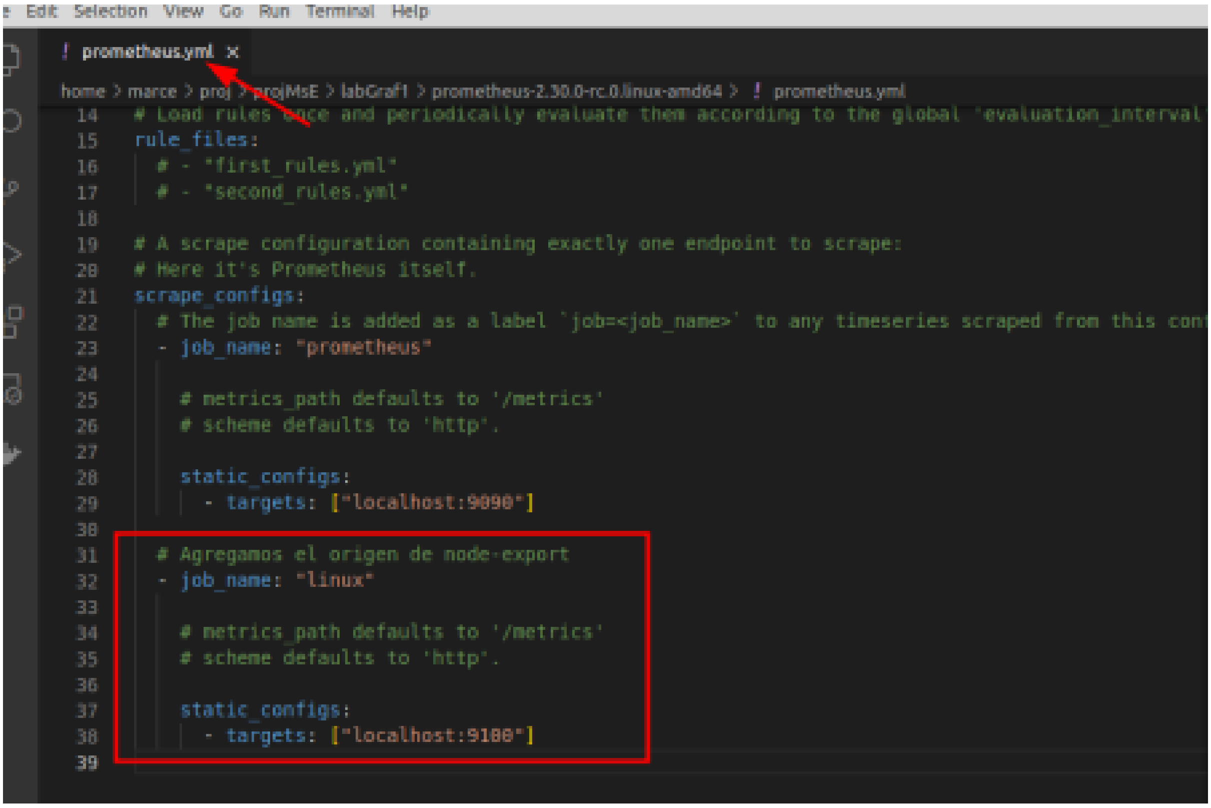
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	6.806s ago	3.390ms	

Podríamos modificar tanto el directorio de almacenamiento como la retención de datos expresada en días o en espacio de disco.

```
> cat /etc/systemd/system/prometheus.service
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/opt/Prometheus/prometheus \
--config.file /etc/prometheus/prometheus.yml \
--storage.tsdb.path /var/lib/prometheus/ \
--storage.tsdb.retention.time=1y \
--storage.tsdb.retention.size=10G \
--web.console.templates=/opt/Prometheus/consoles \
--web.console.libraries=/opt/Prometheus/console_libraries
```

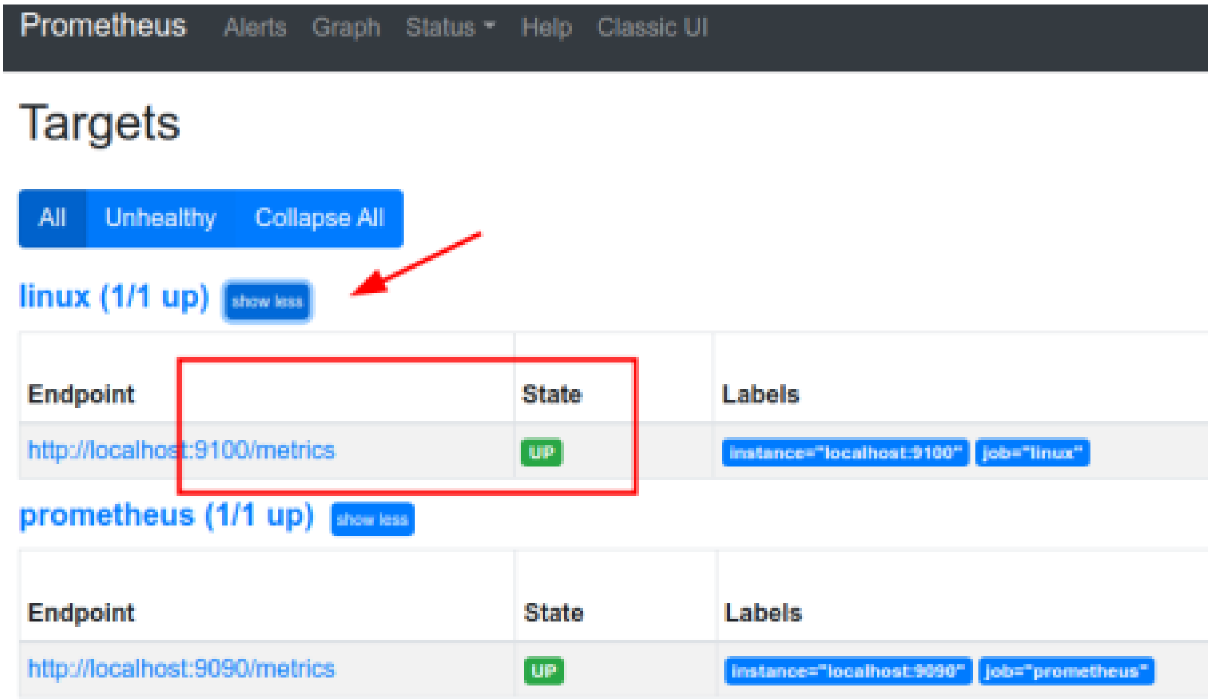
Detenemos prometheus y vamos a agregar el metric desde node-export

editamos ***prometheus.yml*** agregando la sección de métricas para un nuevo job


Edit Selection View Go Run Terminal Help
prometheus.yml
home > marce > proj > projMsE > labGraf1 > prometheus-2.30.0-rc.0.linux-amd64 > prometheus.yml

```
# Load rules once and periodically evaluate them according to the global 'evaluation_interval'
rule_files:
  # - "first_rules.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 conf
  - job_name: "prometheus"

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

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

  # Agregamos el origen de node-export
  - job_name: "linux"

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

    static_configs:
      - targets: ["localhost:9100"]
```

Verificamos que prometehus tome el nuevo servicio



Targets

linux (1/1 up)

Endpoint	State	Labels
http://localhost:9100/metrics	UP	instance="localhost:9100" job="linux"

prometheus (1/1 up)

Endpoint	State	Labels
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"

Sistema de visualización

Para la visualización del monitoreo utilizaremos Grafana

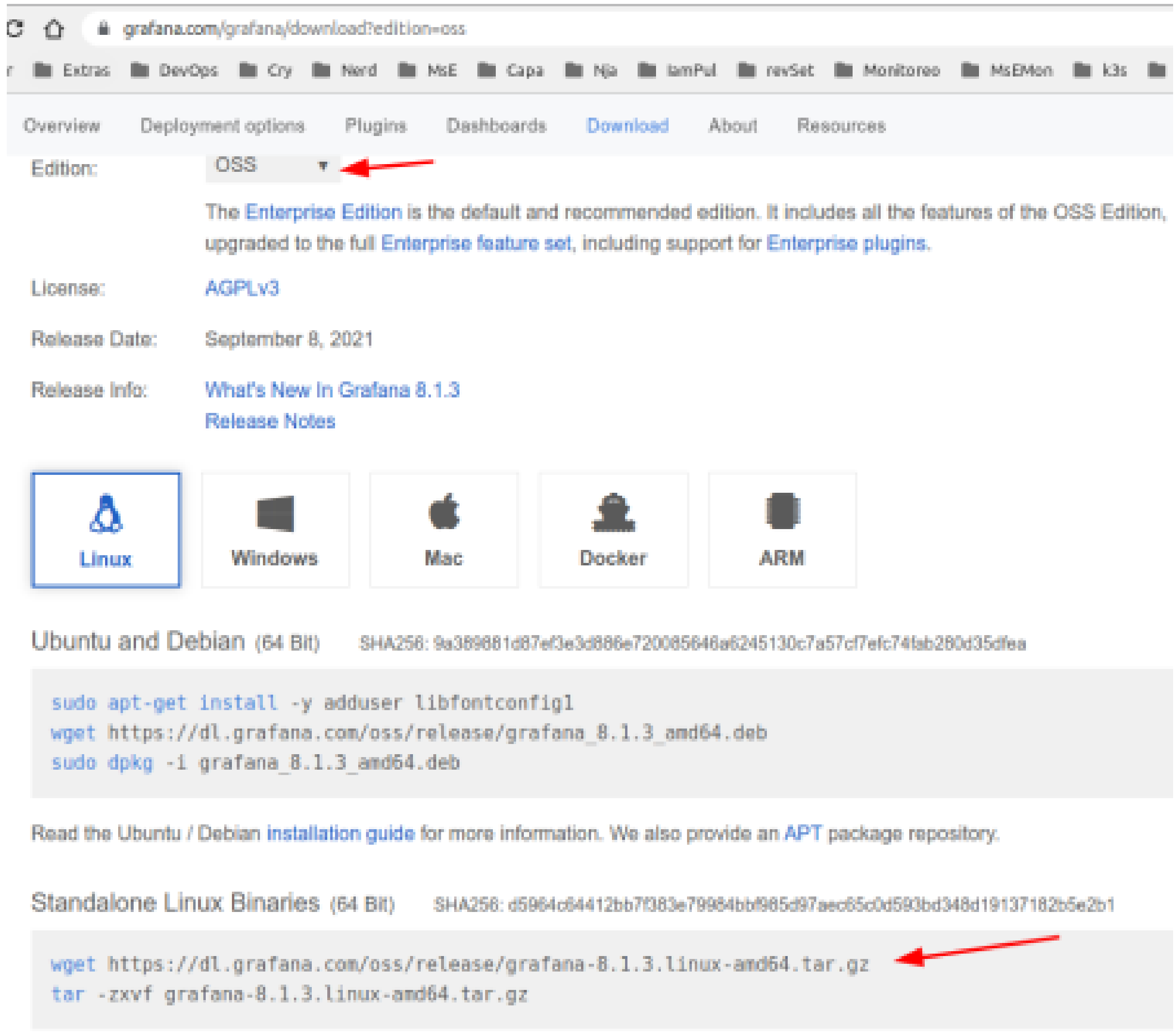
1. Descargar e instalar Grafana

2. Configuración de Grafana

3. Visualización del monitoreo

Vamos a descargar Grafana

https://grafana.com/grafana/download



Edition: OSS

The Enterprise Edition is the default and recommended edition. It includes all the features of the OSS Edition, upgraded to the full Enterprise feature set, including support for Enterprise plugins.

License: AGPLv3

Release Date: September 8, 2021

Release Info: What's New In Grafana 8.1.3
Release Notes

Ubuntu and Debian (64 Bit) SHA256: 9a389881d87ef3e3d886e720085646a6245130c7a57cf7efc74fab280d35dfea

```
sudo apt-get install -y adduser libfontconfig1
wget https://dl.grafana.com/oss/release/grafana_8.1.3_amd64.deb
sudo dpkg -i grafana_8.1.3_amd64.deb
```

Read the Ubuntu / Debian installation guide for more information. We also provide an APT package repository.

Standalone Linux Binaries (64 Bit) SHA256: d5964c64412bb7f383e79984bbf985d97aec65c0d593bd348d19137182b5e2b1

```
wget https://dl.grafana.com/oss/release/grafana-8.1.3.linux-amd64.tar.gz
tar -zxvf grafana-8.1.3.linux-amd64.tar.gz
```

```
❯ wget https://dl.grafana.com/oss/release/grafana-8.1.3.linux-amd64.tar.gz
```

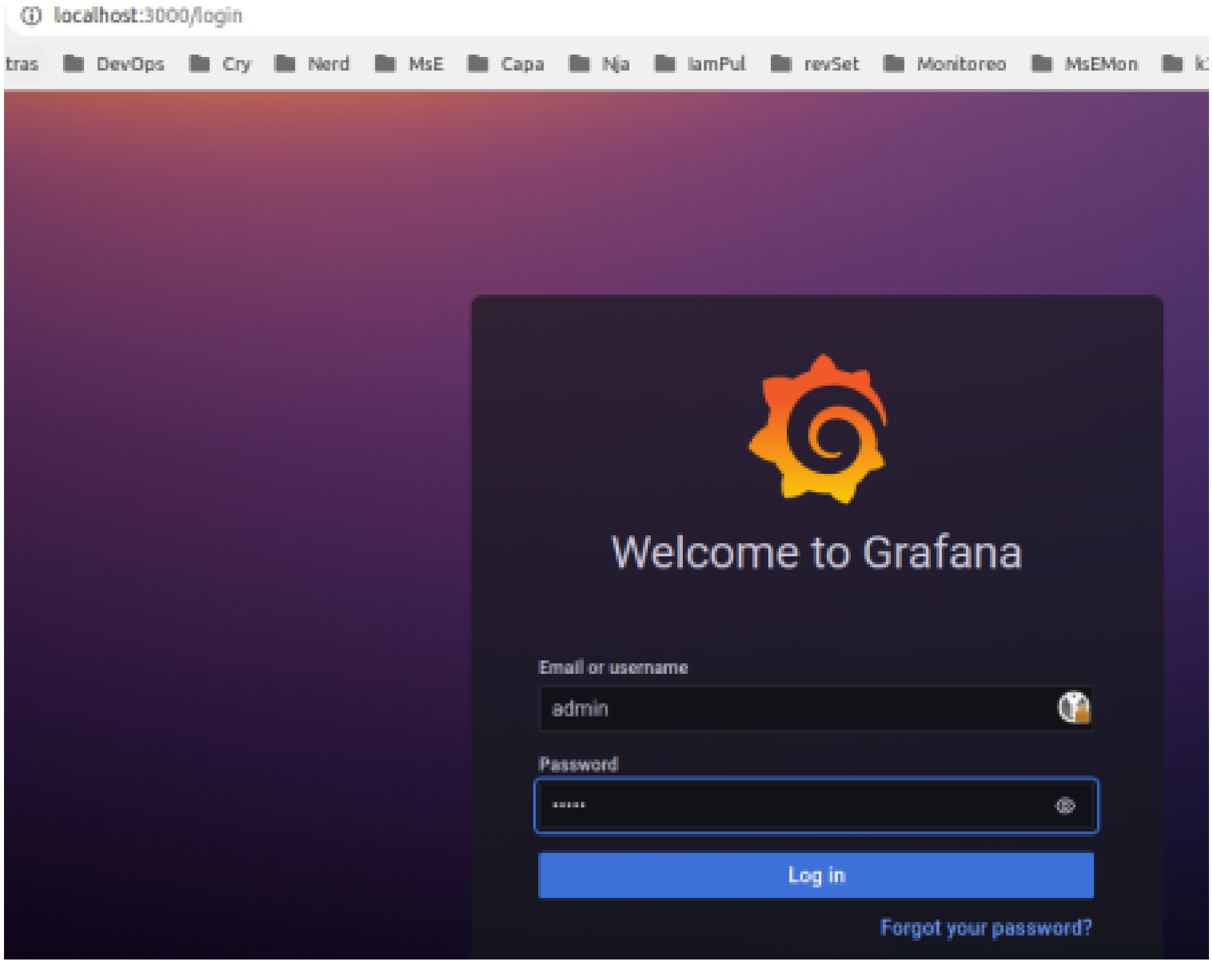
Descomprimimos

```
tar xvf grafana-8.1.3.linux-amd64.tar.gz
```

Levantamos el servicio

```
❯ ./grafana-server
```

Vamos al localhost:3000



localhost:3000/login

Welcome to Grafana

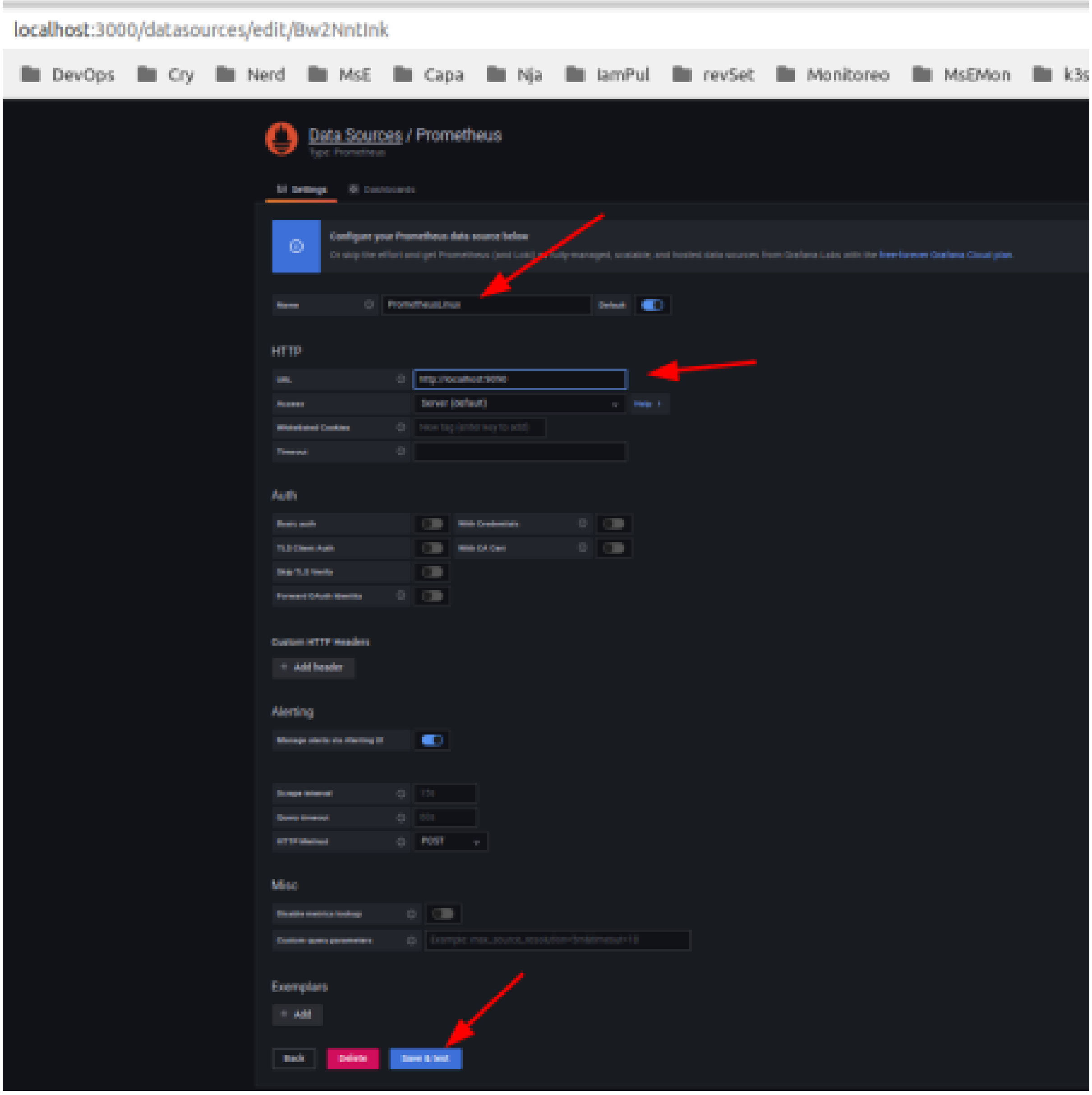
Email or username: admin

Password: •••••

cambiamos la pass

Vamos a agregar nuestro primer data source

tipo prometheus



localhost:3000/datasources/edit/Bw2NntInk

Data Sources / Prometheus

Type: Prometheus

Configure your Prometheus data source below

Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the free-forever Grafana Cloud plan.

Name: PrometheusLinux — Default: on

HTTP

URL: http://localhost:9090

Access: Server (default)

Whitelisted Cookies: New tag (enter key to add)

Timeout:

Auth

Basic auth: off — With Credentials: off

TLS Client Auth: off — With CA Cert: off

Skip TLS Verify: off

Forward OAuth Identity: off

Custom HTTP Headers

Alerting

Manage alerts via Alerting UI: on

Scrape interval: 15s

Query timeout: 60s

HTTP Method: POST

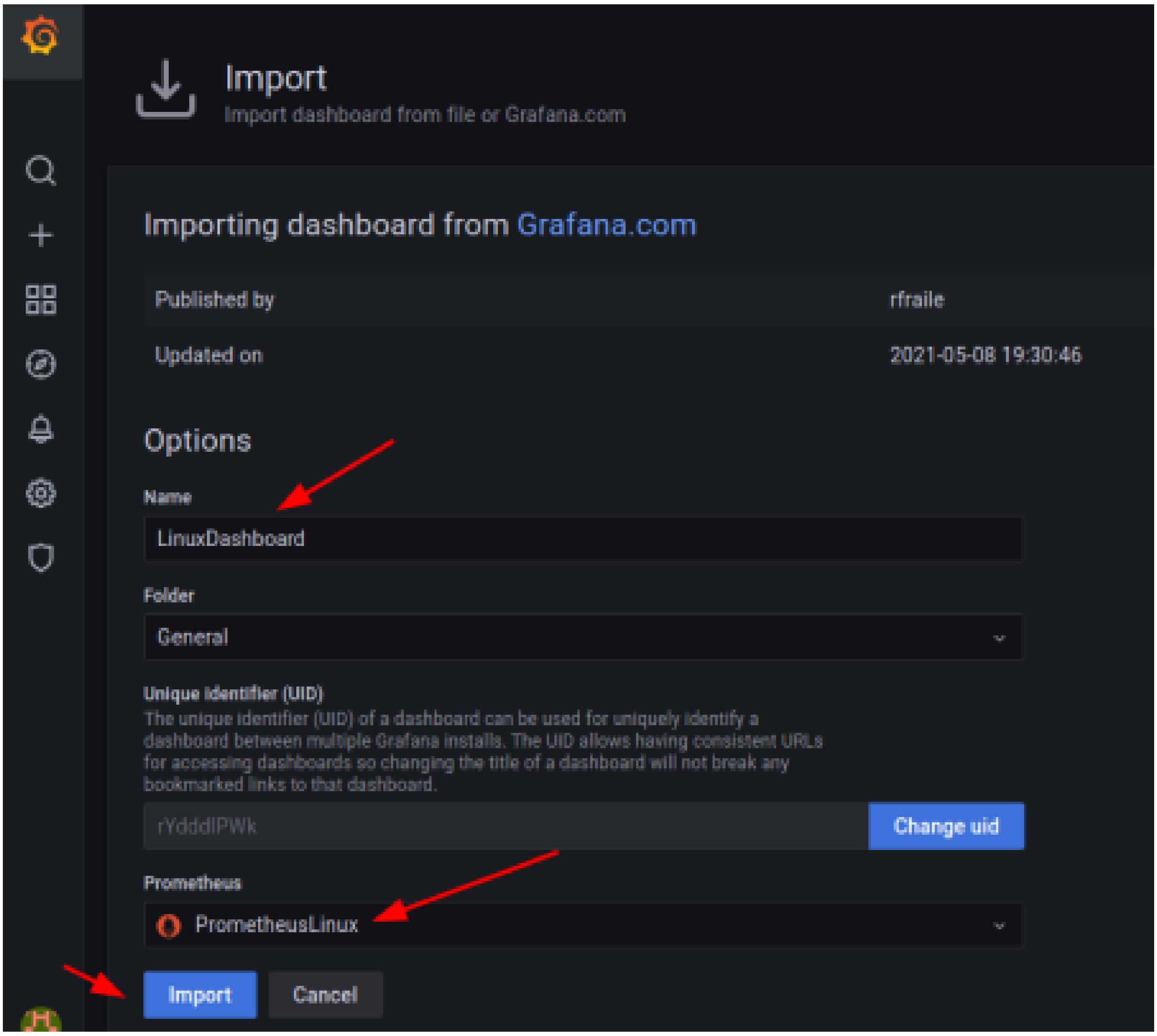
Misc

Disable metrics lookup: off

Custom query parameters: Example: max_source_resolution=5m&timeout=10

Exemplars

Ahora podemos crearnos nuestro dashboard o importar uno pre armado por ejemplo el 1860 sin olvidarnos de elegir el datasource correcto



Import

Import dashboard from file or Grafana.com

Importing dashboard from Grafana.com

Published by	rfraile
Updated on	2021-05-08 19:30:46

Options

Name: LinuxDashboard

Folder: General

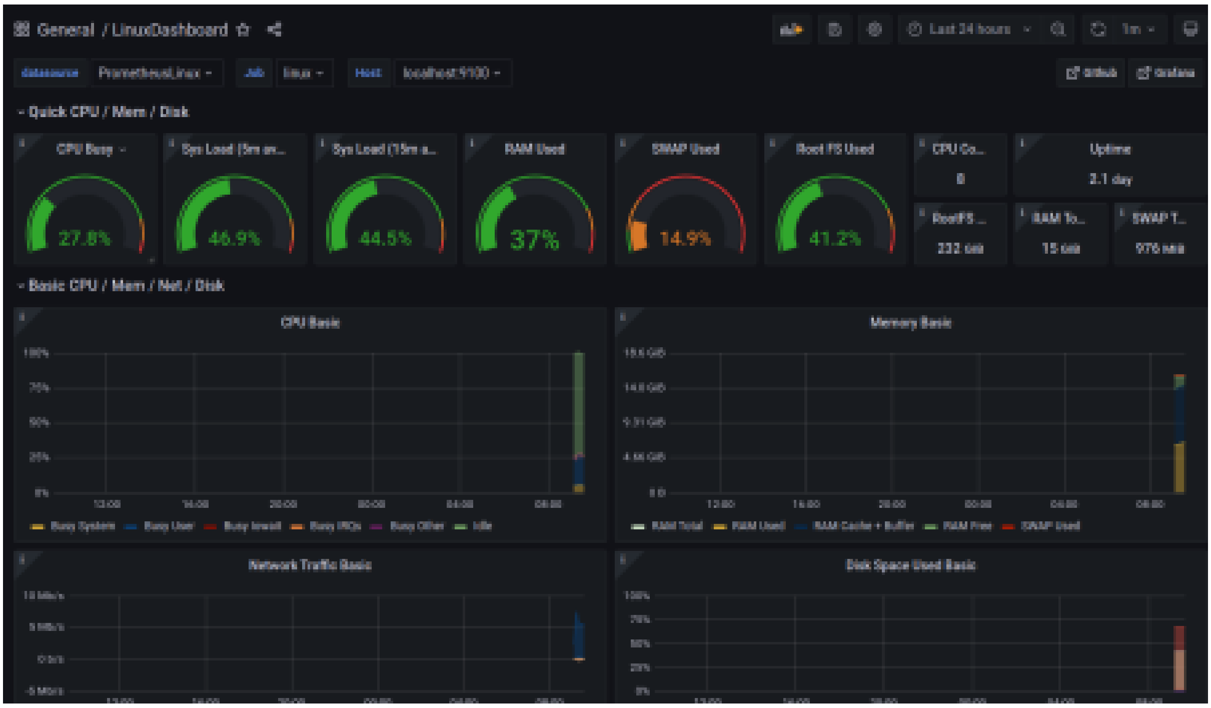
Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

rYdddlPWk

Prometheus: PrometheusLinux

Observamos el dashboard que se nos genero y ya esta registrando las metricas



General / LinuxDashboard
Last 24 hours
1m
datasource
PrometheusLinux
Job
linux
Host
localhost:9100
Quick CPU / Mem / Disk
CPU Busy
27.8%
Sys Load (5m av...
46.9%
Sys Load (15m a...
44.5%
RAM Used
37%
SWAP Used
14.9%
Root FS Used
41.2%
CPU Co...
8
Uptime
2.1 day
RootFS ...
232 GiB
RAM To...
15 GiB
SWAP T...
976 MiB
Basic CPU / Mem / Net / Disk
CPU Basic
Memory Basic
Network Traffic Basic
Disk Space Used Basic