

# Módulo de Herramientas de monitoreo

## Práctico 1

### Instalación de influxdb

Ingresamos a <https://www.influxdata.com/get-influxdb/> y bajamos la version la última version

#### InfluxDB 2.x Open Source Time Series Database

InfluxDB is an open source time series database. It has everything you need from a time series platform in a single binary – a multi-tenanted time series database, UI and dashboarding tools, background processing and monitoring agent. All this makes deployment and setup a breeze and easier to secure.

The InfluxDB Platform also includes APIs, tools, and an ecosystem that includes 10 client and server libraries, Telegraf plugins, visualization integrations with Grafana, Google Data Studio, and data sources integrations with Google Bigtable, BigQuery, and more.

##### Version

InfluxDB v2.2.0

##### Platform

Linux Binaries (64-bit)

SHA256: `87c9e4c356271120e7792b0fc437abac6c8933fbd56d6c4589372119823caf58`

```
wget https://dl.influxdata.com/influxdb/releases/influxdb2-2.2.0-linux-amd64.tar.gz
tar xvfz influxdb2-2.2.0-linux-amd64.tar.gz
```

[Documentation](#) [Release Notes](#)

```
> wget https://dl.influxdata.com/influxdb/releases/influxdb-1.7.6_linux_amd64.tar.gz
```

Descomprimimos

```
tar xvf influxdb-1.7.6_linux_amd64.tar.gz
```

Levantamos el servicio de la base

```
> cd influxdb-1.7.6-1/usr/bin
> ./influxd
```

Levantamos la CLI para consultas

```
> ./influx
```

Revisamos que tenga bases instaladas... debería tener la ***\_internal***

```
> show databases
```

```
name: databases
```

```
name
```

```
----
```

```
_internal
```

Salimos con

```
> quit
```

# Instalación de telegraf

Vamos a <https://portal.influxdata.com/downloads/>

## Telegraf open source data collector

Telegraf is a plugin-driven server agent for collecting and sending metrics and events from databases, systems, and IoT sensors. Telegraf is written in Go and compiles into a single binary with no external dependencies, and requires a very minimal memory footprint.

With 200+ plugins already written by subject matter experts on the data in the community, it is easy to start collecting metrics from your endpoints.

Version

Telegraf v1.22.4

Platform

Linux Binaries (64-bit)

SHA256: `31fdd105943a53a248b783e34c87e98b215582d710b67933e16bc27d4ab41cde`

```
wget https://dl.influxdata.com/telegraf/releases/telegraf-1.22.4_linux_amd64.tar.gz
tar xf telegraf-1.22.4_linux_amd64.tar.gz
```

[Documentation](#) [Release Notes](#)

# Descargar

```
wget https://dl.influxdata.com/telegraf/releases/telegraf-1.10.3_linux_amd64.tar.gz
```

# Descomprimir

```
tar xf telegraf-1.10.3_linux_amd64.tar.gz
```

# Lo configuramos eligiendo algunas metricas a capturar

```
> cd telegraf-1.20.0/usr/bin
```

```
> telegraf -sample-config -input-filter cpu:mem:disk -output-filter influxdb > telegraf.conf
```

# Chequeamos el archivo de configuracion en especial al final

```

telegraf.conf x
home > marce > proj > projMsE > labGraf1 > telegraf-1.20.0 > usr > bin > telegraf.conf
942 # ## The fields for which the values will be counted
943 #   fields = []
944
945
946 #####
947 #                               INPUT PLUGINS                               #
948 #####
949
950
951 # Read metrics about cpu usage
952 [[inputs.cpu]]
953     ## Whether to report per-cpu stats or not
954     percpu = true
955     ## Whether to report total system cpu stats or not
956     totalcpu = true
957     ## If true, collect raw CPU time metrics
958     collect_cpu_time = false
959     ## If true, compute and report the sum of all non-idle CPU states
960     report_active = false
961
962
963 # Read metrics about disk usage by mount point
964 [[inputs.disk]]
965     ## By default stats will be gathered for all mount points.
966     ## Set mount_points will restrict the stats to only the specified mount points.
967     # mount_points = ["/"]
968
969     ## Ignore mount points by filesystem type.
970     ignore_fs = ["tmpfs", "devtmpfs", "devfs", "iso9660", "overlay", "aufs", "squashfs"]
971
972
973 # Read metrics about memory usage
974 [[inputs.mem]]
975     # no configuration
976
977

```

# Arrancar el agente de telegraf

```
> ./telegraf --config telegraf.conf
```

2021-09-08T00:49:09Z !! Starting Telegraf 1.10.3

2021-09-08T00:49:09Z !! Loaded inputs: mem cpu disk

2021-09-08T00:49:09Z !! Loaded aggregators:

2021-09-08T00:49:09Z !! Loaded processors:

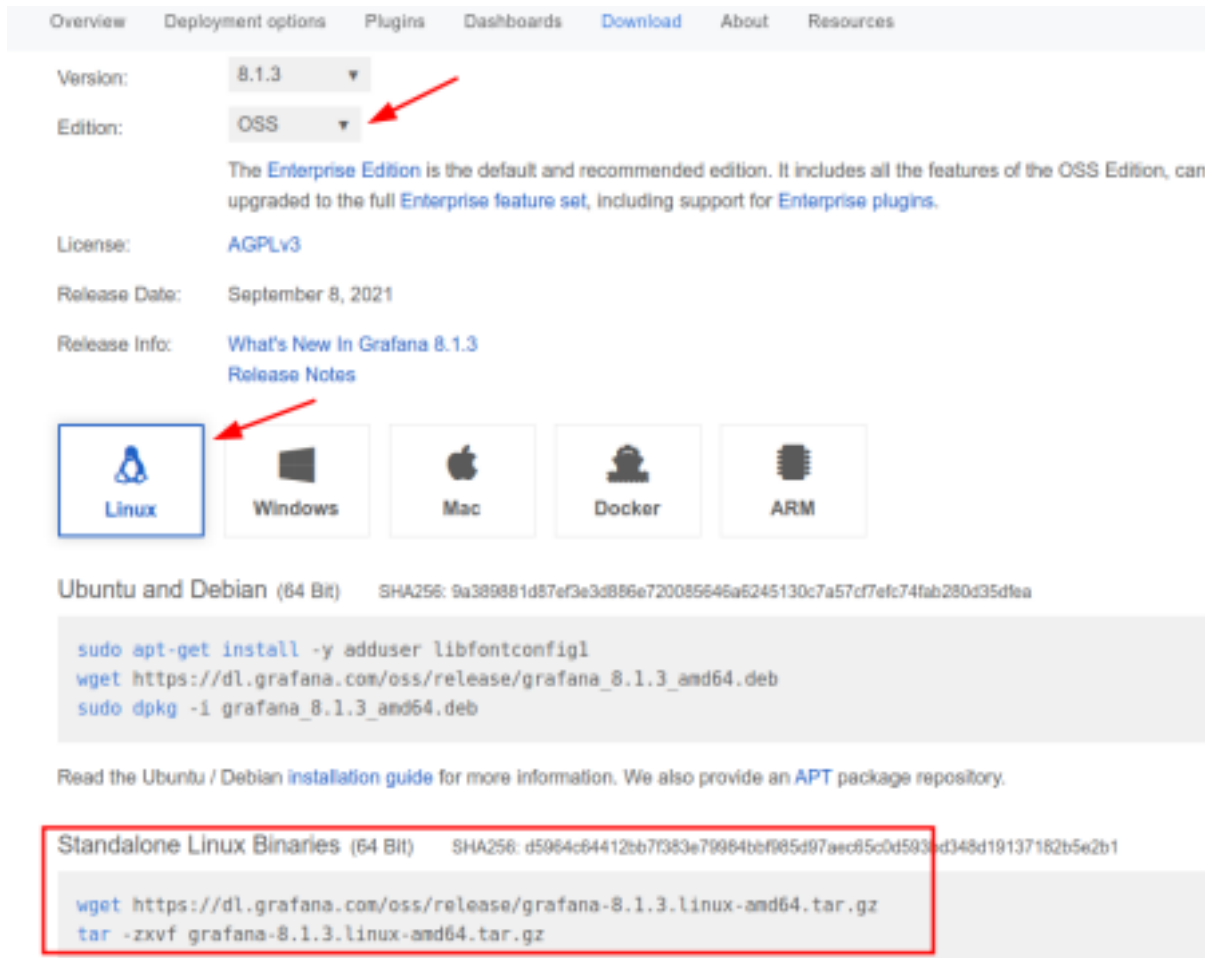
2021-09-08T00:49:09Z !! Loaded outputs: influxdb

2021-09-08T00:49:09Z !! Tags enabled: host=dark02

2021-09-08T00:49:09Z !! [agent] Config: Interval:10s, Quiet:false, Hostname:"dark02", Flush Interval:10s

# Instalar Grafana

Vamos a <https://grafana.com/grafana/download> y elegimos la version OS



The screenshot shows the Grafana download page. The 'Download' tab is active. Under 'Version', '8.1.3' is selected. Under 'Edition', 'OSS' is selected, indicated by a red arrow. Below this, there is a note about the Enterprise Edition. The 'License' is 'AGPLv3'. The 'Release Date' is 'September 8, 2021'. The 'Release Info' includes links to 'What's New In Grafana 8.1.3' and 'Release Notes'. Below the release info, there are five icons for different operating systems: Linux, Windows, Mac, Docker, and ARM. The 'Linux' icon is highlighted with a red box and a red arrow. Below the icons, there are two sections for Linux binaries. The first section is for 'Ubuntu and Debian (64 Bit)' and provides a list of commands to install Grafana using APT. The second section is for 'Standalone Linux Binaries (64 Bit)' and provides a list of commands to download and extract the binary. The 'Standalone Linux Binaries' section is highlighted with a red box.

Overview Deployment options Plugins Dashboards **Download** About Resources

Version: 8.1.3  
Edition: OSS  
License: AGPLv3  
Release Date: September 8, 2021  
Release Info: [What's New In Grafana 8.1.3](#)  
[Release Notes](#)

Linux Windows Mac Docker ARM

Ubuntu and Debian (64 Bit) SHA256: 9a389881d87e3e3d886e720085646a6245130c7a57cf7efc74fab280d35dfea

```
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: d5964c64412bb7f383e7f9864b6f065d97aec65c0d593d348d19137182b5e2b1

```
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
```

# Descargar

```
> wget https://dl.grafana.com/oss/release/grafana-6.1.4.linux-amd64.tar.gz
```

# Descomprimir

```
> tar -zxvf grafana-6.1.4.linux-amd64.tar.gz
```

# Arrancamos el servicio

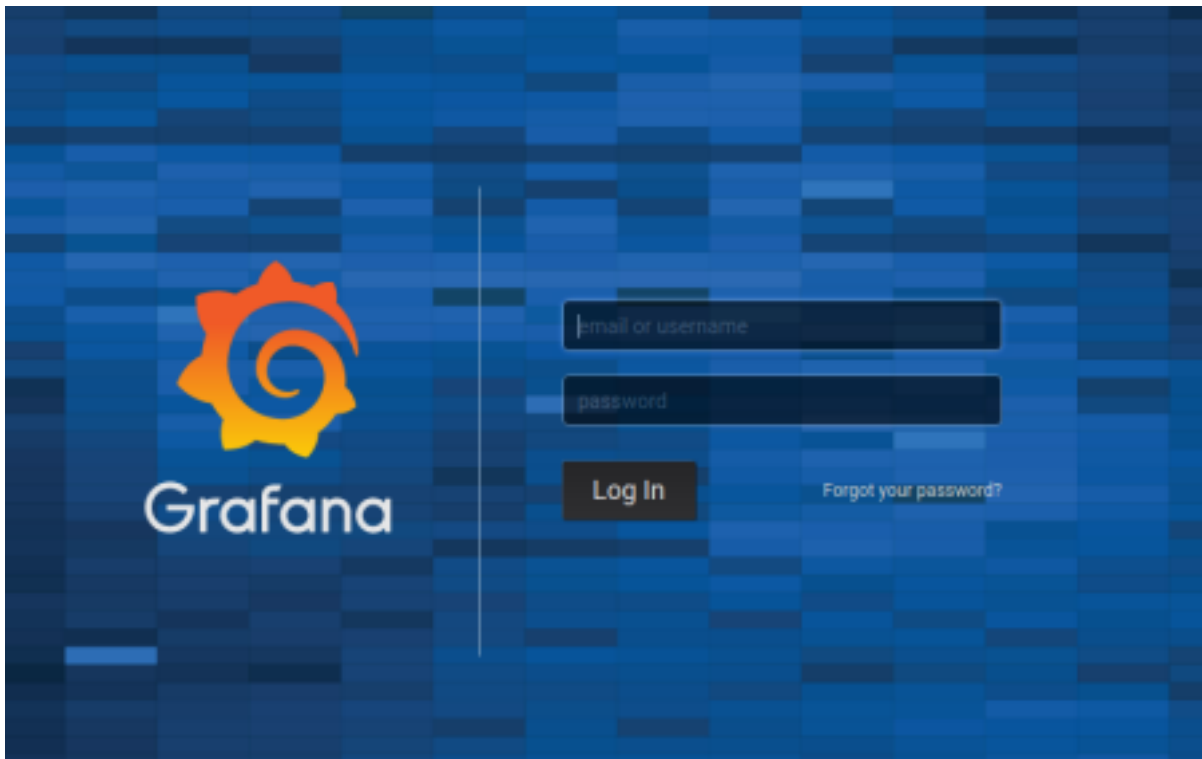
```
> cd grafana-6.1.4/bin
```

```
> ./grafana-server
```

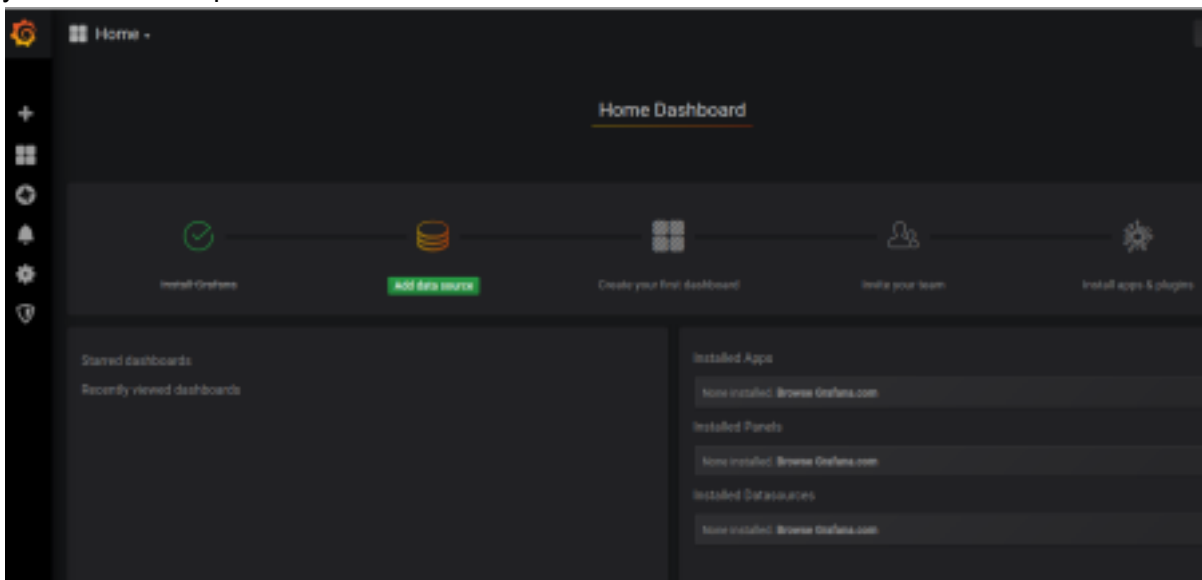


a tenemos la base de datos para almacenar la info, el agente que sacara las mediciones y solo nos falta configurar el tablero de monitoreo, osea Grafana


Vamos a localhost:3000 con **admin:admin**, luego nos pedirá cambio de password



y accedemos al panel inicial



Agregamos el datasource correcto InfluxDB y sus valores

 **Data Sources / InfluxDB**  
Type: InfluxDB

Settings

Name

InfluxDB

Default

☒

HTTP

URL

http://localhost:8086

Access

Server (Default)

Help ▶

Whitelisted Cookies

Add Name

Auth

Basic Auth

☐

With Credentials

TLS Client Auth

☐

With CA Cert

Skip TLS Verify

☐

Forward OAuth Identity


☐

InfluxDB Details

Database

telegraf

User



Password

Luego vamos a crear un nuevo panel

## Parametros de CPU

Queries to InfluxDB

▼ A

FROM	default	cpu	WHERE	+	
SELECT	field (usage_idle)	mean ()	+		
GROUP BY	time (\$__interval)	fill (null)	+		
FORMAT AS	Time series				
ALIAS BY	Naming pattern				

Min time interval 0 Relative time 1h Time shift 1h

## Parametros de memoria

Queries to InfluxDB

▼ A

FROM	default	mem	WHERE	+	
SELECT	field (available)	mean ()	+		
GROUP BY	time (\$__interval)	fill (null)	+		
FORMAT AS	Time series				
ALIAS BY	Naming pattern				

Y vemos nuestro dashboard ya creado

