

# Monitor your Systems with Telegraf, InfluxDB, and Grafana

## A FreeBSD-focused Howto

Benedict Reuschling  
bcr@FreeBSD.org

National Taiwan University of Science and Technology

COSCU 2019  
Taipei, Taiwan

## Who am I?

- Benedict Reuschling, M.Sc. Computer Science
- Big Data Cluster Admin at University of Applied Sciences, Darmstadt, Germany
- FreeBSD Committer since 2009
- Vice-President FreeBSD Foundation
- 2nd term on FreeBSD's elected Core Team
- Weekly podcast called BSDNow.tv with Allan Jude

## Monitoring

Typically, monitoring deals with the following three aspects:

**Availability** - Is the host or service available on the network?

**Metrics** - **Data collected from hosts and services**

**Logs** - Messages (errors, warnings) written to log files from hosts and services

# Overview

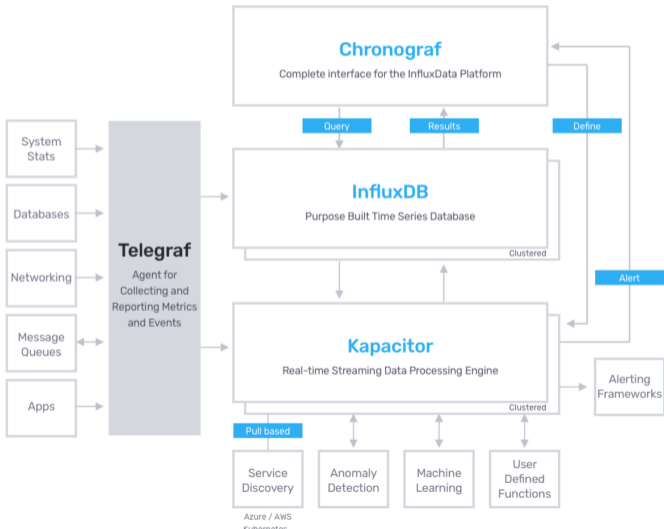
## ① Configuring the Monitoring System

- Configuring InfluxDB

- Configuring Telegraf

- Configuring Grafana

# InfluxDB Architecture



## Components

The following tools were chosen for their simplicity, which will quickly get you started monitoring your systems.

**Telegraf** Sends collected metrics to InfluxDB

**InfluxDB** Stores the collected metrics for retrieval by Grafana

**Grafana** Dashboard to display the metrics

Generally, many combinations of tools possible. For example, InfluxDB can be used as a database by many metrics collectors. Likewise, Grafana allows more than just InfluxDB as a data source.

## Installing the Required Software Packages

```
# pkg install influxdb telegraf grafana6
```

We use `sysrc(8)` to add entries to `/etc/rc.conf` to make these services start upon reboot:

```
sysrc influxd_enable=yes  
sysrc telegraf_enable=yes  
sysrc grafana_enable=yes
```

# Overview

## ① Configuring the Monitoring System

Configuring InfluxDB

Configuring Telegraf

Configuring Grafana



## InfluxDB's Configuration File 1/2

Not much to change in `/usr/local/etc/influxd.conf`:

```
1 [http]
2   enabled = true
3   bind-address = ":8086"
```

The rest are sensible defaults that we can keep for now.

We can now start the influxd service:

```
# service influxd start
```

## Creating the Database and Defining Retention Policy

Now that the service is started, we run the interactive influx shell: `$ influx`

```
Connected to http://localhost:8086 version 1.7.6
```

```
InfluxDB shell version: 1.7.6
```

```
Enter an InfluxQL query
```

```
> CREATE DATABASE telegraf
```

```
> CREATE RETENTION POLICY "a_year" ON "telegraf" DURATION 52w REPLICATION 1
```

```
> SHOW DATABASES
```

```
name: databases
```

```
name
```

```
----
```

```
_internal
```

```
telegraf
```

## Creating the Database User for Grafana

```
> CREATE USER telegraf WITH PASSWORD 'yourhardtoguesspasswordgoeshere '  
> GRANT ALL ON telegraf TO telegraf  
> SHOW USERS  
user      admin  
----      -  
telegraf  false
```

Exit out of the influxd shell by entering quit or typing `Ctrl`+`D`.

# Overview

## ① Configuring the Monitoring System

Configuring InfluxDB

Configuring Telegraf

Configuring Grafana

## About Telegraf

*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.*

`https://www.influxdata.com/time-series-platform/telegraf/`

## Telegraf's Configuration File

Make the following changes in `/usr/local/etc/telegraf.conf`:

```
[[outputs.influxdb]]
  urls = ["http://ip.of.influxdb.server:8086"] # required
logfile = "/var/log/telegraf/telegraf.log"
[[inputs.cpu]]
  percpu = true
  totalcpu = true
  collect_cpu_time = false
  report_active = false
[[inputs.disk]]
  ignore_fs = ["tmpfs", "devtmpfs", "devfs", "overlay", "aufs", "squashfs"]
[[inputs.diskio]]
[[inputs.kernel]]
[[inputs.mem]]
[[inputs.processes]]
[[inputs.swap]]
[[inputs.system]]
```

## Starting the InfluxDB Service

Once inputs are defined as needed, then we start the InfluxDB service:

```
# service influxd start
```

# Overview

## ① Configuring the Monitoring System

Configuring InfluxDB

Configuring Telegraf

Configuring Grafana



## Starting Grafana

Start the Grafana service:

```
# service grafana start
```

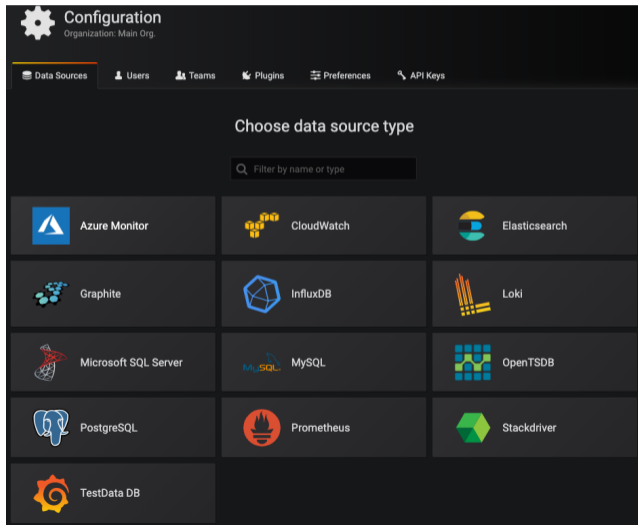
Wait until Grafana becomes available on port 3000.

Check with: `sockstat -l|grep 3000`

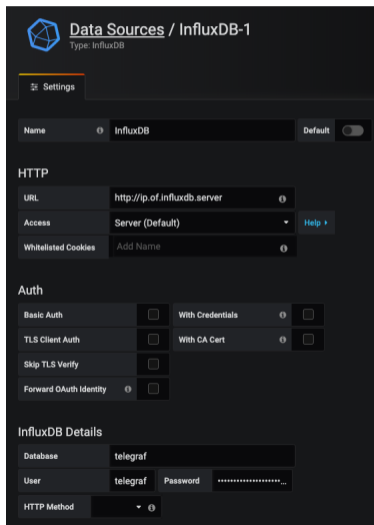
Once the service is running, open a browser and go to

```
http://<ip.of.grafana.server>:3000
```

# Configuring Grafana - Adding a Data Source: InfluxDB



# Configuring Grafana - Configuring the InfluxDB Data Source



The screenshot shows the Grafana configuration interface for an InfluxDB data source. The page title is "Data Sources / InfluxDB-1" with a sub-label "Type: InfluxDB". A "Settings" tab is active. The configuration is organized into sections: "Name" (InfluxDB, Default toggle), "HTTP" (URL: http://ip.of.influxdb.server, Access: Server (Default), Whitelisted Cookies: Add Name), "Auth" (Basic Auth, TLS Client Auth, Skip TLS Verify, Forward OAuth Identity), and "InfluxDB Details" (Database: telegraf, User: telegraf, Password: masked, HTTP Method: dropdown).

**Data Sources / InfluxDB-1**  
Type: InfluxDB

Settings

Name: InfluxDB Default

**HTTP**

URL: http://ip.of.influxdb.server

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**

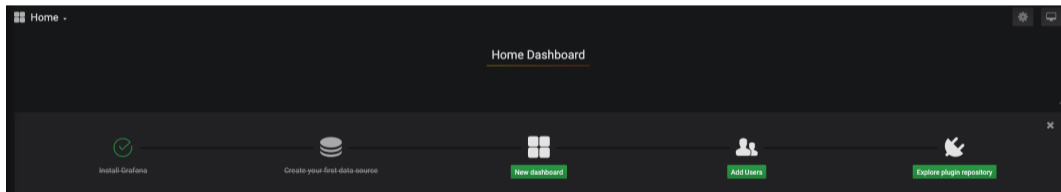
Database: telegraf

User: telegraf Password: .....

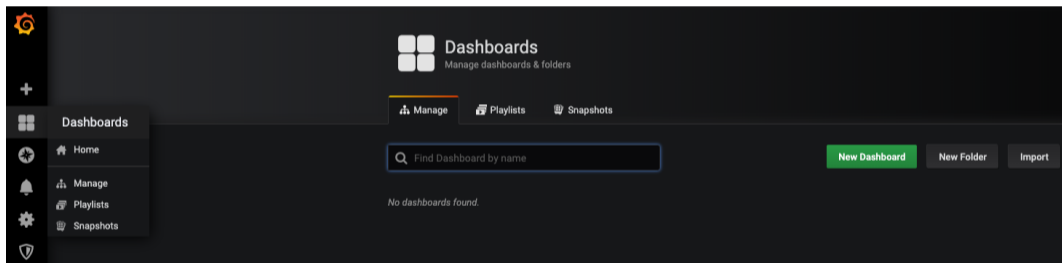
HTTP Method:

We add the datasource by clicking the "Save & test" 19 / 26

## Configuring Grafana - Setup Steps

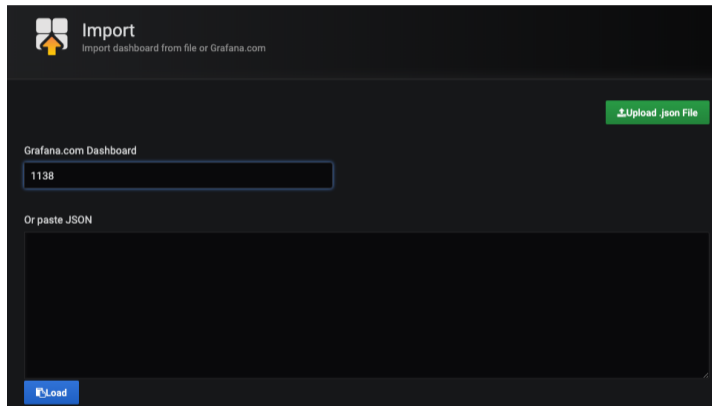


## Configuring Grafana - Importing a Dashboard from grafana.com



Click the "Import" button on the right

## Configuring Grafana - Importing the Dashboard



Import  
Import dashboard from file or Grafana.com

Upload json File

Grafana.com Dashboard

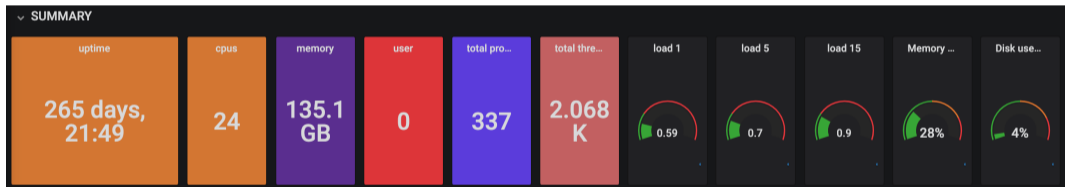
1138

Or paste JSON

Load

Just paste the dashboard ID from grafana.com into the first field and then tab to another. The information will be filled out for you. Select the influxdb data source we created earlier. Done!

## The Result



## Summary

InfluxDB, Grafana, and Telegraf work very well together and are fairly easy to set up. They can be extended with plugins and a lot of other integrations are possible. Setting up this monitoring stack on FreeBSD is easy and takes less time. It enables FreeBSD to become your logging and graphing solution so that you know what is going on in your networked machines.



# Questions?

Questions?

Thanks for listening!

## More Information



Grafana Website

<https://grafana.com/>



Telegraf Website

<https://www.influxdata.com/time-series-platform/telegraf/>



InfluxDB Website

<https://www.influxdata.com/products/influxdb-overview/>