Grafana and InfluxDB with 128T

Overview

This document will guide you through the setup of InfluxDB and Grafana to collect metrics from your 128T router or conductor.

Please note, previous versions of this guide were written to run Grafana and InfluxDB co-resident on a router or conductor instance. With the release of 128T version 4.0, this is no longer recommended due the incorporation of InfluxDB into the 128T product.

Running a separate instance of InfluxDB will ensure that there are no conflicts.

Setup InfluxDB

Basic steps are listed below, but we recommend using the latest release and following the official documentation for installing your InfluxDB instance.

Download and install InfluxDB

Note: Do not install InfluxDB on your 128T router or conductor running version 4.0 or greater. It is already installed as a dependency and will cause problems if you install a different version.

```
$ wget https://dl.influxdata.com/influxdb/releases/influxdb-1.7.2.x86_64.rpm
$ sudo yum localinstall influxdb-1.7.2.x86_64.rpm
```

Start the InfluxDB service

```
$ sudo systemctl start influxdb
```

Create a database

Use the Influx CLI to create a new database. We will call ours 'analytics', but you can name it whatever you'd like.

```
$ influx
Connected to http://localhost:8086 version 1.7.2
InfluxDB shell version: 1.7.2
>
> create database analytics
>
> show databases
name: databases
name
----
_internal
analytics
> quit
```

Install Influx-Importer

Influx-importer is an application that utilizes the 128T API to collect metrics and insert them into InfluxDB. If pointed at a conductor it is able to collect data from all of the connected routers. However, be aware that this can generate a lot of traffic between those routers and conductor. Keep this in mind as you decided what data you want to collect.

Download and Install

The latest release of influx-importer can be found on github: https://github.com/128technology/influx-importer/releases

Extract the archive, and then place only the influx-importer executable in /usr/local/bin/influx-importer

Verify the user and group of the influx-importer binary, if necessary chown to root.

```
$ wget
https://github.com/128technology/influx-importer/releases/download/v0.1.1/in
flux-importer_linux_amd64.tar.gz
…
2019-01-03 15:49:14 (3.18 MB/s) - ‘influx-importer_linux_amd64.tar.gz’ saved
[2788026/2788026]
$ tar zxvf influx-importer_linux_amd64.tar.gz influx-importer
influx-importer
$ sudo mv influx-importer /usr/local/bin/
$ sudo chown root:root /usr/local/bin/influx-importer
```

Generate initial config file

The Influx-importer ‘init’ command will prompt you for the address and credentials of your 128T and will create a starter config file for you. The credentials are used just once to obtain a token which will be used for all future API calls.

```
$ influx-importer init --out influx-importer.conf
128T URL: https://ncricket2.lab
128T Username: admin
128T Password: *********

Retriving 128T token…
Retriving 128T available metrics…
Done.

Configuration successfully writen to "influx-importer.conf"
Additional changes are required within the configuration file before you start
the application.
```

Modify configuration

In the [influx] section update the parameters to match your influxdb config.

In this example, we will assume that InfluxDB and influx-importer are running on the same system. If this is not the case, set the address field accordingly.

```
[influx]
# The address of the Influx instance which is typically a HTTP address.
address=http://127.0.0.1:8086
username=
password=
database=analytics
```

The [metrics] section lists all of the available metrics you are able to collect. Uncomment the ones you want to start collecting, or paste the list below to the bottom of your config as a starting point.

```
interface/received/utilization
interface/sent/packets
interface/sent/bytes
interface/sent/error
interface/sent/utilization
bfd/received/total
bfd/peer-path/latency
bfd/peer-path/jitter
bfd/peer-path/loss
session/flow/active
session/active
routing-agent/fib-meter
aggregate-session/device-interface/bandwidth
aggregate-session/device-interface/session-count
aggregate-session/device-interface/session-arrival-rate
aggregate-session/device-interface/session-departure-rate
```

```
aggregate-session/device-interface/total-data
aggregate-session/device-interface/rx-data
aggregate-session/device-interface/tx-data
aggregate-session/network-interface/bandwidth
aggregate-session/network-interface/session-count
aggregate-session/network-interface/session-arrival-rate
aggregate-session/network-interface/session-departure-rate
aggregate-session/node/bandwidth
aggregate-session/node/session-count
aggregate-session/node/session-arrival-rate
aggregate-session/node/session-departure-rate
aggregate-session/node/total-data
aggregate-session/service/bandwidth
aggregate-session/service-class/bandwidth
aggregate-session/service-group/bandwidth
aggregate-session/service-route/bandwidth
aggregate-session/tenant/session-count
aggregate-session/tenant/session-arrival-rate
aggregate-session/tenant/session-departure-rate
cpu/utilization
memory/capacity
memory/used
disk/capacity
disk/used
disk/component
```

Run Influx-importer

When run with the ‘export’ command and --config file specified, you should see output similar to the following. Be sure to look out for any WARN or ERRO messages.

```
$ influx-importer extract --config=influx-importer.conf
Fri Jan  4 15:45:01 UTC 2019 Starting influx-importer

2019/01/04 15:45:02.423188 INFO Exported last 312 seconds of
bfd/peer-path/latency(router=...).
```

```
2019/01/04 15:45:02.591342 INFO Exported last 312 seconds of
cpu/utilization(node=aws-vm,router=conductor-field-eng,core=3).

2019/01/04 15:45:03.099756 INFO Exported last 312 seconds of
aggregate-session/node/bandwidth(router=...).

2019/01/04 15:45:03.135462 INFO Exported last 312 seconds of
memory/used(node=...).
...
```

If Influx-importer has run successfully, you should now see measurements in your database.

```
$ influx
Connected to http://localhost:8086 version 1.7.2
InfluxDB shell version: 1.7.2
>
> use analytics;
Using database analytics
> show measurements;
name: measurements
name
----
aggregate-session/node/bandwidth
aggregate-session/node/session-arrival-rate
aggregate-session/node/session-count
aggregate-session/node/session-departure-rate
aggregate-session/service-class/bandwidth
aggregate-session/service-group/bandwidth
aggregate-session/service-route/bandwidth
aggregate-session/service/bandwidth
aggregate-session/tenant/session-arrival-rate
aggregate-session/tenant/session-count
aggregate-session/tenant/session-departure-rate
Alarm-history
bfd/peer-path/jitter
bfd/peer-path/latency
```

```
bfd/peer-path/loss
cpu/utilization
disk/capacity
disk/component
disk/used
memory/capacity
memory/used
> quit
```

Setup Cron Job

The final step is to setup a cron job to automate the collection.

Add the following entry as a single line to /etc/crontab. Substitute ‘t128’ for the user you wish to use and specify the full path to your config file. We optionally direct the output to a log file for troubleshooting purposes.

In this example we are collecting metrics every 15 minutes, however you can adjust the frequency according to your needs. Just be aware that as the number of routers increases, your collection time will also increase. Requesting stats faster than time required to collect them can cause requests to start backing up and overload a system.

```
*/15 * * * * t128 /usr/local/bin/influx-importer extract
--config=/path/to/your/influx-importer.conf > /var/log/influx-importer.log
```

Download and install Grafana

Basic steps are listed below, but we recommend using the latest release and following the official install guide.

```
$ wget https://dl.grafana.com/oss/release/grafana-5.4.2-1.x86_64.rpm

$ sudo yum localinstall grafana-5.4.2-1.x86_64.rpm
```

Start Grafana

The grafana-server service should be enabled by default but not automatically started.

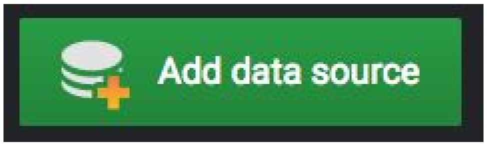
```
$ sudo systemctl start grafana-server
```

Open your browser and connect to your grafana server, default is port 3000

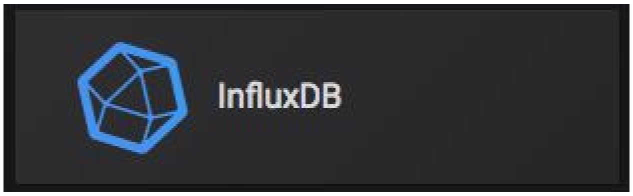
http://<server_address>:3000

Create datasource

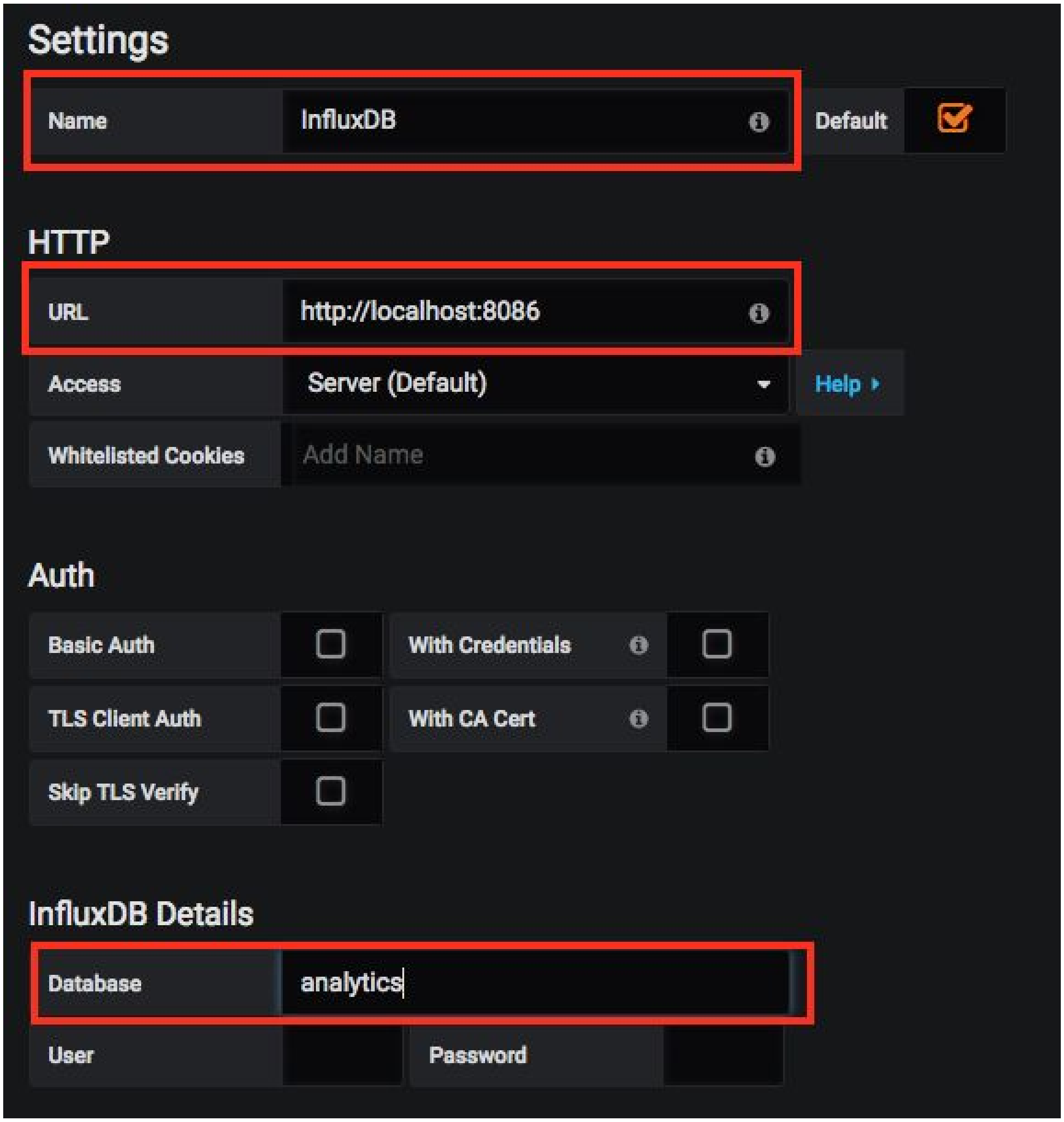
First we need to add our InfluxDB instance to as a datasource. Under Configuration select ‘Data Sources’, and then ‘Add Data Source’.



Select ‘InfluxDB’



The key fields to populate are the ‘Url’ and ‘Database’. Remember to set these fields according to your choices earlier in this guide.



Settings

Name InfluxDB
Default ☑

HTTP

URL http://localhost:8086
Access Server (Default)
Whitelisted Cookies Add Name

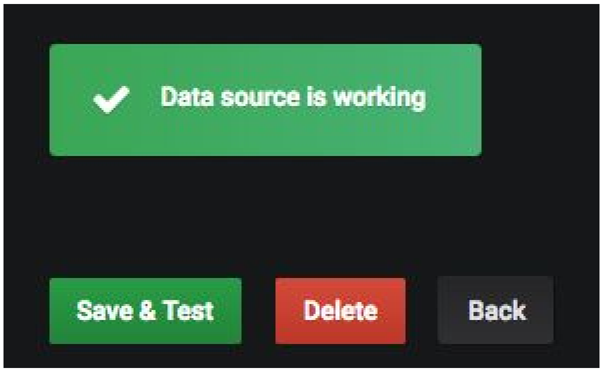
Auth

Basic Auth □
With Credentials □
TLS Client Auth □
With CA Cert □
Skip TLS Verify □

InfluxDB Details

Database analytics
User
Password

When you click ‘Save’ at the bottom, you should get a response in green. If not, double-check your settings.



✓ Data source is working

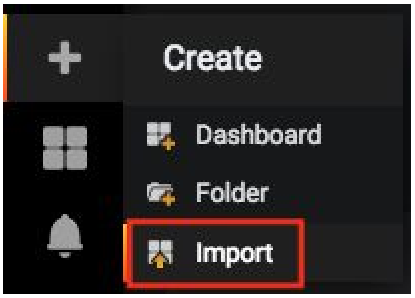
Import starter dashboard

We have created two starter dashboards as examples to build on.

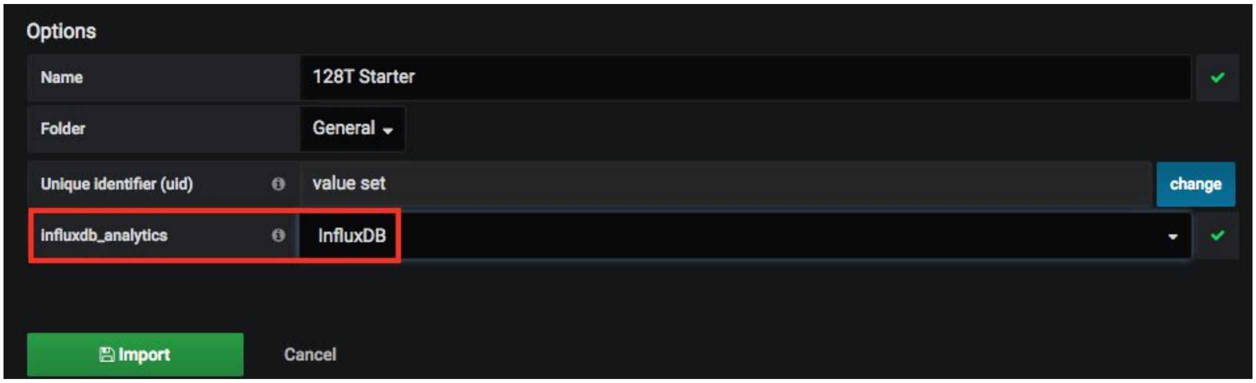
Download the ‘Grafana 128T Starter Dashboards’ from the community library.

https://community.128technology.com/communities/community-home/librarydocuments?communitykey=deb37e38-c10e-4d8f-b00b-d0aa8400e0ec&tab=librarydocuments&defaultview=folder&libraryfolderkey=7bf10d24-6730-4185-a25a-0d208420de02

Next, select ‘Import’ from the create menu.



Upload the json file, or paste the contents into the box and click 'Load', then make sure your datasource is selected and select ‘Import’



Options

Name	128T Starter
Folder	General
Unique identifier (uid)	value set
influxdb_analytics	InfluxDB

Done! You should now have two dashboards with rendered graphs!