

carbon, graphite and grafana

carbon

install carbon-cache service

```
apt-get install graphite-carbon
```

enable boot startup in */etc/default/graphite-carbon*

enable ENABLE_LOGROTATION in */etc/carbon/carbon.conf*

extend retentions

[/etc/carbon/storage-schemas.conf](#)

```
[carbon]
pattern = ^carbon\.
retentions = 60:90d

[default]
pattern = .*
retentions = 30s:7d,5m:30d,10m:1y,1d:10y
```

after changing **schemas.conf* older **wsp* has to be changed with *whisper-resize*

```
for i in `find /var/lib/graphite/whisper/ha/ -name "*wsp"`; do
  whisper-resize $i 30s:7d 5m:30d 10m:1y 1d:10y
done
```

graphite-web

web interface

```
apt-get install graphite-web
```

adjust SECRET_KEY (with random string) and TIME_ZONE in */etc/graphite/local_settings.py*

```
# choose user and password
graphite-manage syncdb
chown _graphite:_graphite /var/lib/graphite/graphite.db
#python /usr/lib/python2.7/dist-packages/graphite/manage.py syncdb
```

apache

```
sudo apt-get install apache2 libapache2-mod-wsgi
```

```
sudo cp /usr/share/graphite-web/apache2-graphite.conf /etc/apache2/sites-available  
sudo a2ensite apache2-graphite
```

nginx

```
apt install nginx uwsgi uwsgi-plugin-python
```

[/etc/uwsgi/apps-enabled/graphite.ini](#)

```
[uwsgi]  
  
vacuum = true  
master = true  
processes = 2  
pidfile = /tmp/uwsgi.pid  
socket = /tmp/uwsgi.sock  
chmod-socket = 666  
gid = _graphite  
uid = _graphite  
chdir = /usr/share/graphite-web  
wsgi-file = graphite.wsgi  
pymodule-alias =  
graphite.local_settings=/etc/graphite/local_settings.py  
buffer-size = 65536  
plugin = python
```

```
systemctl restart uwsgi
```

[/etc/nginx/sites-enabled/graphite](#)

```
upstream graphite {  
    server unix:///tmp/uwsgi.sock;  
}  
  
server {  
    listen 9002;  
    server_name localhost;  
  
    access_log /var/log/nginx/graphite-access.log;  
    error_log /var/log/nginx/graphite-error.log;
```

```
root /usr/share/graphite-web/static;

location / {
    add_header Access-Control-Allow-Origin "*";
    add_header Access-Control-Allow-Methods "GET, OPTIONS";
    add_header Access-Control-Allow-Headers "origin, authorization,
accept";

    uwsgi_pass graphite;
    include /etc/nginx/uwsgi_params;
}

location /media {
    # This makes static media available at the /media/ url. The
    # media will continue to be available during site downtime,
    # allowing you to use styles and images in your maintenance
page.
    alias /usr/lib/python2.7/dist-
packages/django/contrib/admin/media;
}
}
```

grafana

install

```
echo "deb https://packagecloud.io/grafana/stable/debian/ jessie main" >
/etc/apt/sources.list.d/grafana.list
curl https://packagecloud.io/gpg.key | sudo apt-key add -
apt update
apt install -y grafana
systemctl enable grafana-server
systemctl start grafana-server
```

Login to <http://localhost:3000> (admin/admin)

[add datasource](#)

The screenshot shows the 'Edit data source' configuration page in Grafana. At the top, there is a navigation bar with a gear icon and the text 'Data Sources'. Below this, the main heading is 'Edit data source'. There are two tabs: 'Config' (selected) and 'Dashboards'. The configuration section includes a table with the following fields:

Name	ha	<input type="checkbox"/>	Default	<input type="checkbox"/>
Type	Graphite			

Below the table is the 'Http settings' section with the following fields:

Url	http://localhost:9002	<input type="checkbox"/>		
Access	proxy	<input type="checkbox"/>		
Http Auth	Basic Auth	<input type="checkbox"/>	With Credentials	<input type="checkbox"/>

A green success message banner at the bottom of the configuration area reads: 'Success Data source is working'. At the very bottom, there are three buttons: 'Save & Test' (green), 'Delete' (orange), and 'Cancel' (grey).

plugins

Add plugin in /var/lib/grafana/plugins/ directory

piechart

```
cd /var/lib/grafana/plugins/  
git clone https://github.com/grafana/piechart-panel.git  
systemctl restart grafana-server.service
```

influxdb

```
curl -sL https://repos.influxdata.com/influxdb.key | apt-key add -
source /etc/lsb-release
echo "deb https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME}
stable" | tee -a /etc/apt/sources.list
apt update
apt install influxdb
```

enable admin service in /etc/influxdb/influxdb.conf and

```
systemctl restart influxdb
```

create database

```
root@graphite:~# influx
Connected to http://localhost:8086 version 1.2.1
InfluxDB shell version: 1.2.1
> create database captive
> CREATE USER "captive" WITH PASSWORD 'captive' WITH ALL PRIVILEGES
> show databases
name: databases
name
----
_internal
captive
>
```

test

```
curl -G http://carbon:8086/query --data-urlencode "q=SHOW DATABASES"
```

From:
<https://wiki.csgalileo.org/> - **Galileo Labs**

Permanent link:
<https://wiki.csgalileo.org/projects/internetofthings/graphite?rev=1491225331>

Last update: **2017/04/03 15:15**

