

Setup a Graphite instance

Fedora 4 comes with built-in profiling machinery that keeps track of how many times specific services have been requested, how long each request takes to be serviced, etc.

There are presently two ways of visualizing the output of this "metrics" collection.

- [Graphite graphing server](#)
- [JMX](#)

This guide is designed to help you install and configure a Graphite instance. The directions below are based on the [instructions provided by Graphite](#).

- [Prerequisites](#)
- [Graphite modules](#)
 - [Install whisper](#)
 - [Install Carbon](#)
 - [Install Graphite-web](#)
 - [Configure Graphite](#)
 - [Configure Apache](#)
 - [Setup Database](#)
 - [Create the database](#)
 - [Start Data Collector](#)
 - [Restart Apache](#)
 - [Connecting Fedora](#)
 - [Related articles](#)

Prerequisites

You will need the following software pre-installed:

1. Apache HTTPD
2. mod_wsgi
3. Cairo
4. Python (2.6, 2.7)
5. Django (1.6)
6. Various Python modules
 - a. pycairo
 - b. django-tagging
 - c. json or simplejson

On a Debian system, the following commands should install the above.

```
sudo apt-get update
sudo apt-get install git -y
sudo apt-get install apache2-mpm-prefork -y
sudo apt-get install libapache2-mod-wsgi -y
sudo apt-get install python-cairo -y
sudo apt-get install python2.7 -y
sudo apt-get install python-django -y
sudo apt-get install python-django-tagging -y
sudo apt-get install python-simplejson -y
```

Graphite modules

Setup Whisper, Carbon and Graphite-web

Clone the [GitHub repositories](#) (use either the 0.9.x [stable] branch or master [unstable/alpha] branch).

The source files found at <https://launchpad.net/graphite/+download/> are very out of date and do not support Django versions greater than 1.4.

On an Linux-based system, the following commands should get the correct branches of the three Graphite modules.

```
cd
git clone https://github.com/graphite-project/whisper.git
git clone https://github.com/graphite-project/carbon.git
git clone https://github.com/graphite-project/graphite-web.git
pushd carbon/
git checkout 0.9.x
popd;pushd graphite-web
git checkout 0.9.x
popd;pushd whisper
git checkout 0.9.x
popd
```

Your starting directory may look something like the following.

```
ls -l /home/ubuntu
drwxrwxr-x  9 ubuntu ubuntu 4096 Sep 18 20:59 carbon/
drwxrwxr-x 11 ubuntu ubuntu 4096 Sep 18 20:59 graphite-web/
drwxrwxr-x  4 ubuntu ubuntu 4096 Sep 18 20:59 whisper/
```

Install whisper

```
cd whisper
sudo python setup.py install
cd ..
```

Install Carbon

```
cd carbon
sudo python setup.py install
cd ..
```

This installs to **/opt/graphite** by default.

Install Graphite-web

Check for any fatal errors and resolve them, (ie. missing modules, libraries). There are 3 warnings that can be ignored (they are related to optional features).

```
cd graphite-web
sudo python check-dependencies.py
```

Install

```
sudo python setup.py install
cd ..
```

Configure Graphite

```
cd /opt/graphite
sudo chown -R ubuntu:ubuntu *
cd /opt/graphite/conf
cp carbon.conf.example carbon.conf
cp storage-schemas.conf.example storage-schemas.conf
cp graphite.wsgi.example graphite.wsgi
```

This sets up a default data retention period of gathering data every second and storing it for 1 day. You can [configure this](#) by editing the **storage-schemas.conf** file.

Configure Apache

There is an Apache Virtual Hosts file provided in the **graphite-web/examples** directory. You can use this to configure your Apache installation.

The following, minimal Apache vhost configuration file should be enough to get started

```
<IfModule !wsgi_module.c>
    LoadModule wsgi_module modules/mod_wsgi.so
</IfModule>

WSGISocketPrefix /var/run/wsgi

<VirtualHost *:80>
    ServerAdmin webmaster@myhost.edu

    ServerName graphite
    DocumentRoot "/opt/graphite/webapp"

    ErrorLog /opt/graphite/storage/log/webapp/error.log
    CustomLog /opt/graphite/storage/log/webapp/access.log common

    WSGIDaemonProcess graphite processes=5 threads=5 display-name='%{GROUP}' inactivity-timeout=120
    WSGIProcessGroup graphite
    WSGIApplicationGroup %{GLOBAL}
    WSGIImportScript /opt/graphite/conf/graphite.wsgi process-group=graphite application-group=%{GLOBAL}
    WSGIScriptAlias / /opt/graphite/conf/graphite.wsgi

    Alias /content/ /opt/graphite/webapp/content/
    <Location "/content/">
        SetHandler None
        Order deny,allow
        Allow from all
    </Location>
    <Directory /opt/graphite/conf/>
        Options All
        AllowOverride All
        Require all granted
    </Directory>
    <Directory /opt/graphite/webapp>
        Options All
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Setup Database

By default Graphite uses a sqlite3 database, but you can configure it to use a MySQL, Postgresql or Oracle database instead.

```
cd /opt/graphite/webapp/graphite
cp local_settings.py.example local_settings.py
```

Then edit local_settings.py, to uncomment:

- **SECRET_KEY**
- *Logging* section

```
SECRET_KEY = 'UNSAFE_DEFAULT <or your own value>'
...
# Logging
LOG_RENDERING_PERFORMANCE = True
LOG_CACHE_PERFORMANCE = True
LOG_METRIC_ACCESS = True
```

- Optional: If you are **NOT** using a sqlite3 database, find the **DATABASES** variable, un-comment it and change the **ENGINE** parameter and add required parameters (username, password, hostname, port).

Create the database

```
cd /opt/graphite/webapp/graphite
sudo python manage.py syncdb
```

- If you are using the default sqlite3 database, create an administrator <username> and <password> when prompted and then re-open **local_settings.py**, un-comment the **DATABASES** variable and add the administrator username/password to the **DATABASES** variable.

```
DATABASES = {
    'default': {
        'NAME': '/opt/graphite/storage/graphite.db',
        'ENGINE': 'django.db.backends.sqlite3',
        'USER': '<username>',
        'PASSWORD': '<password>',
        'HOST': '',
        'PORT': ''
    }
}
```

If you receive an error "*ImportError: cannot import name execute_manager*", you have an old version of graphite-web that requires Django 1.4. Get the latest changes from the GitHub repositories.

Set the permissions of the entire storage directory (default */opt/graphite/storage*) and all files to be owned by the webserver process.

```
cd /opt/graphite/storage
sudo chown -R www-data:www-data .
```

Start Data Collector

```
cd /opt/graphite/bin/
sudo -u www-data python carbon-cache.py start
> Starting carbon-cache (instance a)
```

This starts the listener on localhost:2003, this can be configured in */opt/graphite/conf/carbon.conf*.

If you receive the message:

 'WHISPER_FALLOCATE_CREATE is enabled but linking failed.'

This can be disabled by setting

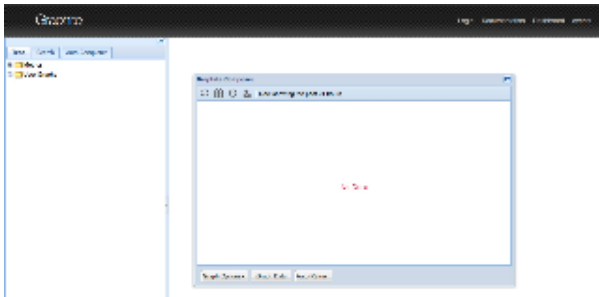
```
WHISPER_FALLOCATE_CREATE = False
```

in */opt/graphite/conf/carbon.conf* but this will not harm the running process and does not need to be disabled.

Restart Apache

```
sudo service apache2 restart
```

If all is well then browsing to your webserver's homepage should look something like this.



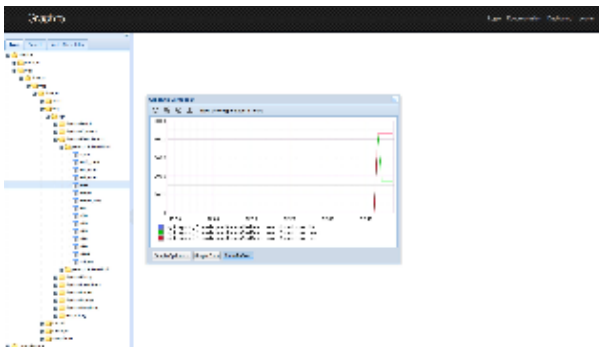
Connecting Fedora

To enable Metrics reporting to Graphite, activate the Spring profile **metrics.graphite**. The system properties **fcrepo.metrics.host** (defaults to *localhost*) and **fcrepo.metrics.port** (defaults to *2003*) can also be set.

When testing with Maven use:

```
MAVEN_OPTS="-Xmx512m -Dspring.profiles.active=metrics.graphite -Dfcrepo.metrics.host=<default-localhost> -Dfcrepo.metrics.port=<default-2003>" mvn jetty:run
```

Once Fedora 4 is connected to your Graphite server, the Graphite console should resemble the following:



Related articles

- [Fedora Repository Home](#)
- [Community Feedback/Concerns](#)