ericz82

[Logging] Setup Documentation

Introduction

This document will outline the steps taken to setup Statsd, Graphite, and Grafana.

Update your system:

```
sudo apt-get update && sudo apt-get upgrade
```

Install required packages:

```
export DEBIAN_PRIORITY=high
export DEBIAN_FRONTEND=noninteractive

sudo apt-get install build-essential graphite-web graphite-carbon
python-dev apache2 libapache2-mod-wsgi libpq-dev python-psycopg2
```

When asked to 'Remove database files when purging graphite-carbon?' Say no.

Configure Carbon

Open file `/etc/carbon/storage-schemas.conf`

This configuration file details retention rates for storing metrics. It matches metric paths to patterns, and tells pper what frequency and history of datapoints to store.
A given rule is made up of 3 lines:

- A name, specified inside square brackets.
- A regex, specified after “pattern=”
- A retention rate line, specified after “retentions=”

The retentions line can specify multiple retentions. Each retention of `frequency:history` is separated by a comma.

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

[test]
pattern = ^test\.
retentions = 5s:3h,1m:1d

[default_1min_for_1day]
pattern = .*
retentions = 60s:1d
```

Enable Carbon's cache to run on boot in file `/etc/default/graphite-carbon`

```
CARBON_CACHE_ENABLED=true
```

To turn on log rotation, open `/etc/carbon/carbon.conf` and set

```
ENABLE_LOGROTATION = True
```

Optional: `/etc/carbon/storage-aggregation.conf` defines how to aggregate data to lower-precision retentions

- This file is optional. If it is not present, defaults will be used.
- There is no `retentions` line. Instead, there are `xFilesFactor` and/or `aggregationMethod` lines.
- `xFilesFactor` should be a floating point number between 0 and 1, and specifies what fraction of the previous retention level's slots must have non-null values in order to aggregate to a non-null value. The default is 0.5.
- `aggregationMethod` specifies the function used to aggregate values for the next retention level. Legal methods are `average` , `sum` , `min` , `max` , and `last` . The default is `average` .
- These are set at the time the first metric is sent.

Example:

```
[all_min]
pattern = \.min$
xFilesFactor = 0.1
aggregationMethod = min
```

For an example file, copy the following:

```
sudo cp /usr/share/doc/graphite-carbon/examples/storage-aggregation.conf.example /etc/carbon/storage-aggregation.conf
```

Start the Carbon cache service:

```
sudo service carbon-cache start
```

Install and Configure PostgreSQL

By default, graphite uses sqlite3 for its database. In our dockerized version, we stayed with sqlite3 due to difficulties with dockerizing PostgreSQL.

Install PostgreSQL

```
sudo apt-get install postgresql
```

Create a database user for Graphite:

```
su - postgres
createuser graphite --pwprompt
```

Create the databases `graphite` and `grafana` with the system's `graphite` user as the owner:

```
createdb -O graphite graphite
createdb -O graphite grafana
```

Switch back to `graphite` user:

```
su - graphite
```

Alternative Instructions:

```
sudo -u postgres psql
CREATE USER graphite WITH PASSWORD 'password';
CREATE DATABASE graphite WITH OWNER graphite;
CREATE DATABASE grafana WITH OWNER graphite;
\q
```

Configure Graphite Web Application

Open file `/etc/graphite/local_settings.py`

```
USE_REMOTE_USER_AUTHENTICATION = True
TIME_ZONE = 'Your/Timezone'
SECRET_KEY = 'somelonganduniquesecretstring'
```

Find timezone here

Update Graphite's `DATABASES` dictionary definition with the settings for the PostgreSQL database created earlier in file `/etc/graphite/local_settings.py`

```
DATABASES = {
'default': {
    'NAME': 'graphite',
    'ENGINE': 'django.db.backends.postgresql_psycopg2',
    'USER': 'graphite',
    'PASSWORD': 'password_you_chose_earlier_for_database',
    'HOST': '127.0.0.1',
    'PORT': ''
    }
}
```

Initialize the database with:

```
sudo graphite-manage syncdb
```

Then answer the prompts to create a superuser account which will be used to access Graphite's web interface.

Configure Apache for Graphite

Copy Graphite's Apache config template into Apache's `sites-available` directory:

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

Change Graphite's port from 80 to 8080 (port 80 will be used for Grafana later) in file `/etc/apache2/sites-available/apache2-graphite.conf`

```
<VirtualHost *:8080>
```

Make sure Apache is listening on port 8080. Add `Listen 8080` after `Listen 80` in `/etc/apache2/ports.conf`

```
Listen 80
Listen 8080
```

Disable the default Apache site to avoid conflicts:

```
sudo a2dissite 000-default
```

Enable Graphite’s virtual site:

```
sudo a2ensite apache2-graphite
```

Reload Apache to apply the changes:

```
sudo service apache2 reload
```

At this point, you should be able to view the graphite-web interface by going to `http://server_domain_name_or_IP:8080`

Install and Configure Grafana

Add Grafana’s repository to `sources.list`:

```
echo 'deb https://packagecloud.io/grafana/stable/debian/ wheezy main'
|  sudo tee -a /etc/apt/sources.list
```

Add the Package Cloud key to install signed packages:

```
curl https://packagecloud.io/gpg.key | sudo apt-key add -
```

Update apt and install Grafana:

```
sudo apt-get update && sudo apt-get install grafana
```

Configure Grafana to use the PostgreSQL database created earlier in file `/etc/grafana/grafana.ini`:

```
[database]
# Either "mysql", "postgres" or "sqlite3", it's your choice
type = postgres
```

```
host = 127.0.0.1:5432
name = grafana
user = graphite
password = graphiteuserpasswordyouchoseearlier
```

Also in `/etc/grafana/grafana.ini`, configure the `domain` and `root_url`, and set a strong admin password and secret key in file `/etc/grafana/grafana.ini:`

```
[server]
protocol = http
http_addr = 127.0.0.1
http_port = 3000
domain = example.com
enforce_domain = true
root_url = %(protocol)s://%(domain)s/

[security]
admin_user = admin
admin_password = SecureAdminPass
secret_key = somelongrandomstringkey
```

Enable proxy modules for Apache reverse proxying to work:

```
sudo a2enmod proxy proxy_http xml2enc
```

Create an Apache site configuration file to proxy requests to Grafana. Remember to change `example.com` to your own domain in file `/etc/apache2/sites-available/apache2-grafana.conf`:

```
<VirtualHost *:80>
        ProxyPreserveHost On
        ProxyPass / http://127.0.0.1:3000/
        ProxyPassReverse / http://127.0.0.1:3000/
        ServerName example.com
</VirtualHost>
```

Enable Grafana’s site configuration with:

```
sudo a2ensite apache2-grafana
```

Configure Grafana to run after boot and then start service:

```
sudo update-rc.d grafana-server defaults 95 10
sudo service grafana-server start
```

Restart Apache to load the new modules and configuration changes:

```
sudo service apache2 restart
```

At this point, you should be able to open your IP address or domain in a browser to see Grafana's login page.
(Note: If you have trouble logging in as admin, the username is 'admin' and the default password is 'admin').

Install and Configure Statsd

ACQUIRE THE COMPONENTS

Get more packages:

```
sudo apt-get install git nodejs devscripts debhelper
```

Create a build repository:

```
mkdir ~/build
cd ~/build
git clone https://github.com/etsy/statsd.git
```

BUILD AND INSTALL THE PACKAGE

```
cd statsd
dpkg-buildpackage
cd ..
```

(Note: If build dependencies are not satisfied, install the dependencies by 'sudo apt-get install __dependency_name__')
Stop carbon service before installing:

```
sudo service carbon-cache stop
```

Install package:

```
sudo dpkg -i statsd*.deb
```

Statsd starts automatically so we need to stop it before configuring

```
sudo service statsd stop
sudo service carbon-cache start
```

CONFIGURE STATSD

Open file `/etc/statsd/localConfig.js` and change it so it is the following:

```
{
  graphitePort: 2003
, graphiteHost: "localhost"
, port: 8125
, graphite: {
     legacyNamespace: false
  }
}
```

CREATE A STORAGE SCHEMA FOR STATSD

Open file `/etc/carbon/storage-schemas.conf` and add new rule:

```
[statsd]
pattern = ^stats.*
retentions = 10s:1d,1m:7d,10m:1y
```

CREATE A DATA AGGREGATION CONFIGURATION

Open file `/etc/carbon/storage-aggregation.conf` and change the file to be:

```
[min]
pattern = \.min$
xFilesFactor = 0.1
aggregationMethod = min

[max]
pattern = \.max$
xFilesFactor = 0.1
aggregationMethod = max

[count]
pattern = \.count$
xFilesFactor = 0
aggregationMethod = sum

[lower]
pattern = \.lower(_\d+)?$
xFilesFactor = 0.1
aggregationMethod = min

[upper]
pattern = \.upper(_\d+)?$
xFilesFactor = 0.1
aggregationMethod = max

[sum]
pattern = \.sum$
xFilesFactor = 0
aggregationMethod = sum
```

```
[gauges]
pattern = ^.*\.gauges\..*
xFilesFactor = 0
aggregationMethod = last

[default_average]
pattern = .*
xFilesFactor = 0.5
aggregationMethod = average
```

START THE SERVICES

```
sudo service carbon-cache stop      ## wait a few seconds here sudo
service carbon-cache start
sudo service statsd start
```