NetApp

Quick Start: Installing Graphite and Grafana

Christopher Madden, NetApp
7 September, 2015

Abstract

This guide has basic installation steps for the Open Source software Graphite and Grafana. These software are commonly used with OnCommand Performance Manager (OPM) and/or NetApp Harvest available on the ToolChest.

TABLE OF CONTENTS

1 Introduction

1.1 About Graphite

Graphite is an open source enterprise-scale monitoring tool that does two things:

- Store numeric time-series data
- Render graphs or API responses of this data on demand

Graphite consists of three main components, all written in Python:

1) **carbon** – a Twisted daemon that listens for time-series data
2) **whisper** - a simple database library for storing time-series data (similar in design to RRD)
3) **graphite webapp** - A Django webapp that renders graphs on-demand using Cairo and requires a webserver such as Apache or nginx

Graphite itself has no metrics collection engine but rather receives metrics sent to it. Graphite supports various techniques to scale up and out the three main components and is quite powerful; 3 million metrics per minute on a single physical server has been reported!

Metrics can be retained by Graphite for a configurable retention period and frequency. The Graphite webapp also supports dozens of functions that allow for easy manipulation of collected data into useful information.

1.2 About Grafana

Grafana is an open source feature rich metrics dashboard and graph editor for Graphite [and other metrics databases]. It provides a powerful and elegant way to create, share, and explore data and dashboards from your disparate metric databases. It supports a wide variety of graphing options for ultimate flexibility.

Grafana supports authenticated login and a basic role based access control implementation.

Grafana is deployed as a single software installation and includes a webserver and presentation logic and is written in Go and Javascript.

1.3 About OnCommand Performance Manager External Data Provider

Beginning with OnCommand Performance Manager 1.1 a feature to forward metrics to an ‘External Data Provider’ was introduced. This integration allows metrics to be forwarded to a Graphite installation for long-term retention and integration into other metrics portals.

1.4 About NetApp Harvest

NetApp Harvest is a flexible data collector available on the NetApp ToolChest that collects performance metrics from Clustered Data ONTAP, Data ONTAP 7-mode, and capacity metrics from OnCommand Unified Manager 6.x servers. The metrics are then forwarded to a Graphite installation for long-term retention and integration into other metrics portals.

2 Installation Preparation

With enough effort and expertise Graphite can be installed on pretty much any Linux flavor. While the overall steps are documented in the Graphite documentation the exact commands for a given Linux flavor and release are not.

To ease installation we have installed and documented the steps for the following Linux releases:

- Ubuntu 14.04
- Red Hat Enterprise Linux 6.6
- Red Hat Enterprise Linux 7.1

Our experience has shown that installation on Ubuntu, using the native package repositories, is an easy choice for a quick implementation at zero cost.

We also see that many enterprises have standardized on Red Hat Enterprise Linux (RHEL), and thus prefer to deploy on this Linux flavor. For RHEL we have documented instructions for both online (installation from internet archives) and offline (installation via copying files) scenarios.

Either way you go, the feature/function of the solution is the same so choose the Linux flavor and steps that best meet your needs.

2.1 System Requirements

Platform requirements vary based on the quantity of nodes being monitored, amount of instances monitored (i.e. # vols, luns, lifs, etc), and the frequency of monitoring. This document includes a basic installation capable of servicing approximately 75,000 metric updates per minute. A typical monitored node might submit anywhere from 500 to 5000 metrics per polling interval (1 or 5 min typically). If this is insufficient Graphite can be scaled in various ways to support millions of metrics updates per minute if needed. Discussion of scaling higher is however outside the scope of this document.

System requirements:

- Linux OS x86-64, one of:
  - Ubuntu 14.04 LTS
  - RHEL 6.6
  - RHEL 7.1
- CPU: 1 vCPU
- Memory: 1GB RAM
- Disk capacity: 2.5-5GB per monitored node
- Disk performance: 20-200 random IOPs per monitored node
  - Use NetApp storage (NFS, LUN, vmdk, vhd, vhdx) or local SSD

Additionally, software packages are required from:

- Ubuntu
  - Ubuntu package repository
  - Grafana package repository
- RHEL
  - Red Hat package repository (Install ISO or Red Hat package repository)
  - Python PIP package repository
  - Github
  - Grafana package repository

2.2 Protocol and Port Requirements

Various network protocols and port reachability is required between Graphite, Grafana, and the client web browser. Below are details of the required network connectivity. Firewalls must be configured to allow access. The port numbers used in the instruction steps are listed but any item with a ‘*’ can optionally be modified to other ports or protocol (HTTP to HTTPS) to match your environment standards.

Initiated by the data collector (OPM, NetApp Harvest):

Destination	Protocol	Port	Description
Graphite Server IP	Graphite plaintext protocol	2003/TCP*	Used to post metrics to the Graphite metrics DB

Initiated by the Grafana server (in this guide both Graphite and Grafana run on the same host):

Destination	Protocol	Port	Description
Graphite Server IP	HTTP*	81/TCP*	Uses the render and query APIs to fetch JSON formatted metrics

Initiated by the Client browser (in this guide both Graphite and Grafana run on the same host):

Destination	Protocol	Port	Description
Graphite Server IP	HTTP*	81/TCP*	Access the Graphite native web GUI
Grafana Server IP	HTTPS*	443/TCP*	Access the Grafana dashboard GUI

3 Installation on Ubuntu

To install on Ubuntu follow the steps in this chapter. For RHEL installation steps see the chapter Installation on RHEL.

3.1 Basic server setup

Perform a basic installation of Ubuntu Server 14.04 LTS. During installation be sure to set your static IP address and default gateway, DNS servers, and time zone.

Also ensure time is correct and will be synced with true time by either (a) using the hypervisor guest tools to sync the VM clock with the host clock that is sync'd to true time or (b) install and configure NTP software to have the VM sync with true time, and then disable clock sync in the hypervisor guest tools.

3.2 Install Graphite and Prerequisites

Graphite is very easy to install using native packages using the short installation instructions that follow. For additional details and discussion this blog post may be useful as it served as the basis for these instructions.

First, install all needed packages:

```
sudo apt-get update
sudo apt-get -y install apache2 libapache2-mod-wsgi graphite-web graphite-carbon
```

- Answer Y to install other prerequisite packages
- A question is asked if you want to remove your database files if purging graphite-carbon. Answer "No". If you uninstall later you can always remove the data directory (`/var/lib/graphite/whisper`) manually.

3.3 Configure Graphite

3.3.1 Configure Carbon cache

1. Edit the Carbon cache service startup file:

   ```
   sudo nano /etc/default/graphite-carbon
   ```

   a. Find the line with CARBON_CACHE_ENABLED and modify it from false to true to enable the service to auto start on boot:

   ```
   CARBON_CACHE_ENABLED=false

   change to

   CARBON_CACHE_ENABLED=true
   ```

   b. Save the file and exit to the command prompt.

2. Edit the Carbon cache service configuration file:

   ```
   sudo nano /etc/carbon/carbon.conf
   ```

   a. Find the line with ENABLE_LOGROTATION and modify it from false to true to enable logs to be rotated:

   ```
   ENABLE_LOGROTATION = False

   change to

   ENABLE_LOGROTATION = True
   ```

   b. Find the line with MAX_CREATES_PER_MINUTE and modify it from 50 to 600 so more new metric files can be created per minute speeding display of newly added monitored hosts:

```
MAX_CREATES_PER_MINUTE = 50

change to

MAX_CREATES_PER_MINUTE = 600
```

c. Save the file and exit to the command prompt.

3. Copy a default storage-aggregation.conf file in place:

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

4. Start the Carbon cache service:

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

3.3.2 Configure graphite-web

1. Edit the Graphite web app configuration file:

```
sudo nano /etc/graphite/local_settings.py
```

a. Find the line with SECRET_KEY, uncomment it, and fill the value with some long random set of characters:

```
#SECRET_KEY = 'UNSAFE_DEFAULT'

change to (but fill with your own random characters!)

SECRET_KEY = ‘KJKzzxcviouzxcvytwkdd94944asmdf9ads9ds9dspp83jlkjLKJL98798KLOIK’
```

b. Uncomment and populate the TIME_ZONE variable with the time zone for the Graphite UI (see list here, use TZ column values) :

```
#TIME_ZONE = 'America/Los_Angeles'

change to (but replace with your Timezone!)

TIME_ZONE = 'Europe/Amsterdam'
```

c. Save the file and exit to the command prompt.

2. Initialize the Graphite configuration DB:

```
sudo graphite-manage syncdb
```

a. You will be prompted to create a superuser account. Choose yes and enter credentials. These can be used in the Graphite interface to save graphs.

3. Set ownership for access by the _graphite user:

```
sudo chown _graphite:_graphite /var/lib/graphite/graphite.db
```

3.3.3 Configure Apache

1. Configure web sites:

```
sudo a2dissite 000-default
sudo cp /usr/share/graphite-web/apache2-graphite.conf /etc/apache2/sites-available
sudo a2ensite apache2-graphite
```

2. Open the ports configuration file in a text editor.

```
sudo nano /etc/apache2/ports.conf
```

a. Find the line with Listen and set it to a value of 81

```
Listen 80

change to

Listen 81
```

b. Save the file and exit to the command prompt.

3. Open the Graphite vhost http configuration file in a text editor.

```
sudo nano /etc/apache2/sites-available/apache2-graphite.conf
```

a. Find the line with <VirtualHost> and modify the line to listen on port 81 and then add the three Header set entries listed below.

```
<VirtualHost *:80>

change to

<VirtualHost *:81>
```

4. Reload Apache configuration:

```
sudo service apache2 reload
```

3.4 Install Grafana

Grafana is very easy to install by adding a custom repository. Below are short installation instructions. For additional details please see the Grafana project website instructions Installing on Debian / Ubuntu.

1. Edit the apt-get sources file:

```
sudo nano /etc/apt/sources.list
```

a. Add the line for the Grafana server repository to the end of the file:

```
deb https://packagecloud.io/grafana/stable/debian/ wheezy main
```

b. Save the file and exit to the command prompt.

2. Add the package cloud key to allow installation of signed packages:

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

3. Update Apt repositories and install Grafana:

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

3.5 Configure Grafana

1. To configure Grafana to run over HTTPS either provide certificate and key files from a trusted CA, or create your own self signed certificate. If creating a self-signed certificate a sample command syntax is shown. You will be prompted with questions which should be answered according to your environment:

```
cd /etc/grafana
sudo openssl req -x509 -newkey rsa:2048 -keyout cert.key -out cert.pem -days 3650 -nodes
```

2. Open the Grafana configuration file in a text editor.

```
sudo nano /etc/grafana/grafana.ini
```

   a. Find the line with ;protocol, remove the semi-colon, and set it to a value of https as shown below:

```
;protocol = http

change to

protocol = https
```

   b. Find the line with ;http_port, remove the semi-colon, and set it to a value of 443 as shown below:

```
;http_port = 3000

change to

http_port = 443
```

   c. Find the section with https certificate information and remove the semi-colons and set the values of the certificate and key files created earlier:

```
# https certs & key file
;cert_file =
;cert_key =

change to

# https certs & key file
cert_file = /etc/grafana/cert.pem
cert_key = /etc/grafana/cert.key
```

   d. Save the file and exit to the command prompt.

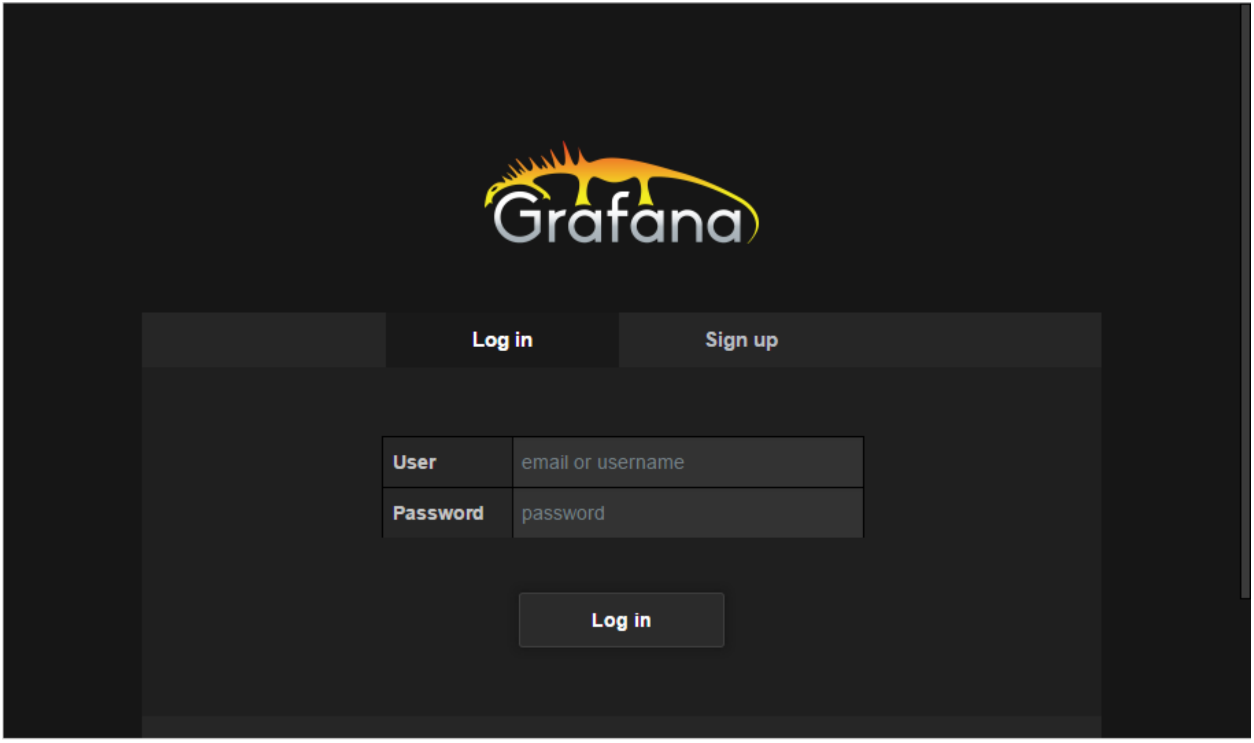
3. Set grafana-server to be able to run on any port (should you choose later to run on a privileged port):

```
sudo setcap 'cap_net_bind_service=+ep'  /usr/sbin/grafana-server
```

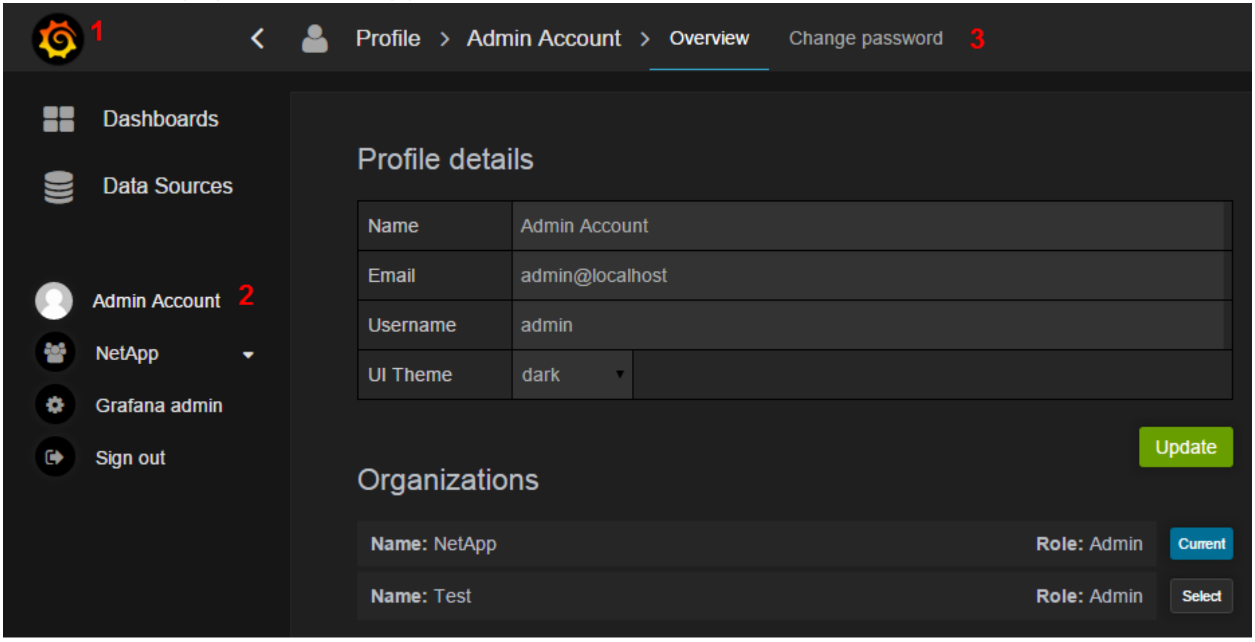
4. Set Grafana to start on boot and start it now:

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

5. Open your web browser to https://IP_ADDRESS_OR_HOSTNAME and you should see the Grafana login page. Login with the default username of ‘admin’ and password of ‘admin’:



6. Change the admin password by clicking the logo in the corner (1) and then the Admin account (2) and then Change password link (3):



Profile details

Name	Admin Account
Email	admin@localhost
Username	admin
UI Theme	dark

Organizations

Name: NetApp	**Role:** Admin
Name: Test	**Role:** Admin

Complete the current and new password and click the Change Password button. If successful a green box will be displayed that the password has been changed successfully.

4 Installation on RHEL

To install on RHEL follow the steps in this chapter. For Ubuntu installation steps see the chapter Installation on Ubuntu.

> **Note:** Execute all steps in this chapter as the root user. Login as root or use “`sudo -i`" to acquire root privileges

4.1 Basic server setup

Perform a basic installation of RHEL 6 or RHEL 7. During installation be sure to set your static IP address and default gateway, DNS servers, and timezone.

Also ensure time is correct and will be synced with true time by either (a) using the hypervisor guest tools to sync the VM clock with the host clock that is sync’d to true time or (b) install and configure NTP software to have the VM sync with true time, and then disable clock sync in the hypervisor guest tools.

4.1.1 Disable SELinux

1. Open the selinux configuration file in a text editor.

```
nano /etc/sysconfig/selinux
```

   a. Modify the value of SELINUX to be disabled:

```
SELINUX=enforcing

change to:

SELINUX=disabled
```

   b. Save the file and exit to the command prompt.

   c. Reboot the host:

```
reboot
```

4.1.2 Modify Firewall Rules

By default incoming traffic is blocked for remote access by the Linux firewall. The software installed in later steps will run on these ports so we must open them. Ports opened are 443 (Grafana), 81 (Graphite), and 2003 (Carbon cache metrics receiver).

- For RHEL 6 use the following commands:

```
iptables -I INPUT -p tcp --dport 443 -j ACCEPT
iptables -I INPUT -p tcp --dport 81 -j ACCEPT
iptables -I INPUT -p tcp --dport 2003 -j ACCEPT
service iptables save
```

- For RHEL 7 use the following commands:

```
firewall-cmd --permanent --add-port 443/tcp
firewall-cmd --permanent --add-port 81/tcp
firewall-cmd --permanent --add-port 2003/tcp
firewall-cmd --reload
```

4.2 Install Graphite and Prerequisites

As open source software Graphite is easiest to install when internet access is available. If internet access is not available an alternative set of offline installation instructions is also included.

Follow the appropriate section for your environment.

4.2.1 Online / Internet based installation

This section contains installation steps to be used if your Graphite host has a direct access to the Internet (i.e. via a transparent proxy).

If access is via a named proxy server configure the proxy first using environment variables which curl, yum, and git will use:

```
export http_proxy=http://proxy.myproxy.com:####
export https_proxy=https://proxy.myproxy.com:####
```

For PIP a proxy server can be configured using a modified pip install syntax like:

```
pip install --proxy="user:password@server:port" packagename
```

RPM packages

Install prerequisite and/or helpful system packages:

```
yum install -y python-devel mod_wsgi pycairo dejavu-sans-fonts gcc git pytz python-memcached nc
net-tools lsof unzip
```

Python packages

Install pip, and then use pip to install prerequisite Python packages:

```
cd /tmp
curl -O https://bootstrap.pypa.io/get-pip.py
python get-pip.py

pip install pyparsing
pip install 'Twisted<12.0'
pip install 'django<1.5'
pip install 'django-tagging==0.3.6'
```

Graphite components from source

Graphite and PIP have packages in EPEL but as of today (21-Aug-2015) each has issues installing on RHEL. For this reason we will install from source. Because all Graphite components are Python code installing from source is not much more difficult than from a package.

Execute the following commands:

```
cd /tmp
git clone https://github.com/graphite-project/graphite-web.git
git clone https://github.com/graphite-project/carbon.git
git clone https://github.com/graphite-project/whisper.git

cd whisper; git checkout 0.9.13-pre1; python setup.py install
cd ../carbon; git checkout 0.9.13-pre1; python setup.py install
cd ../graphite-web; python check-dependencies.py; git checkout 0.9.13-pre1; python setup.py
install
```

Copy the configuration files from the example files:

```
cd /opt/graphite/conf
cp carbon.conf.example carbon.conf
cp graphite.wsgi.example graphite.wsgi
cp storage-schemas.conf.example storage-schemas.conf
cp storage-aggregation.conf.example storage-aggregation.conf

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

cp /opt/graphite/examples/example-graphite-vhost.conf /etc/httpd/conf.d/graphite-vhost.conf
```

4.2.2 Offline / Jump host based installation

This section contains installation steps to be used if your Graphite host does not have direct access to the Internet. The steps require an Internet connected host to download files from the Internet source, but thereafter they can be copied to the Graphite host using other techniques (LAN, USB stick, ISO, etc).

RPM packages

To install required packages from the RHEL installation ISO use the following steps:

1. Mount the ISO at /media

```
mount /dev/cdrom /media
```

2. Add a yum repository by creating a new repo file with a text editor :

```
nano /etc/yum.repos.d/local.repo
```

   a. Paste the following contents into the file:

```
[LocalRepo]
name=Local Repository
baseurl=file:///media
enabled=1
gpgcheck=0
```

3. Install prerequisite and/or helpful system packages:

```
yum install -y python-devel mod_wsgi pycairo dejavu-sans-fonts gcc git pytz python-
memcached nc net-tools lsof unzip
```

Python packages

1. Download the Linux format packages (.tar.gz, .tar.bz2) manually from your internet connected host

   a. Manually from a web browser

      i. https://pypi.python.org/pypi/pip

      ii. https://pypi.python.org/pypi/argparse

      iii. https://pypi.python.org/pypi/wheel

      iv. https://pypi.python.org/pypi/setuptools

      v. https://bootstrap.pypa.io/get-pip.py

      vi. https://pypi.python.org/pypi/pyparsing/2.0.3

      vii. https://pypi.python.org/pypi/zope.interface/4.1.2

      viii. https://pypi.python.org/pypi/Twisted/11.1.0

      ix. https://pypi.python.org/pypi/Django/1.4.22

      x. https://pypi.python.org/pypi/django-tagging/0.3.6

   b. Or using Python and PIP from an internet connected host:

```
mkdir /tmp/pip_local
curl -o /tmp/pip_local/get-pip.py https://bootstrap.pypa.io/get-pip.py
pip install --download=/tmp/pip_local pip
pip install --download=/tmp/pip_local argparse
pip install --download=/tmp/pip_local wheel
pip install --download=/tmp/pip_local pyparsing
pip install --download=/tmp/pip_local 'Twisted<12.0'
pip install --download=/tmp/pip_local 'django<1.5’
pip install --download=/tmp/pip_local 'django-tagging==0.3.6'
```

2. Copy the downloaded files to the Graphite host `/tmp/pip_local` directory.
3. Install from the local archive

```
python /tmp/pip_local/get-pip.py --no-index --find-links=/tmp/pip_local
pip install pyparsing --no-index --find-links=/tmp/pip_local
pip install 'Twisted<12.0' --no-index --find-links=/tmp/pip_local
pip install 'django<1.5' --no-index --find-links=/tmp/pip_local
pip install 'django-tagging==0.3.6' --no-index --find-links=/tmp/pip_local
```

Graphite components

1. Download the Linux format packages (.tar.gz, .tar.bz2) manually from your internet connected host
    a. Manually from a web browser
        i. https://github.com/graphite-project/graphite-web/archive/0.9.13-pre1.zip
        ii. https://github.com/graphite-project/carbon/archive/0.9.13-pre1.zip
        iii. https://github.com/graphite-project/whisper/archive/0.9.13-pre1.zip
    b. Or using curl from an internet connected host:

```
mkdir /tmp/graphite_local

curl -L -o /tmp/graphite_local/graphite-web-0.9.13-pre1.zip
https://github.com/graphite-project/graphite-web/archive/0.9.13-pre1.zip

curl -L -o /tmp/graphite_local/carbon-0.9.13-pre1.zip https://github.com/graphite-
project/carbon/archive/0.9.13-pre1.zip

curl -L -o /tmp/graphite_local/whisper-0.9.13-pre1.zip
https://github.com/graphite-project/whisper/archive/0.9.13-pre1.zip
```

2. Copy the downloaded files to the Graphite host `/tmp/graphite_local` directory.
3. Install from the local archive (if your filenames are different because they were download

```
cd /tmp/graphite_local
unzip graphite-web-0.9.13-pre1.zip; unzip carbon-0.9.13-pre1.zip; unzip whisper-0.9.13-
pre1.zip
cd whisper-0.9.13-pre1; python setup.py install
cd ../carbon-0.9.13-pre1; python setup.py install
cd ../graphite-web-0.9.13-pre1; python check-dependencies.py; python setup.py install
```

4. Copy the configuration files from the example files:

```
cd /opt/graphite/conf
cp carbon.conf.example carbon.conf
cp graphite.wsgi.example graphite.wsgi
cp storage-schemas.conf.example storage-schemas.conf
cp storage-aggregation.conf.example storage-aggregation.conf

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

cp /opt/graphite/examples/example-graphite-vhost.conf /etc/httpd/conf.d/graphite-
vhost.conf
```

4.3 Configure Graphite

4.3.1 Configure Carbon cache

1. Open the carbon configuration file in a text editor.

```
nano /opt/graphite/conf/carbon.conf
```

    a. Find the line with MAX_CREATES_PER_MINUTE and modify it from 50 to 600 so more new metric files can be created per minute speeding display of newly added monitored hosts:

    ```
    MAX_CREATES_PER_MINUTE = 50

    change to

    MAX_CREATES_PER_MINUTE = 600
    ```

    b. Save the file and exit to the command prompt.

2. Set the carbon cache service to auto start and start it:

```
chmod 755 /etc/init.d/carbon-cache
chkconfig carbon-cache on
service carbon-cache start
```

4.3.2 Configure graphite-web

1. Open the webapp configuration file in a text editor:

```
nano /opt/graphite/webapp/graphite/local_settings.py
```

   a. Find the line with SECRET_KEY, uncomment it, and fill the value with some long random set of characters:

   ```
   #SECRET_KEY = 'UNSAFE_DEFAULT'

   change to (but fill with your own random characters!)

   SECRET_KEY = ‘KJKzzxcviouzxcvytwkdd94944asmdf9ads9ds9dspp83jlkjLKJL98798KLOIK’
   ```

   b. Uncomment and populate the TIME_ZONE variable with the timezone for the Graphite UI (see list here, use TZ column values) :

   ```
   #TIME_ZONE = 'America/Los_Angeles'

   change to (but replace with your Timezone!)

   TIME_ZONE = 'Europe/Amsterdam'
   ```

   c. Save the file and exit to the command prompt.

2. Initialize the Graphite configuration DB:

```
django-admin.py syncdb --pythonpath /opt/graphite/webapp --settings graphite.settings
```

   a. You will be prompted to create a superuser account. Choose yes and enter credentials. These can be used in the Graphite interface to save graphs.

3. Set ownership for access by the apache user:

```
chown -R apache:apache /opt/graphite/storage
```

4.3.3 Configure Apache

1. Open the http configuration file in a text editor.

```
nano /etc/httpd/conf/httpd.conf
```

    a. Find the line with Listen and set it to a value of 81

    ```
    Listen 80

    change to

    Listen 81
    ```

b. Find the line with ServerName, uncomment it, and set it to the hostname of the server:

```
#ServerName www.example.com:80

change to

ServerName graphite
```

c. Save the file and exit to the command prompt.

2. Open the Graphite vhost http configuration file in a text editor.

```
nano /etc/httpd/conf.d/graphite-vhost.conf
```

a. Find the line with <VirtualHost> and modify the line to listen on port 81 and then add the three Header set entries listed below.

```
<VirtualHost *:80>

change to

<VirtualHost *:81>
Header set Access-Control-Allow-Origin "*"
Header set Access-Control-Allow-Methods "GET, OPTIONS"
Header set Access-Control-Allow-Headers "origin, authorization, accept"
```

b. Find the line with <Directory> and replace that section (including directory path!) as shown below taking care that the text to replace differs between RHEL 6 and RHEL 7:

```
<Directory /opt/graphite/conf/>
 Order deny,allow
 Allow from all
<Directory>

change to for RHEL 6

<Directory /opt/graphite/>
 Options All
 AllowOverride All
</Directory>

Or, change to for RHEL 7

<Directory /opt/graphite/>
 Options All
 AllowOverride All
 Require all granted
</Directory>
```

c. Save the file and exit to the command prompt.

3. Set Apache to auto start at system boot and start it. On the first startup some structures are created in the SQLite db and it is possible that due to a race condition it will become locked for GUI access. For this reason we restart once after this initialization should have completed:

```
chkconfig httpd on
service httpd start;sleep 15; service httpd restart
```

4.4 Install Grafana

As open source software Graphite is easiest to install when internet access is available. If internet access is not available an alternative set of offline installation instructions is also included.

Follow the appropriate section for your environment.

4.4.1 Online / Internet based installation

Grafana is very easy to install by adding a custom repository. Below are short installation instructions. For additional details please see the Grafana project installation instructions Installing on RPM-based Linux.

1. Create a new yum repository for Grafana using a text editor.

```
nano /etc/yum.repos.d/grafana.repo
```

   a. Place the following text in the file:.

```
[grafana]
name=grafana
baseurl=https://packagecloud.io/grafana/stable/el/6/$basearch
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packagecloud.io/gpg.key https://grafanarel.s3.amazonaws.com/RPM-
GPG-KEY-grafana
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

   b. Save the file and exit to the command prompt.

2. Install Grafana using yum

```
yum install -y grafana
```

4.4.2 Offline / Jump host based installation

Grafana can also be easily downloaded and installed from a single RPM .

1. Download the latest RPM from http://grafana.org/download/ from your host with Internet access
2. Copy the downloaded file to the Graphite host `/tmp/` directory.
3. Install from the local archive replacing <BUILD> with your actual release level

```
rpm -i /tmp/grafana-<BUILD>.x86_64.rpm
```

4.5 Configure Grafana

1. To configure Grafana to run over HTTPS either provide certificate and key files from a trusted CA, or create your own self signed certificate. If creating a self-signed certificate a sample command syntax is shown. You will be prompted with questions which should be answered according to your environment:

```
cd /etc/grafana
openssl req -x509 -newkey rsa:2048 -keyout cert.key -out cert.pem -days 3650 -nodes
```

2. Open the Grafana configuration file in a text editor.

```
nano /etc/grafana/grafana.ini
```

   a. Find the line with ;protocol, remove the semi-colon, and set it to a value of https as shown below:

```
;protocol = http

change to

protocol = https
```

   b. Find the line with ;http_port, remove the semi-colon, and set it to a value of 443 as shown below:

```
;http_port = 3000
```

```
change to

http_port = 443
```

c. Find the section with https certificate information and remove the semi-colons and set the values of the certificate and key files created earlier:

```
# https certs & key file
;cert_file =
;cert_key =

change to

# https certs & key file
cert_file = /etc/grafana/cert.pem
cert_key = /etc/grafana/cert.key
```

d. Save the file and exit to the command prompt.

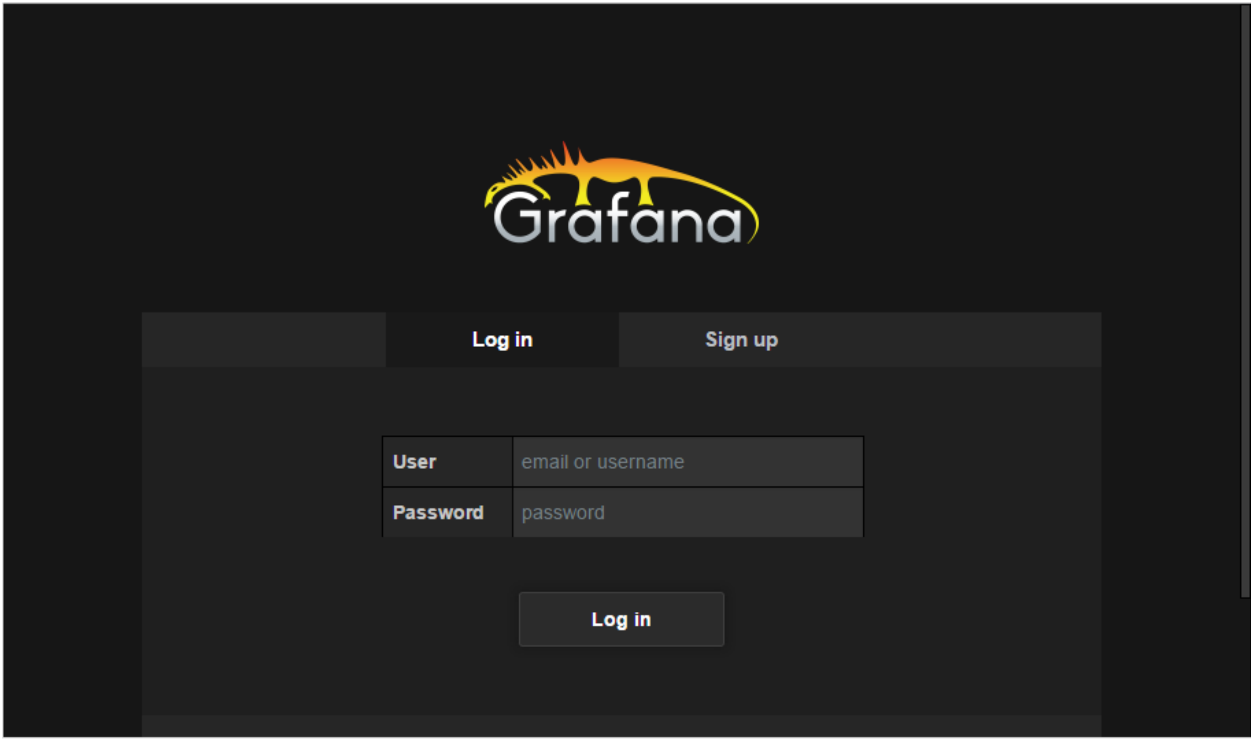
3. Set grafana-server to be able to run on a reserved port:

```
setcap 'cap_net_bind_service=+ep'  /usr/sbin/grafana-server
```

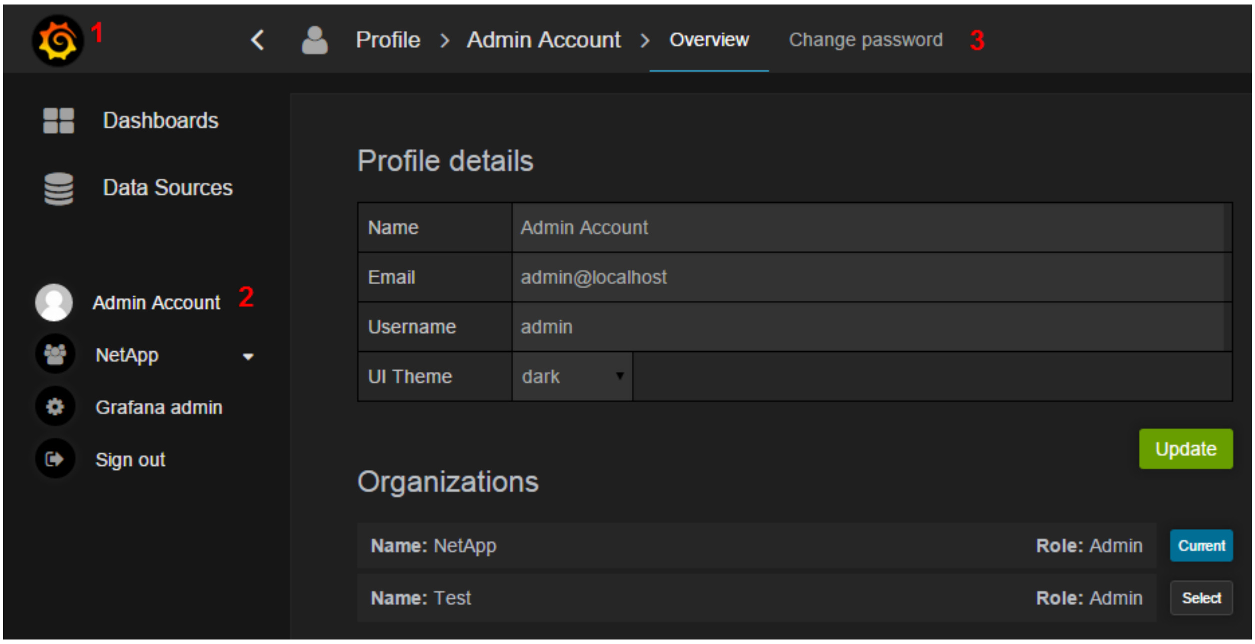
4. Set Grafana to auto start at system boot and start it:

```
chkconfig --add grafana-server
chkconfig grafana-server on
service grafana-server start
```

5. Open your web browser to https://IP_ADDRESS_OR_HOSTNAME and you should see the Grafana login page. Login with the default username of ‘admin’ and password of ‘admin’:



6. Change the admin password by clicking the logo in the corner (1) and then the Admin account (2) and then Change password link (3):



Profile details

Name	Admin Account
Email	admin@localhost
Username	admin
UI Theme	dark

Organizations

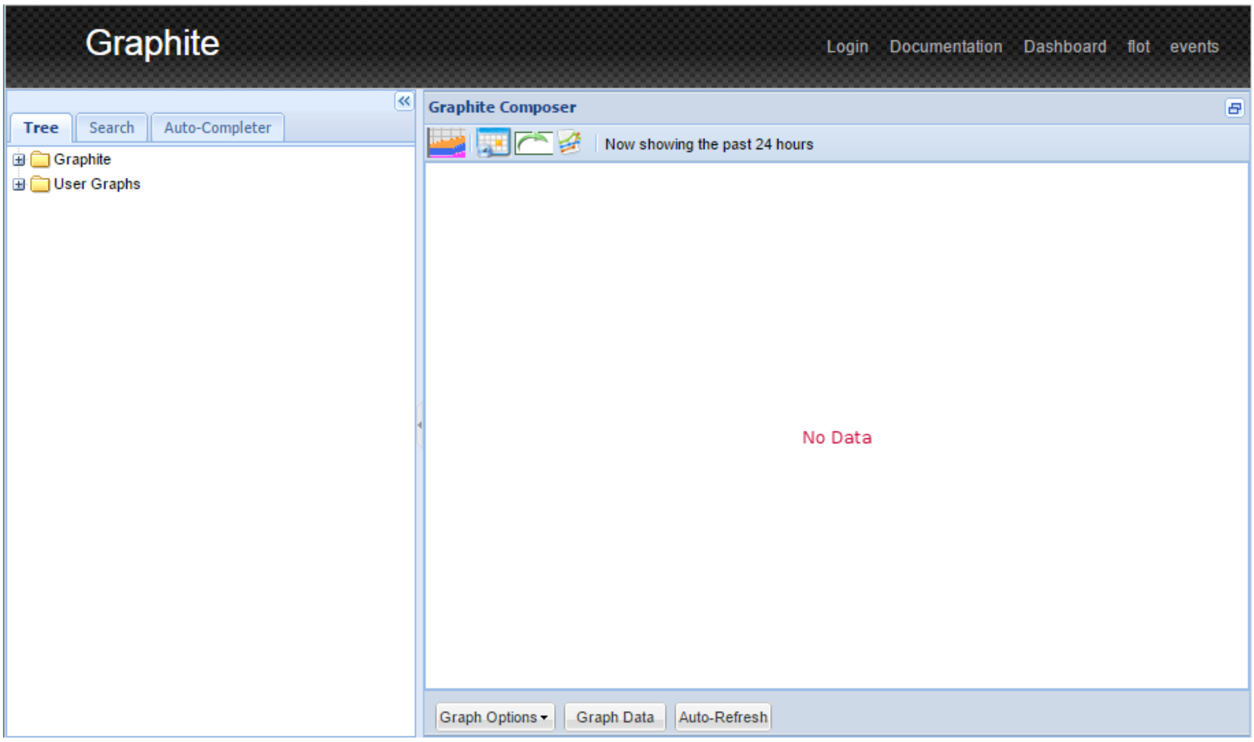
Name: NetApp	**Role:** Admin	Current
Name: Test	**Role:** Admin	Select

Complete the current and new password and click the Change Password button. If successful a green box will be displayed that the password has been changed successfully.

5 Verify Operations

5.1 Graphite (graphite-web and carbon-cache)

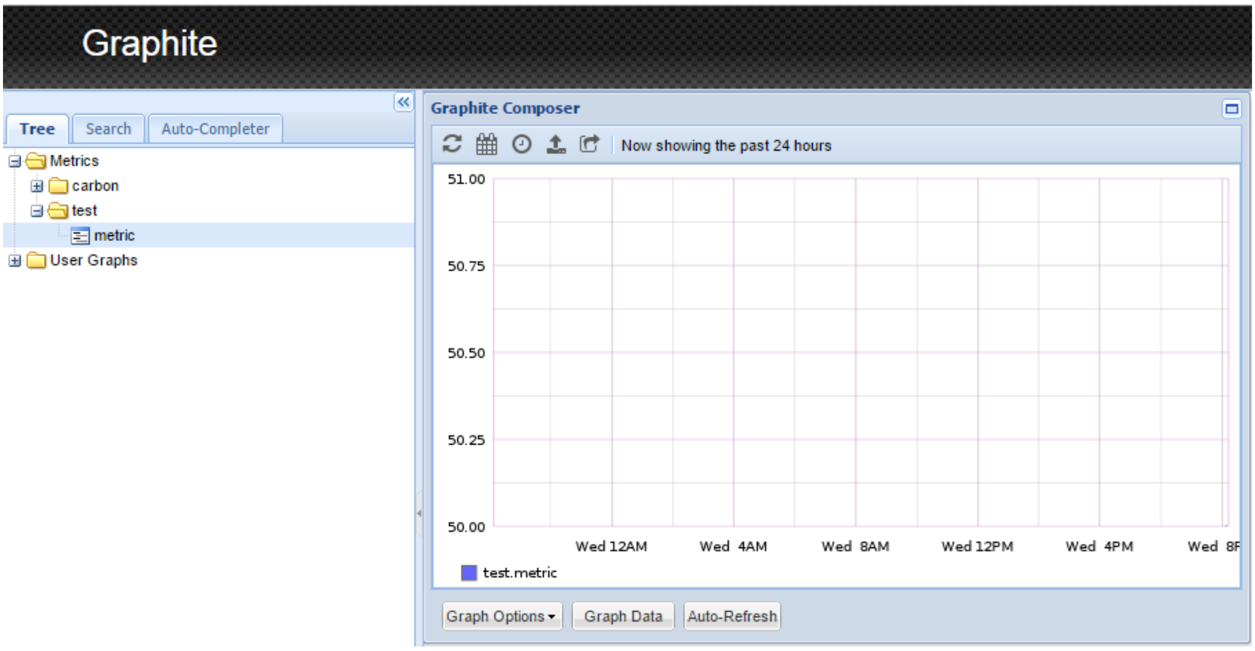
Open a web browser and browse to your Graphite server (http://IP_ADDRESS_OR_HOSTNAME:81) and you should see a screen that looks like this:



Next, send a metric to verify that carbon cache is running, accepts it, and the web graphite web UI can display it:

```
echo "test.metric 50 `date +%s`" | nc 127.0.0.1 2003
```

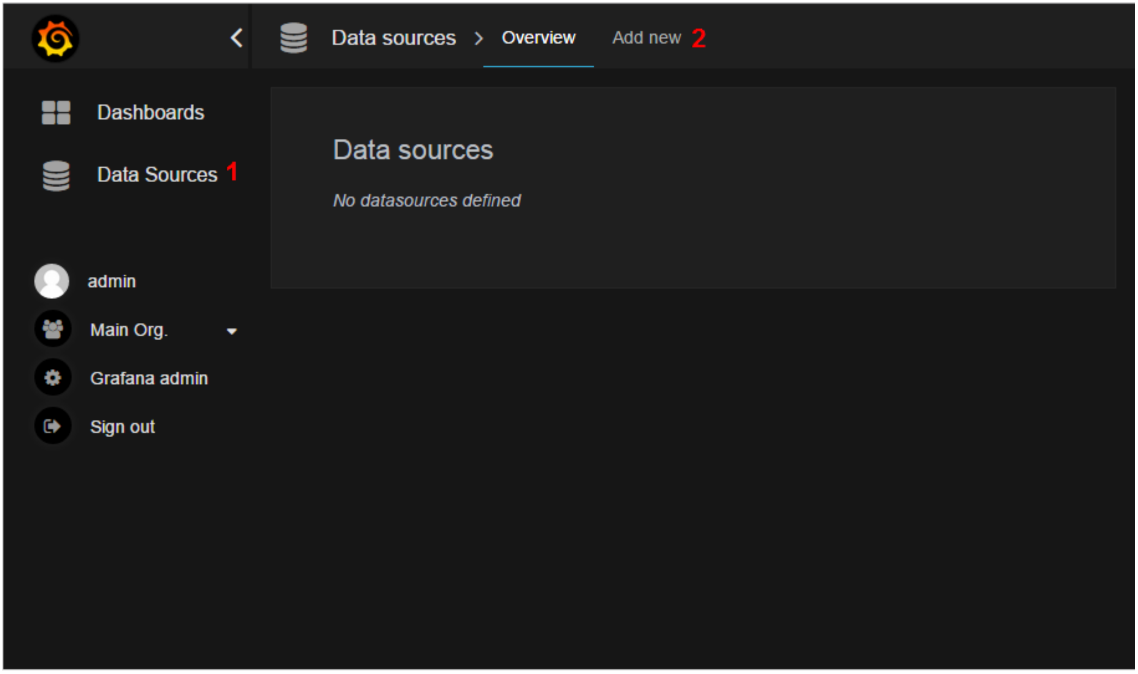
In your browser, reload the page (F5) and expand the Graphite folder and you should see a test folder and metric named metric:



5.2 Grafana

Open your web browser to https://IP_ADDRESS_OR_HOSTNAME and you should see the Grafana login page. Login with the default username of ‘admin’ and the password set in the configuration step earlier.

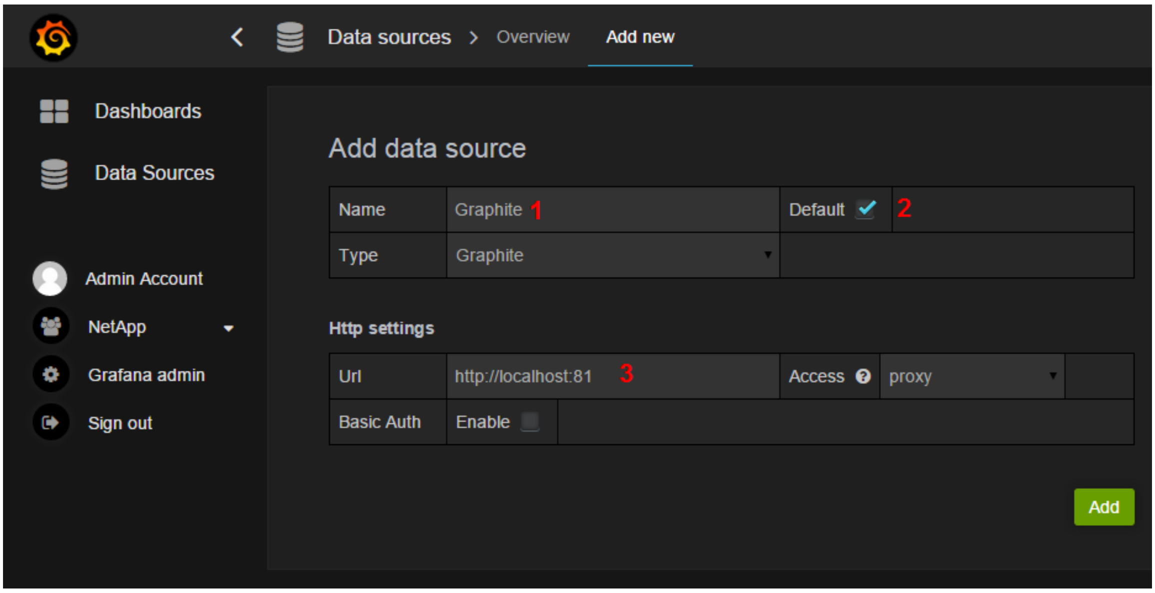
1. Add a datasouce by clicking Data Sources (1) and then Add New (2):



Data sources

No datasources defined

2. In the add data source dialog give a name for your Graphite server (1), check the Default checkbox (2), and enter the URL of the graphite server (3), then click Add:



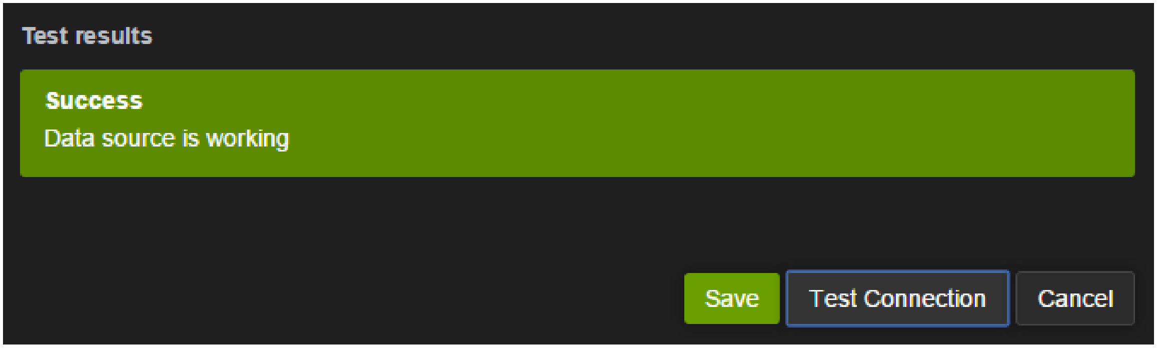
Add data source

Name	Graphite	Default ☑	
Type	Graphite		

Http settings

Url	http://localhost:81	Access	proxy
Basic Auth	Enable □		

3. Verify the data source is reachable by clicking the Test Connection button:



Test results

Success

Data source is working

You should see a Green Success message. Grafana is now installed and you can proceed to create or import dashboards.

6 Troubleshooting

Full troubleshooting procedures for the Graphite and Grafana solution is out of scope of this document but in the following subsections are some hints to get you started.

6.1 Start, stop, check status of Graphite and Grafana services

Check status of services:

Ubuntu	RHEL

`sudo service carbon-cache status`
`sudo service grafana-server status`
`sudo service apache2 status`
	`sudo service carbon-cache status`
	`sudo service grafana-server status`
	`sudo service httpd status`

Stop services:

Ubuntu	RHEL

`sudo service carbon-cache stop`
`sudo service grafana-server stop`
`sudo service apache2 stop`
	`sudo service carbon-cache stop`
	`sudo service grafana-server stop`
	`sudo service httpd stop`

To start services:

Ubuntu	RHEL

`sudo service carbon-cache start`
`sudo service grafana-server start`
`sudo service apache2 start`
	`sudo service carbon-cache start`
	`sudo service grafana-server start`
	`sudo service httpd start`

6.2 Whisper storage directory available space

When a metric is first received a corresponding whisper .wsp file is created. The file size of the .wsp will correspond to the configured frequency and retention; it is created ‘thick’ and will not grow over time. The storage directory is different on Ubuntu and RHEL. See the appropriate command to check disk space on the Whisper storage directory.

Check sufficient disk space is available on Ubuntu:

```
df -h /var/lib/graphite
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        29G  5.0G   23G  18% /
```

Check sufficient disk space is available on RHEL:

```
df -h /opt/graphite/storage
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_sdtgraphite2-lv_root
                       45G  2.2G   41G   6% /
```

6.3 Log file locations

Each component has one or more log files with details as found in the table below:

- **TIP**: use `ls -ltr` to reverse sort the files by modification timestamp; the last file in the output is a good place to start!

Component	Log directory	Log file description

Apache webserver	Ubuntu:
	`/var/log/apache2`
	RHEL:
	`/var/log/httpd`
		`error.log` shows overall webserver status messages
		`graphite-web_access.log` shows web page views. Use the referrer field to see which dashboards are most popular!
		`graphite-web_error.log` shows error message related to the virtual host serving the Graphite webapp

Graphite webapp	Ubuntu:
	`/var/log/graphite`
	RHEL:
	`/opt/graphite/storage/log/webapp`
		`exception.log` shows webapp exceptions
		`rendering.log` shows use of functions and response time of each. For a slow loading page this log might give an idea of which function(s) are slowest
		`metricaccess.log` shows metrics requested
		`cache.log` shows webapp lookups to carbon cache for datapoints
		`info.log` shows webapp query details (building index for metric searches)

Carbon cache metrics receiver	Ubuntu:
	`/var/log/carbon`
	RHEL:
	`/opt/graphite/storage/log/carbon-cache/carbon-cache-a`
		`listener.log` shows TCP sessions connected and disconnected from metrics sending clients
		`creates.log` shows metrics db file creation details such as matching schema and frequency and retention times used.

Grafana	Ubuntu and RHEL: `/var/log/grafana`	`grafana.log` shows accesses and errors

7 References

The following references were used in this paper:

- Ubuntu server download
  http://www.ubuntu.com/download/server
- Red Hat
  http://www.redhat.com/
- Graphite project site
  http://graphite.readthedocs.org/en/latest/
- Grafana project site
  http://grafana.org
- NetApp OnCommand Performance Manager
  http://mysupport.netapp.com/NOW/cgi-bin/software/
- NetApp Harvest Data Collector
  http://mysupport.netapp.com/tools/index.html

8 Version History

Version	Date	Document Version History
Version 1.0	July 2015	Initial Release
Version 1.1	July 2015	Addition of sizing information and minor updates to formatting
Version 1.2	August 2015	Added troubleshooting section
Version 1.3	August 2015	Added steps for Red Hat 6 and 7
Version 1.4	September 2015	Corrected service stop/start steps for Red Hat

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of this information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

NetApp®
www.netapp.com