Graphite für Icinga2 und Icingaweb2 installieren

Icinga2 eignet sich hervorragend zur Auswertung von Performancedaten der Plugins durch externe Tools. Hierzu bietet sich Graphite an. Die Unterstützung für Performancedaten und Graphite bringt Icinga2 gleich mit.

In diesem kurzen Howto installieren wir das icingaweb2-Modul für Graphite sowie andere Komponenten, die die grafische Darstellung der Daten in der GUI ermöglichen.

Vorrausetzungen

Es wird davon ausgegangen, dass auf dem Server eine funktionierende Icinga2 Installation mit Icingaweb2 läuft und funktioniert.

Das How-To basiert auf Debian 8 (Jessie).

Installation der Komponenten

Für die Nutzung von Graphite, die Darstellung der Graphen und die Einbindung in Icingaweb2 werden Graphite-Carbon, Graphite-Web und das Graphite-Modul für Icingaweb2 benötigt. Für Graphite-Web muss der Apache2 auch noch mit dem Python-WSGI-Modul erweitert werden:

```
apt-get install graphite-carbon graphite-web libapache2-mod-wsgi
```

Das Graphite-Modul für Icingaweb2 muss von Github geholt werden. Der Einfachheit halber wird diese einfach in den richtigen Pfad geklont. Dazu muss git (`apt-get install git`) auf dem Server vorhanden sein:

```
cd /usr/share/icingaweb2/modules/
git clone https://github.com/findmypast/icingaweb2-module-graphite.git graphite
```

Konfiguration

Zunächst müssen die Performancedaten und Graphite in Icinga2 aktiviert werden:

```
icinga2 feature enable perfdata graphite
```

Anschließend muss Icinga2 neu gestartet werden:

```
service icinga2 restart
```

Nach dem Neustart sollten unter `/var/spool/icinga2/perfdata/` bereits Daten zu sehen sein. Sie liegen dort im Format service-perfdata.< timestamp >. Dieses sind die Rohdaten welche Icinga liefert. Diese können nun durch externe Software verarbeitet werden.

```
ls -la /var/spool/icinga2/perfdata/
```

Als Nächstes muss eine lokale Config für das Graphite-Modul der GUI erstellt werden. An diesem Socket werden die Performance-Graphen abgeholt und anschließend in der GUI entsprechend dargestellt. Wenn hier etwas Falsches eingestellt wird, können die Bilder nicht angezeigt werden.

Nur absolute URLs verwenden. Also nicht `/render?` sondern `[hostname or ip]/render?`. Es muss jene URL angegeben werden unter welcher die normale Graphite-GUI zu erreichen ist.

```
mkdir -p /etc/icingaweb2/modules/graphite/
nano /etc/icingaweb2/modules/graphite/config.ini

[graphite]
metric_prefix = icinga
base_url = http://[hostname or IP]/render?
legacy_mode = true
```

Der Legacy-Mode muss gesetzt sein, sonst werden die URLs falsch gebildet. Mir ist derzeit nicht bekannt, warum und ab welcher Version von was der nicht mehr gebraucht wird.

Die Standard-Einstellungen in der `graphite.conf` können wir vorerst so belassen, es sei denn du möchtest den Carbon-Port anpassen. An diesem Port erwartet Graphite den Carbon-Listener.

```
nano /etc/icinga2/features-enabled/graphite.conf
```

Diesen Listen-Port von Carbon kann man selbstverständlich auch anpassen.

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

LINE_RECEIVER_INTERFACE = 0.0.0.0
LINE_RECEIVER_PORT = 2003
```

In der `carbon.conf` sind noch andere Dinge einzustellen, welche für uns aber vorerst nicht von Belangen sind.

Damit die Aggregation der Daten in Carbon funktioniert, müssen in der Datei `storage-schemas.conf` die Intervalle und Vorhaltezeiten eingetragen werden, nach denen Carbon arbeiten soll. Die Icinga2-Docs geben hier schon die benötigte Config vor.

```
nano /etc/carbon/storage-schemas.conf

[icinga_internals]
pattern = ^icinga\..*\.(max_check_attempts|reachable|current_attempt|execution_time|latency|state|state_type)
retentions = 5m:7d

[icinga_default]
pattern = ^icinga\.
retentions = 1m:2d,5m:10d,30m:90d,360m:4y
```

Unbedingt beachten, dass die Retention-Zeiten mit deinen Check-Intervallen übereinstimmen. Es müssen immer Intervalle und Vorhaltezeiten eingetragen werden (1m:2d = Checks, die jede Minute ausgeführt werden, werden bis zu zwei Tage gespeichert). Entsprechend wird mit den höheren Werten verfahren. Du solltest also darauf achten, dass deine Check-Intervalle entweder 1m, 5m, 30m oder 360m betragen, ansonsten musst du hier die Config entsprechend anpassen.

Für Graphite-Web solle unbedingt noch einen Key in den Einstellungen gesetzt werden. Dieser wird dazu verwendet Hashes zu bilden. Er ist also kein Passwort, welches man sich merken muss. Mindestens 20 Stellen am besten 30.

```
vim /etc/graphite/local_settings.py

SECRET_KEY = '[somelongarrayofrandomcharacters]'
```

Es fehlt noch die Webserver-Konfiguration, welche freundlicherweise bereits mitgeliefert wird. Unter `/usr/share/graphite-web/apache2-graphite.conf` findest du die Config für Apache. Man kann hier noch das Loglevel anpassen, ansonsten kann die Standard-Config beibehalten werden. Ich habe die Datei einfach zu Apache verlinkt, so ist nach einem Graphite-Update gleich die neue Config aktiv. Anschließend wird der Vhost aktiviert.

```
ln -s /usr/share/graphite-web/apache2-graphite.conf /etc/apache2/sites-available/
a2ensite apache2-graphite.conf
```

Anschließend muss noch mit Python synchronisiert werden. Du wirst einen User anlegen müssen. Mit diesem User kannst du dich am Graphite Webinterface anmelden (*auch wenn du das wahrscheinlich nicht benötigen wirst, da wir es nur für Icinga2 verwenden*).

```
python /usr/lib/python2.7/dist-packages/graphite/manage.py syncd
```

Bitte unbedingt darauf achten, der Maschine, auf der Icinga2 läuft, ausreichend RAM zur Verfügung zu stellen (>=1GB). Obwohl Icinga2 selbst nicht allzu performancehungrig ist, benötigt Graphite diesen um die gespeicherten Daten zu visualisieren. Ich hatte vorher Bild-Ladezeiten von ca. 20-30 Sekunden bei 256MB RAM.

Zum Abschluss müssen noch alle Dienste neu gestartet werden:

```
service icinga2 restart
service carbon-cache restart
service apache2 restart
```