Grafana - v8.3.2

Sommaire

Introduction

Grafana est une plateforme permettant de créer des tableaux de bord de visualisation pour les métriques. Dans ces tableaux de bord, la création de différents types de widget et de nombreuses options sont disponibles pour la visualisation des métriques.

Plus de détails sont disponibles sur le site officiel : https://grafana.com/

Les versions suivantes comportent des failles de sécurité critique (CVE-2021-43798). Shinken déconseille de les installer :

- 8.0.0 à 8.0.6
- 8.1.0 à 8.1.7
- 8.2.0 à 8.2.6
- 8.3.0

Installation

RHEL / CentOS 7 - RHEL / Alma / Rocky 8 - RHEL / Alma / Rocky 9

L'installation de Grafana sous RedHat et ses dérivés se fait via un paquet RPM. La version de Grafana testée avec Shinken Entreprise est la **v8.3.2-1**.

Pour l'installer, utiliser la commande suivante :

```
yum install --nogpgcheck https://dl.grafana.com/enterprise/release/grafana-enterprise-8.3.2-1.x86_64.rpm
```

Debian 13

L'installation de Grafana sous Debian se fait via un paquet DEB. La version de Grafana testée avec Shinken Entreprise est la **v8.3.2**.

Pour l'installer, utiliser les commandes suivantes :

```
apt install -y musl wget
wget -P /tmp/ https://dl.grafana.com/enterprise/release/grafana-enterprise_8.3.2_amd64.deb
apt install -y /tmp/grafana-enterprise_8.3.2_amd64.deb
rm -f /tmp/grafana-enterprise_8.3.2_amd64.deb
```

Activation du service

Une fois l'installation terminée, lancer Grafana et configurer son lancement automatique au démarrage du système avec les commandes suivantes :

```
systemctl enable grafana-server
systemctl start grafana-server
```

Après avoir installé puis lancé Grafana, l'interface sera accessible sur le **port 3000**. Les identifiants par défaut sont admin/admin

Connexion avec Graphite

Pour pouvoir récupérer les métriques générées par Shinken et enregistrées dans Graphite. Grafana doit avoir accès à Graphite. Pour cela, il faut ajouter dans Grafana une source de données Graphite.

Dans l'interface, allez dans la catégorie Data Sources , puis ajoutez une nouvelle source de type Graphite, qui sera paramétrée comme ci-dessous:

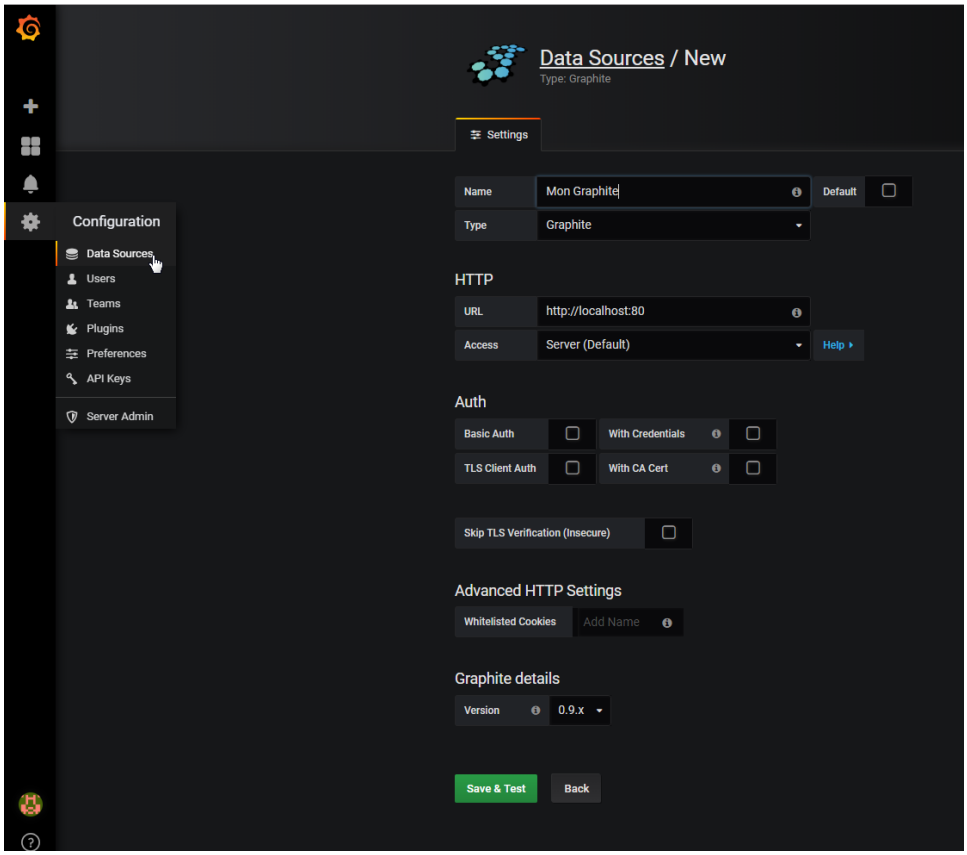
- **URL**: adresse du serveur hébergeant Graphite, sur le port 80. Par exemple:

  ```
  http://localhost:80
  ```

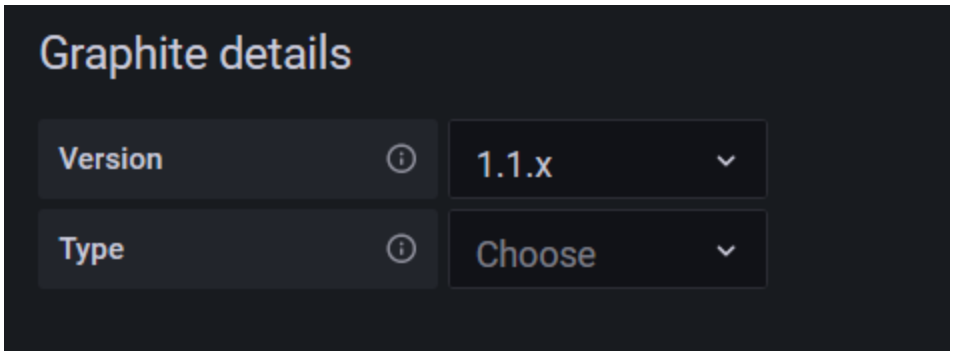
- **HTTPS**: si Graphite est configuré en HTTPS, alors il faut adapter l'URL

  ```
  https://localhost:443
  ```

- **Version**: La version de Graphite utilisée dépend de la version de Shinken Enterprise:
  - **Avant** la version V02.08.02-RC012, il faut choisir **0.9.x**



- **Après** la version V02.08.02-RC012, il faut choisir **1.1.x**



Grafana sur un serveur différent du serveur Graphite

Si Grafana est installé sur un serveur différent du serveur Graphite, il faudra effectuer une étape de configuration supplémentaire. Par défaut, Graphite autorise seulement les connexions locales.

RHEL / CentOS 7 - RHEL / Alma / Rocky 8 - RHEL / Alma / Rocky 9

Pour permettre à des serveurs distants d'accéder à ses données, il faut modifier le fichier de configuration Apache de Graphite **/etc/httpd/conf.d/graphite.conf** :

- Changer la ligne :

```
<VirtualHost 127.0.0.1:80>
```

- En :

```
<VirtualHost 0.0.0.0:80>
```

Cette ligne permet de spécifier les interfaces réseau de la machine sur lesquelles effectuer l'écoute. Spécifier 0.0.0.0 ou * permettent d'écouter sur toutes les interfaces réseau. On peut mettre une seule interface à la place en spécifiant l'IP de l'interface réseau concernée.

- Redémarrer Apache pour prendre en compte les modifications

```
systemctl restart httpd
```

Debian 13

Pour permettre à des serveurs distants d'accéder à ses données, il faut modifier le fichier de configuration Apache de Graphite **/etc/apache2/sites-available/graphite.conf** :

- Changer la ligne :

```
<VirtualHost 127.0.0.1:80>
```

- En :

```
<VirtualHost 0.0.0.0:80>
```

Cette ligne permet de spécifier les interfaces réseau de la machine sur lesquelles effectuer l'écoute. Spécifier 0.0.0.0 ou * permettent d'écouter sur toutes les interfaces réseau. On peut mettre une seule interface à la place en spécifiant l'IP de l'interface réseau concernée.

- Redémarrer Apache pour prendre en compte les modification

```
systemctl restart apache2
```

Gestion des certificats SSL/TLS par Grafana

Pour connecter Grafana à une source de données Graphite en HTTPS, il peut être nécessaire d'analyser le certificat de cette dernière. La commande suivante permet de récupérer les informations du certificat :

```
openssl x509 -in graphite.cert -text -noout
```

- Il est possible que la version de Grafana refuse les certificats qui utilisent uniquement le champ 'Common Name' (CN) pour identifier le propriétaire du certificat (*C'est le cas de la version de Grafana que Shinken recommande*).
  Dans ce cas, Grafana enregistrera l'erreur suivante dans le fichier **/var/log/grafana/grafana.log** : "proxy error: x509: certificate relies on legacy Common Name field, use SANs instead".
  - Pour résoudre ce problème, le certificat Graphite doit impérativement utiliser le champ 'Subject Alternative Name' (SAN) pour identifier le propriétaire du certificat.
- Si l'adresse IP ou le nom de domaine (DNS) de l'URL de la source de données n'est pas inclus dans la liste des machines certifiées par le certificat (*par exemple si l'URL de la source de données est localhost alors que le certificat a été émis pour le nom ou l'adresse IP de la machine*) Grafana refusera alors la connexion. Dans ce cas, Grafana enregistrera l'erreur suivante dans le fichier journal **/var/log/grafana /grafana.log** : "proxy error: x509: certificate is valid for XXXX, not XXXX".
  - Pour résoudre ce problème, il est nécessaire de modifier l'URL de la source Graphite dans Grafana pour qu'elle corresponde aux adresses utilisées par le certificat
- Il est enfin possible que la version de Grafana refuse le certificat de Graphite s'il est signé par une autorité inconnue.
  Dans ce cas, Grafana enregistrera l'erreur suivante dans le fichier **/var/log/grafana/grafana.log** : "proxy error: x509: certificate signed by unknown authority".
  On peut résoudre ce problème de deux manières :
  - Ajouter le certificat de Graphite ou celui de l'autorité de certification à la liste des certificats de confiance du système. Pour ce faire, on peut ajouter le certificat dans la chaîne de confiance du système, puis redémarrer Grafana.

RHEL / CentOS 7

```
cp your_certificate.crt /etc/pki/ca-trust/source/anchors/
update-ca-trust
systemctl restart grafana-server
```

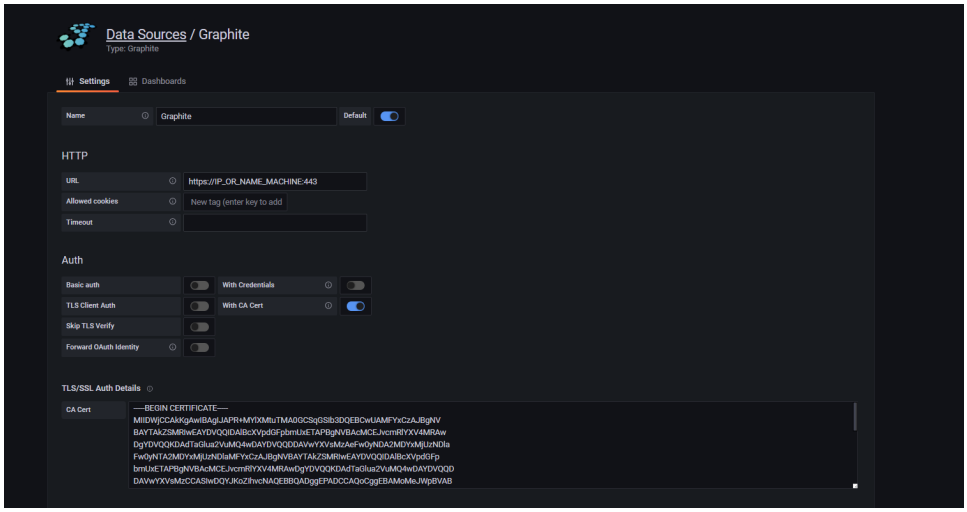
RHEL / Alma / Rocky 8

```
cp your_certificate.crt /etc/pki/ca-trust/source/anchors/
update-ca-trust
systemctl restart grafana-server
```

Debian 13

```
cp your_certificate.crt /usr/local/share/ca-certificates/
update-ca-certificates
systemctl restart grafana-server
```

- On peut également ajouter le certificat de la source de données Graphite directement depuis l'interface de Grafana. Cette méthode peut être plus pratique si on ne souhaite pas modifier les fichiers de configuration du système.



L'installation et la connexion de Grafana avec Graphite sont maintenant terminées.

On peut désormais créer des tableaux de bord, ajouter des utilisateurs et permettre la visualisation des métriques de Shinken.

Intégration dans Shinken

Par défaut, cette version n'accepte pas d'être intégrée dans un widget page web (*pour plus de détail sur la configuration du widget voir la page : Widget Page web*).

> Si l'authentification est activée dans Grafana et pour l'intégrer dans un widget web, il faut que Grafana soit accessible depuis la même adresse IP ou même le nom de domaine (*exemple : shinken-solutions.com est un nom de domaine*). Sans quoi le blocage CORS des navigateurs bloquera la connexion à Grafana.

Pour l'activer, il faut éditer le fichier de configuration Grafana **/etc/grafana/grafana.ini** dans la section "[security]":

```
allow_embedding = true
```

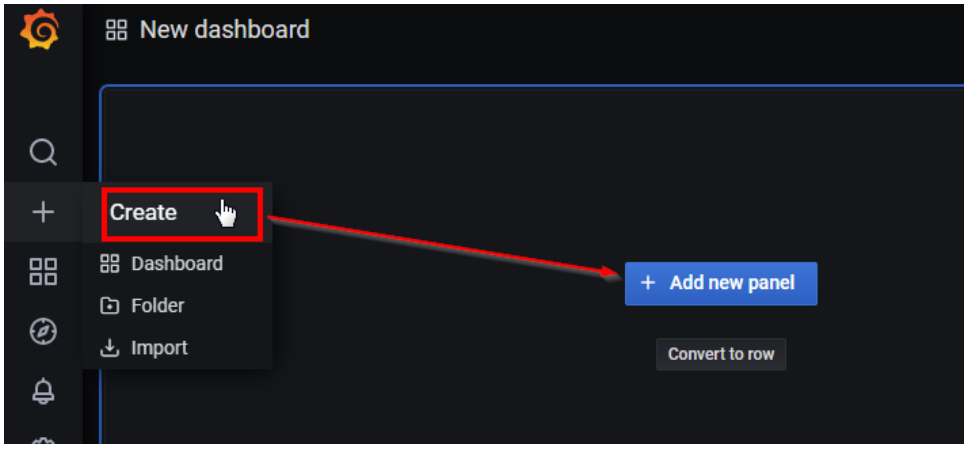
Puis redémarrer Grafana

```
systemctl restart grafana-server
```

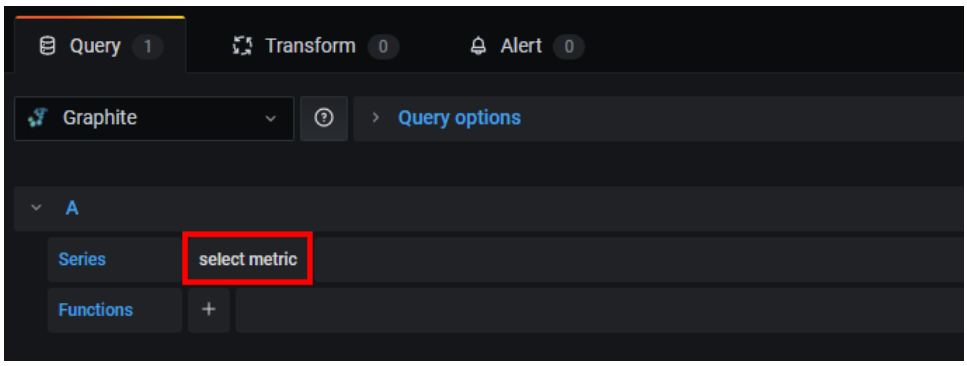
Créer un tableau de bord (*dashboard*)

Pour créer un tableau de bord,

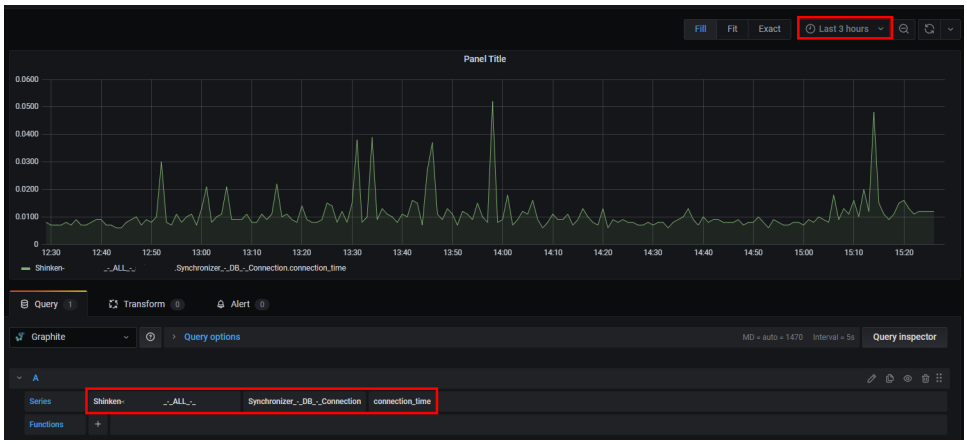
1. Cliquer sur le "+", "Create", puis "Add new panel".



2. Définir un nom dans le menu de droite.
3. Dans la partie basse de l'écran, il y a la composition de la requête.
   Pour générer un graphe, il faut cliquer sur "select metric", puis ajouter la métrique souhaitée.



4. Il faut sélectionner l'intervalle de visualisation.

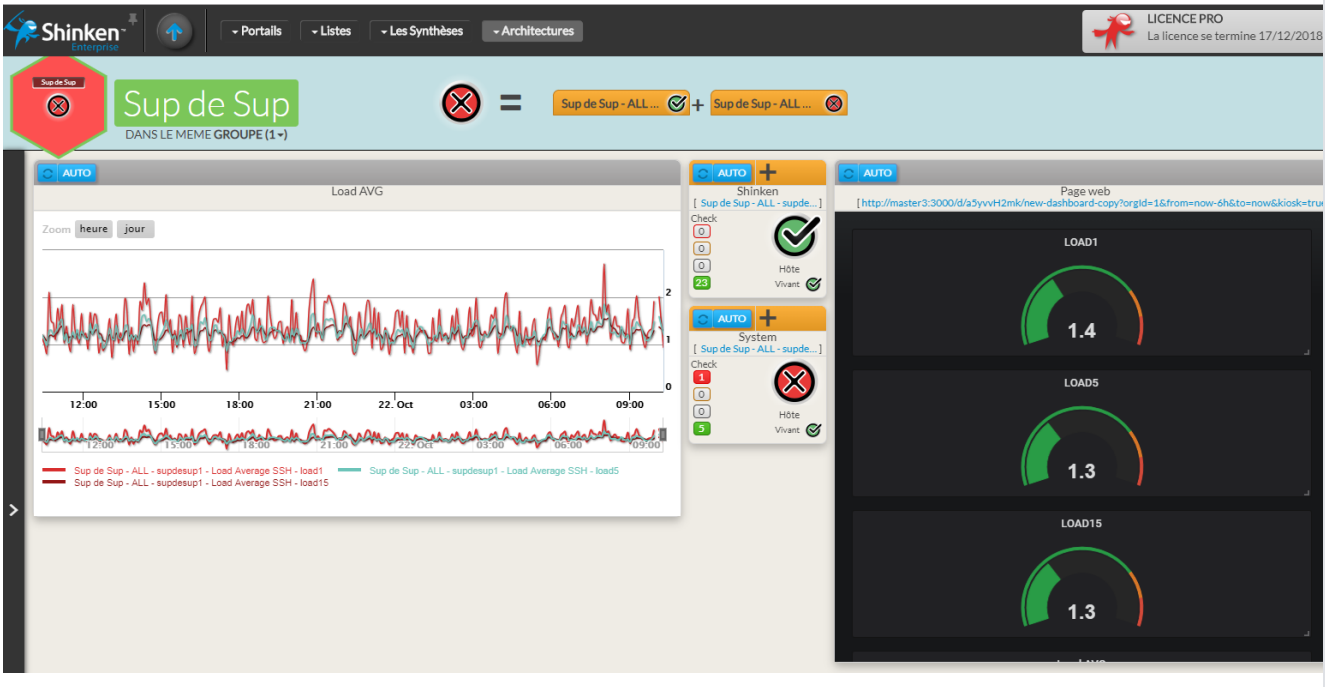


Une fois terminé, le bouton "Save" en haut à droite, va sauvegarde le panneau dans le tableau de bord. Le nom inscrit sera le nom du tableau de bord qui peut être composé de plusieurs panneaux.

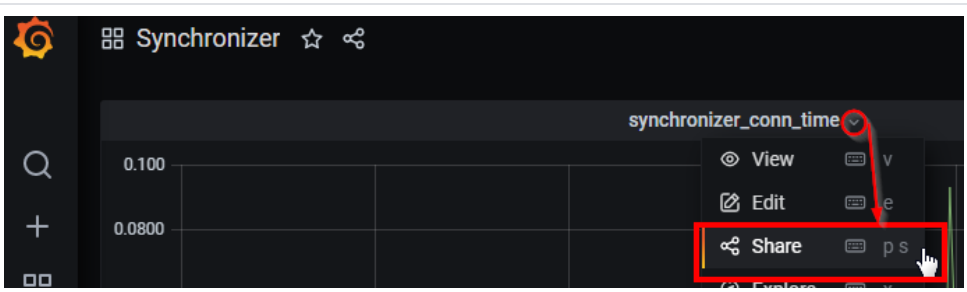
Récupération de l'URL à intégrer dans Shinken

Pour centraliser la visualisation des éléments supervisés par Shinken, il est possible d'intégrer les tableaux de bord Grafana dans un tableau de bord Shinken en utilisant le widget page web (*voir la page* Widget Page web).

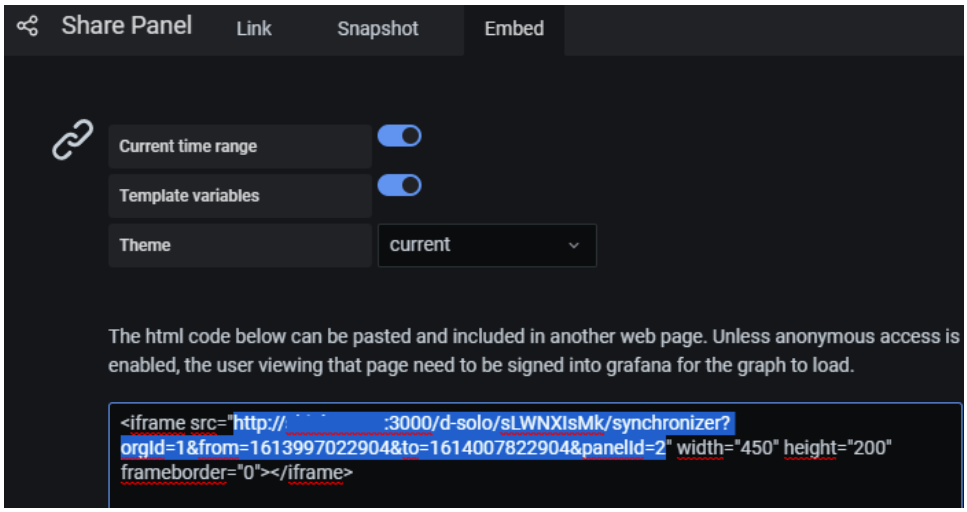
Par exemple :



1. Allez sur le panneau et cliquer sur share :



2. Dans l'onglet "Embed", sélectionner le line comme ci-dessous :

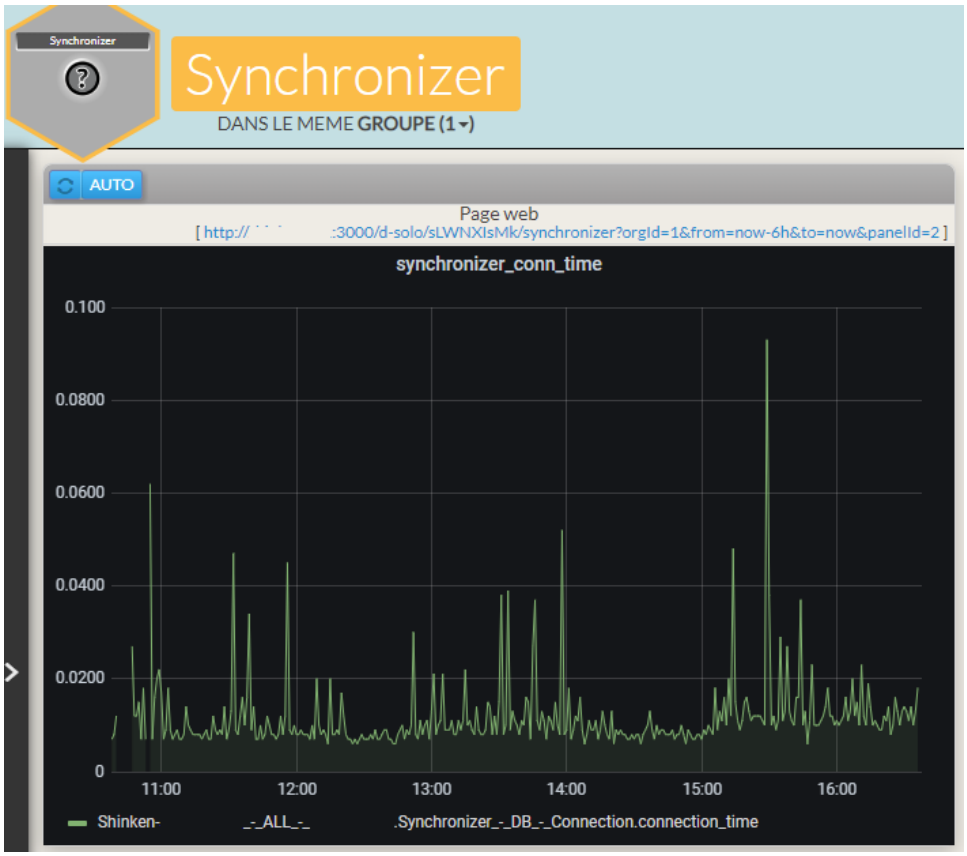


Pour obtenir un graphique évolutif en temps réel, soit :

a. Décocher l'option "Current time range".
b. Remplacer dans le lien toute la partie "from=XXXX&to=XXXX" par "from=now-6h&to=now" pour avoir un intervalle de 6h de visualisation par exemple. En faisant cela, l'intervalle est configuré au niveau du lien et non du tableau de bord.

3. Copier ce lien dans le widget Page web.

Exemple :




```
http://adresse_serveur:3000/d-solo/sLWNXIsMk/synchronizer?orgId=1&from=now-6h&to=now&panelId=2
```

Décomposition de l'adresse :

Paramètre	Définition
adresse_serveur	Adresse IP / Nom DNS du serveur
3000	Port par défaut de Grafana
d-solo	Permet de cacher les barres de navigation de Grafana pour n'afficher que les éléments visualisés
synchronizer	Nom du tableau de bord
from=now-6h	Intervalle d'affichage du graphes sur 6h
to=now	Intervalle jusqu'à maintenant
panelId=2	Numéro du panneau dans le tableau de bord

Passage en HTTPS

Dans le fichier de configuration de Grafana (***/etc/grafana/grafana.ini***),

Ajoutez :

```
protocol = https
cert_file = /chemin/vers/server.cert
cert_key = /chemin/vers/server.key
```

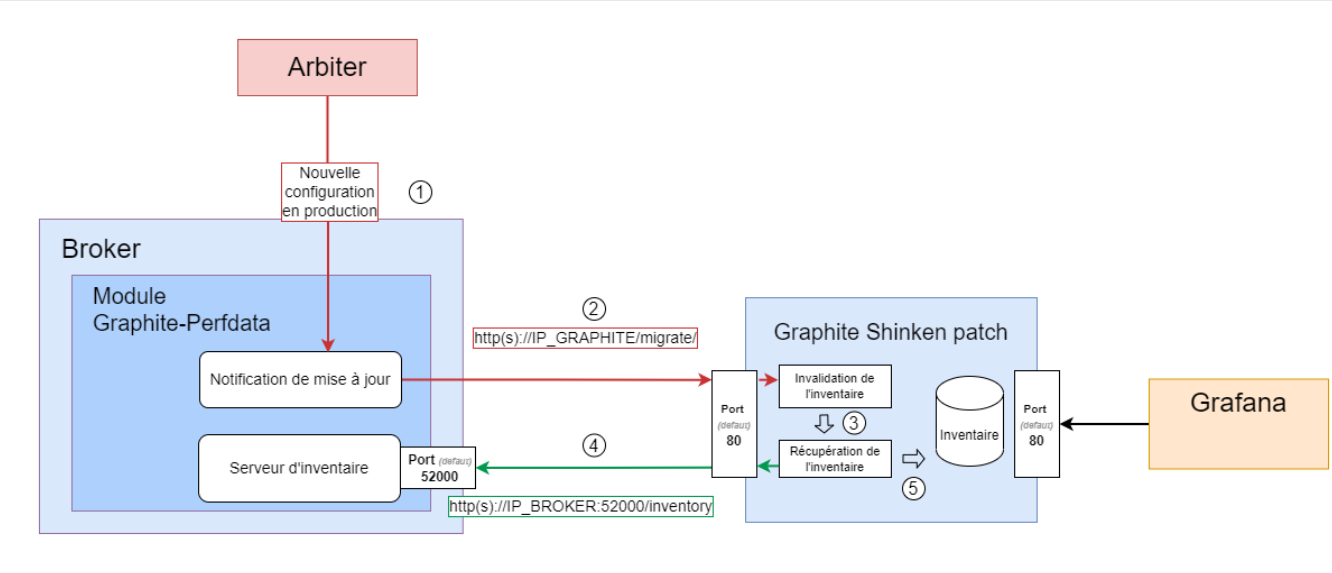
Puis redémarrer Grafana :

```
systemctl restart grafana-server
```

Lien vers le mapping nomuuid nécessaire pour Grafana, et suivi des requêtes

Les métriques stockées dans Graphite utilisent l'UUID des éléments comme clé.

- L'Interface de Visualisation de Shinken va utiliser les UUID pour ses échanges avec Graphite.
- Les outils externes qui échangent avec Graphite comme Grafana vont utiliser le nom des éléments.
    - Cela implique que Graphite possède une table de correspondance UUID nom pour répondre aux outils externes.



- Cette correspondance est fournie par le serveur d'inventaire (*ce serveur est un composant du module de métrologie du Broker de Shinken*).
    - La configuration de Graphite pour l'accès au serveur d'inventaire se fait via le fichier **/opt/graphite/conf/shinken_inventory.conf**
- Si la table de correspondance n'est plus à jour et qu'une requête par nom est demandée à Graphite, alors Graphite va faire la demande d'une nouvelle table de correspondance au serveur d'inventaire.
- Lorsqu'une nouvelle configuration de Shinken est déployée, le serveur d'inventaire envoie une notification à Graphite pour que l'inventaire soit modifié.
    - Afin que tous les processus de Graphite/Apache soient mis au courant, le fichier **/opt/graphite/storage/whisper/.cacheinvalidation** est mis à jour
        - Ce fichier ne doit pas être modifié.
        - En cas de problème, il est recréé, et le cache vidé.

> Lors de l'utilisation d'un cluster graphite, la configuration pour gérer la correspondance *UUID nom* doit être faite sur **tous** les serveurs faisant tourner un Carbon-Cache (*nœud de stockage des métriques*)

Les logs de chargement de la table de correspondance *UUID nom* sont disponibles dans le fichier **/opt/graphite/storage/log/webapp/info.log**

Pour suivre la mise à jour du mapping nom uuid et les requêtes pour un hôte particulier, il suffit de remplir le fichier **/opt/graphite/storage/whisper/.apache_graphite_host_filter_log** avec le nom de l'hôte. Les mises à jour dans la table de correspondance le concernant, ainsi que les requêtes de recherches de métriques seront toutes disponibles dans le fichier de log.

Paramètres de connexion aux serveurs d'inventaire

- Graphite se base sur les informations du fichier **/opt/graphite/conf/shinken_inventory.conf** pour aller chercher les informations qui lui permettront d'assurer la correspondance entre les noms et les UUID

Nom	Type	Unité	Défaut	Description
`ENABLE`	Booléen	---	**1**	Permets d'activer ou désactiver la recherche des correspondances entre les ID et les noms Valeurs possibles : • 1 (*Activé*) • 0 (*Désactivé*)
`URI`	Liste d'URI	---	**http://localhost:52000/inventory/**	URL séparées par des virgules. • Permet de contacter chacun des modules de métrologie qui fournit des métriques à ce serveur Graphite. • Si le serveur d'inventaire utilise SSL, il faudra utiliser https au lieu de http. **Exemple :** https://ip-broker01:52000/inventory/,https://ip-broker02:52000/inventory/,https://ip-broker03:52000/inventory/
`TIMEOUT`	Numérique	Seconde	**10**	Timeout général, utilisé pour les opérations bloquantes comme les tentatives de connexion à un serveur d'inventaire, par exemple.

Après tous changements du fichier de configuration, penser à redémarrer Apache pour que Graphite prenne les modifications en compte :

RHEL / CentOS 7

```
systemctl restart httpd
```

RHEL / Alma / Rocky 8

```
systemctl restart httpd
```

Debian 13

```
systemctl restart apache2
```

Autoriser les connexions aux serveurs d'inventaire

Configurer les modules de métrologie Graphite

Si le serveur Graphite et les Brokers avec les modules de métrologie Graphite sont sur des machines différentes, il faut configurer le serveur d'inventaire des modules de métrologie Graphite pour

- écouter sur les IP publiques de leur machine,

Pour cela, sur le serveur de l'Arbiter, éditer les fichiers de configuration des modules Graphite et décommenter la ligne du paramètre

/etc/shinken/modules/graphite.cfg

```
broker__module_graphite_perfdata__inventory_server__address  0.0.0.0
```

(*pour passer sa valeur de **127.0.0.1** à **0.0.0.0***)

- définir à qui envoyer les modifications de l'inventaire

/etc/shinken/modules/graphite.cfg

```
broker__module_graphite_perfdata__inventory_push__url  http://IP_GRAPHITE/migrate
```

Redémarrer l'Arbiter pour appliquer le changement de configuration

Ouvrir le port du serveur d'inventaire sur le firewall (*firewalld*)

Si le serveur d'un Broker qui fait tourner le module de métrologie Graphite dispose d'un firewall (*firewalld par défaut sur les systèmes Redhat et dérivés*), la commande suivante permet d'obtenir la liste des ports autorisés

```
firewall-cmd --list-ports
```

Exemple de résultat

```
80/tcp 7763/tcp 7765/tcp 7766/tcp 7767/tcp 7768/tcp 7769/tcp 7770/tcp 7771/tcp 7772/tcp 7773/tcp 7777/tcp
7780/tcp 50000/tcp
```

Dans cet exemple, le port 52000/tcp (*port par défaut du serveur d'inventaire du module de métrologie Graphite*) n'est pas listé, il est donc bloqué par défaut

Les commandes suivantes, à lancer sur le serveur du Broker, permettent d'autoriser les connexions :

```
firewall-cmd --add-port=52000/tcp
firewall-cmd --runtime-to-permanent
```

Compatibilité historique

Mode de fonctionnement avant la version de Shinken :

- V02.08.02-RC012
- ou V02.08.01.03

En cas d'impossibilité d'accès au serveur d'inventaire des modules de métrologie (*ports bloqués, paramètres par défaut incompatibles avec la configuration, la version installé est la version V02.08.02-RC012,...*), Graphite peut utiliser l'ancienne méthode que Shinken avait déployé pour fournir ces informations avec MongoDB.

- L'accès est configuré dans Graphite dans le fichier **/opt/graphite/conf/mongodb.conf**.

L'accès via MongoDB est déprécié et est voué à disparaître.

En effet, Graphite ne peut consulter qu'une seule base MongoDB pour obtenir les correspondances de noms, il est ainsi obligé d'utiliser la base centrale, qui est souvent aussi la plus chargée.

Configuration de l'accès à MongoDB

Connexion vers un cluster MongoDB

Pour se connecter à un cluster MongoDB, il faut configurer un démon mongos (*voir page Haute disponibilité de la base MongoDB (mise en place d'un cluster)*) sur les serveurs stockant les métriques, et s'y connecter pour accéder au cluster.

Connexion vers un démon mongod

Pour se connecter au serveur MongoDB, deux méthodes sont disponibles:

- **Connexion directe**: Par défaut, mais non sécurisée.
- **Tunnel SSH**: Shinken se connecte au serveur Mongo au travers d'un tunnel SSH pour plus de sécurité

Connexion directe au serveur MongoDB

Par défaut, Graphite se connecte de manière directe au serveur MongoDB pour y lire et écrire sa table de correspondance.

Dans la configuration de Graphite, on sait que la connexion se fait de manière directe lorsque le paramètre "USE_SSH_TUNNEL" est à 0.

Cette méthode de connexion a pour avantage d'être facile à configurer au niveau de Shinken. Par contre, elle oblige à permettre l'accès à la base MongoDB au monde extérieur, et donc s'exposer à des problèmes de sécurité.

- La sécurisation de la base MongoDB est bien sur toujours possible (*voir la page Sécurisation des connexions aux bases MongoDB*) mais bien plus complexe à mettre en place.
- La méthode de connexion par SSH est donc préférable pour des raisons pratiques et de sécurité.

Connexion par SSH au serveur MongoDB

Graphite peut également se connecter au serveur MongoDB par tunnel SSH (*pour des raisons de sécurité*).

- En effet, le paramétrage de MongoDB (*/etc/mongod.conf*) permet de définir sur quelle adresse ce dernier écoute les requêtes.
    - En n'autorisant seulement l'adresse 127.0.0.1, cela évite d'ouvrir la base au monde extérieur.
        - Dans la configuration du serveur MongoDB (*/etc/mongod.conf*), il faut que le paramètre "*bind_ip*" est positionné pour n'écouter que sur l'interface locale:
          `bind_ip= 127.0.0.1`

Comme toutes les connexions vers MongoDB, il est possible, et même recommandé, de sécuriser la communication via un tunnel chiffré SSH.

Le paramétrage de la connexion à MongoDB depuis Graphite, se fait en éditant les options suivantes (*dans **/opt/graphite/conf/mongodb.conf***):

Nom	Type	Unité	Défaut	Description
`URI`	Texte	---	**mongodb://localhost/?w=1&fsync=false**	URI du serveur MongoDB. L'adresse de la base MongoDB à utiliser est celle configurée dans le Module Graphite-Perfdata.
`DATABASE`	Texte	---	**shinken**	Nom de la base contenant les données d'inventaire sur le serveur MongoDB.
`DATABASE_USERNAME`	Texte	---		Le nom de l'utilisateur à utiliser pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé (*voir la page* MongoDB - activation de l'authentification par mot de passe)
`DATABASE_PASSWORD`	Texte	---		Le mot de passe de l'utilisateur utilisé pour l'authentification avec mot de passe à la base MongoDB. Utile uniquement si l'activation par mot de passe a été activé (*voir la page* MongoDB - activation de l'authentification par mot de passe)
`DATABASE_SSL`	Booléen	---	**0**	Active SSL/TLS pour les communications avec le serveur MongoDB. Valeurs possibles : • **0** (*désactivé*) • **1** (*activé*)
`DATABASE_SSL_CA_FILE`	Texte	---		Chemin vers le fichier de l'autorité de certification (*CA*) utilisé pour vérifier le certificat SSL de MongoDB.
`DATABASE_SSL_PEM_KEY_FILE`	Texte	---		Chemin vers le fichier contenant le certificat SSL du client.
`DATABASE_SSL_PEM_KEY_PASSWORD`	Texte	---		Mot de passe du certificat SSL du client.
`DATABASE_SSL_CRL_FILE`	Texte	---		Chemin vers le fichier CRL (*liste de révocation*) des certificats SSL à rejeter.

DATABASE_SSL_ALLOW_INVALID_HOSTNAMES		---	**0**	Accepter le certificat SSL de MongoDB même si le nom d'hôte ne correspond pas à celui du serveur. Valeurs possibles : • **0** (*désactivé*) • **1** (*activé*)
DATABASE_SSL_ALLOW_INVALID_CERTIFICATES		---	**0**	Accepter le certificat SSL de MongoDB même s'il est invalide, par exemple expiré. Valeurs possibles : • **0** (*désactivé*) • **1** (*activé*)
COLLECTION	Texte	---	**metrology_inventory**	Nom de la collection contenant les données d'inventaire.
USE_SSH_TUNNEL	Booléen	---	**0**	Paramètre permettant d'activer ou non l'utilisation d'un tunnel SSH. Valeurs possibles : • **0** (*désactivé*) • **1** (*activé*)
SSH_USER	Texte	---	**shinken**	Utilisateur sur le serveur MongoDB à contacter pour établir la connexion.
SSH_KEYFILE	Texte	---	**/opt/graphite/conf/id_rsa**	Chemin vers la clé SSH privée utilisée.
SSH_TUNNEL_TIMEOUT	Nombre	Seconde	**5**	Durée du timeout au bout duquel on détermine si l'établissement du tunnel a échoué.

Après tous changements du fichier de configuration, penser à redémarrer Apache pour que Graphite prenne les modifications en compte :

RHEL / CentOS 7

```
systemctl restart httpd
```

RHEL / Alma / Rocky 8

```
systemctl restart httpd
```

Debian 13

```
systemctl restart apache2
```

Graphite étant hébergé par le service apache, il n'a pas accès au répertoire **/var/lib/shinken** et il n'a donc pas accès à la clé SSH **/var/lib/shinken/.ssh/id_rsa**. C'est pour cette raison que la clé SSH utilisée pour le tunnel est situé dans **/opt/graphite/conf/id_rsa**.

Deux solutions sont disponibles :

- Générer une nouvelle clé SSH pour apache / graphite (*voir la page* Création automatique et gestion de la clé SSH de l'utilisateur shinken)
  - Lors de la génération de la clé, il est possible de spécifier directement le chemin suivant : **/opt/graphite/conf/id_rsa**
  - Il faudra ajouter cette nouvelle clé publique (*/opt/graphite/conf/id_rsa.pub*) sur le/les serveurs MongoDB (*dans le fichier ~shinken/.ssh/authorized_keys*)

    - Cette clé sera indépendante et non impactée par un changement de clé SSH sur l'utilisateur "shinken".

- Utiliser la clé SSH de l'utilisateur "shinken" présent sur le serveur.
    - La clé publique est sûrement déjà présente sur les serveurs MongoDB.
    - Il faut copier la clé privée et changer les droits pour l'utiliser et la maintenir à jour en cas de changement.

```
cp /var/lib/shinken/.ssh/id_rsa* /opt/graphite/conf/
chown apache:apache /opt/graphite/conf/id_rsa
```

Attention : un lien symbolique entre les deux fichiers ne fonctionnera pas, car l'utilisateur apache n'a pas les droits suffisants pour lire le fichier originel, et SSH refusera d'utiliser une clé dont les droits d'accès sont trop permissifs.

Mise à jour d'une version supérieure à 5.4.0

RHEL / CentOS 7 - RHEL / Alma / Rocky 8 - RHEL / Alma / Rocky 9

```
yum update https://dl.grafana.com/enterprise/release/grafana-enterprise-X.X.X.x86_64.rpm
```

Debian 13

```
apt install -y musl wget
wget -P /tmp/ https://dl.grafana.com/enterprise/release/grafana-enterprise_X.X.X_amd64.deb
apt install -y /tmp/grafana-enterprise_X.X.X_amd64.deb
rm -f /tmp/grafana-enterprise_X.X.X_amd64.deb
```

Autoriser l'intégration dans le widget "Page Web"

Pour la partie intégration avec le widget web de Shinken, si le paramètre "allow_embedding" ne se trouve pas dans le fichier **/etc/grafana/grafana.ini**, on peut l'ajouter dans la section "[security]" de ce fichier :

```
[security]
allow_embedding = true
```

Redémarrer grafana :

```
systemctl restart grafana-server
```

Une fois Grafana mis à jour, il suffit de rafraîchir la page et de s'authentifier

Authentification avec le widget Page Web

Pour fonctionner avec le widget Page web, Grafana va nécessiter un paramétrage spécifique (*voir la page* Widget Page web).

- L'authentification se fait avec des cookies.
- Dans les dernières versions de Chrome/Edge et Firefox, l'utilisation de requête "cross-site" par un cookie n'est plus autorisé.
- Cela signifie que si l'application n'est pas installée sur le même serveur que la WebUI de Shinken, l'authentification depuis le widget Page Web ne fonctionnera pas.

Pour palier à ce problème, on peut utiliser HAProxy pour récupérer le flux de ce site et simuler sa présence sur le serveur où est installé la WebUI.

Installation de HAProxy

Pour afficher les graphes d'un Grafana qui n'est pas installé sur le serveur où se trouve la WebUI Shinken. Il faut installer HAProxy sur chacun des serveurs où ces graphes doivent être affiché dans le widget Page Web.

RHEL / CentOS 7 - RHEL / Alma / Rocky 8 - RHEL / Alma / Rocky 9

```
yum install haproxy
```

Debian 13

```
apt install haproxy
```

Configuration de SELinux

> **Pour les systèmes utilisant SELinux (par défaut sous RedHat et ses dérivés)**
>
> Ajouter une exception dans SELinux pour HAProxy:
>
> ```
> setsebool -P haproxy_connect_any=1
> ```

Configuration de HAProxy

Modifier le fichier de configuration */etc/haproxy/haproxy.cfg* et ajouter à la fin du fichier la configuration de la redirection vers Grafana :

```
listen  grafana
  bind    SERVEUR_WEBUI:3000
  server    SERVEUR_GRAFANA SERVEUR_GRAFANA:3000
```

> Il faut remplacer **SERVEUR_WEBUI** par **0.0.0.0** ou alors l'adresse IP de la machine et **SERVEUR_GRAFANA** par le nom ou l'adresse IP du serveur

Démarrer HAProxy

```
systemctl enable haproxy
systemctl restart haproxy
```

On peut ajouter l'URL suivante dans le widget Page Web : http://SERVEUR_WEBUI:3000/XXXXXXX mais pas ~~http://SERVEUR_GRAFANA:3000/XXXXXXX.~~ La première authentification est nécessaire.

Log HAProxy

Par défaut HAProxy n'a pas de fichier de log. Pour en générer un il faut :

RHEL / CentOS 7 - RHEL / Alma / Rocky 8 - RHEL / Alma / Rocky 9

- Éditer le fichier */etc/rsyslog.conf* et dé-commenter les lignes suivantes :

  ```
  $ModLoad imudp
  $UDPServerRun 514
  ```

- Créer et ajouter dans le fichier */etc/rsyslog.d/haproxy.conf :*

  ```
  local2.* /var/log/haproxy.log
  ```

- Redémarrer rsyslog et haproxy :

  ```
  systemctl restart rsyslog
  systemctl restart haproxy
  ```

- Le fichier de log devrait être en place :

```
tail /var/log/haproxy.log
```

Debian 13

- Installer le paquet rsyslog

```
apt install -y rsyslog
```

- Éditer le fichier */etc/rsyslog.conf* et dé-commenter les lignes suivantes :

```
module(load="imudp")
input(type="imudp" port="514")
```

- HAProxy fournit par défaut un fichier de configuration pour rsyslog dans */etc/rsyslog.d*. Vérifier qu'il est bel et bien présent :

```
ls -l /etc/rsyslog.d/
```

    - Si ce n'est pas le cas, créer le fichier */etc/rsyslog.d/49-haproxy.conf* et ajouter ce contenu :

```
# Create an additional socket in haproxy's chroot in order to allow logging via
# /dev/log to chroot'ed HAProxy processes
$AddUnixListenSocket /var/lib/haproxy/dev/log

# Send HAProxy messages to a dedicated logfile
:programname, startswith, "haproxy" {
  /var/log/haproxy.log
  stop
}
```

- Redémarrer rsyslog et haproxy :

```
systemctl restart rsyslog
systemctl restart haproxy
```

- Le fichier de log devrait ête en place

```
tail /var/log/haproxy.log
```