Importer ses propres fichiers CFG

Sommaire

Contexte

Ce document décrit comment importer ses propres fichiers CFG à travers sa propre source.

> **Important**
>
> Attention, nous vous conseillons de créer votre propre source. En effet, la source Nagios (et Shinken) sont des sources qui sont placées dans le répertoire central Shinken (/etc/shinken) et peuvent être soumises à des modifications lors de mises à jour. Par contre, la création de nouvelles sources personnalisées se fera dans le répertoire dédié aux utilisateurs (/etc/shinken-user) et ce répertoire sera donc traité avec grande vigilance par le moteur de mise à jour, car incluant vos données utilisateur.

Voici les étapes à suivre afin de créer une source qui vous permettra d'importer vos propres CFG:

- Création d'une source
- Création du répertoire de données de la source
- Configuration du démon Synchronizer pour déclarer la source

Ce qui est déjà disponible dans l'installation de Shinken

Pour vous rendre la vie plus facile, quelques taches de configuration ont déjà été faites:

a. Le type de module "cfg-file-import" est déjà prêt à être utilisé,
b. La possibilité d'utiliser la source pré-configurée "cfg-file-nagios" pour tester rapidement un import de cfg.

Utiliser la source pré-configurée cfg-file-nagios

Attention, encore une fois, pour une mise en production sur le long terme, il est préférable que vous utilisiez votre propre source, passez alors sur le chapitre suivant.

Pour un essai d'import de fichier cfg, l'installation ou la mise à jour de Shinken va mettre en place la source cfg-file-nagios déjà configurée :

- cfg-file-nagios. Vous pouvez la voir dans la page des sources.

Cette source utilise 2 types de configuration de fichiers

- **Les fichiers de définition de la source et de ses modules : /etc/shinken/sources/cfg-file-nagios.cfg**
- **Les fichiers de configuration pour l'import de vos éléments via vos fichiers cfg : /etc /shinken-user/source-data/source-data-cfg-nagios/definition-source-data-cfg-nagios.cfg**
  - Ce fichier est pré-configuré pour pointer vers votre dossier de configuration nagios. Il vous suffit de saisir le chemin correspondant pour importer facilement et rapidement votre configuration nagios.

Il vous suffira alors d'activer la source pour importer vos éléments définis dans le fichier nagios.cfg.

Créer et utiliser sa propre source et module

Contexte

Le fait d'avoir plusieurs sources distinctes peut vous aider si vous avez un grand nombre de fichiers cfg et que vous souhaitez pouvoir choisir et séparer les imports. Par exemple, vous avez des fichiers cfg relatifs à l'organisation de Paris, et d'autres à l'organisation de Bordeaux (Hôtes, Utilisateurs, ..).

A un instant donné, vous pouvez avoir de besoin d'importer les fichiers de Bordeaux uniquement. Ce sera chose possible avec les deux sources distinctes.

Ce chapitre va vous permettre de configurer votre propre source depuis la source exemple : cfg-file-sample

A chaque fois que vous avez besoin de personnaliser Shinken, le répertoire à utiliser est **/etc/shinken-user/**

Création d'une source

Placez vous à présent dans le répertoire des sources Shinken (dans /etc/shinken/sources/) et copiez la source Sample existante (qui a le format d'une source standard sur laquelle vous pouvez vous baser pour créer une source personnalisée) :

```
cd /etc/shinken/sources/
cp cfg-file-sample.cfg cfg-file-ma-source.cfg
```

Modifiez le fichier cfg-file-ma-source.cfg

Supprimez les 4 lignes allant de

 # Shinken Enterprise

à

 # End of Shinken Enterprise part

Modifiez

```
    source_name         cfg-file-sample
        cfg_path            /etc/shinken-user/source-data/source-data-cfg-sample/definition-source-data-cfg-
sample.cfg
```

avec

```
    source_name                         cfg-file-ma-source
    cfg_path            /etc/shinken-user/source-data/source-data-cfg-ma-source/definition-source-data-cfg-
ma-source.cfg
```

(ce chemin sera celui du répertoire de vos fichiers cfg dans le shinken-user)

En ce qui concerne les paramètres des sources, voici leurs descriptions :

Propriété	Exemple	Description
source_name	cfg-file-ma-source	Nom de la source affichée dans l'**UI de configuration** en page d'accueil. **Doit être unique** et d'une **longueur inférieure à 40 caractères**, sans quoi le Synchronizer ne démarrera pas.
module-type	`cfg-file-import`	Définition du type de module utilisé par la source.
cfg_path	/etc/shinken-user /source-data/source-data-cfg-ma-source /definition-source-data-cfg-ma-source.cfg	Indique le chemin du fichier de définition de la source.

not_stored_properties	< liste des propriétés >	Ce paramètre permet de définir un ou plusieurs propriétés que ne seront pas importés dans shinken. Cela peut être utile pour exclure une propriété ou bien utiliser des propriétés personnalisées utiles pour la gestion de vos fichiers .cfg
properties_used_as_synckey	address	Définit la liste de propriétés qui seront utilisées en plus du nom et du SE_UUID de l'élément pour générer les clés de synchronisation (sync_key). Ce paramètre est optionnel. Si ce paramètre n'est pas présent, sa valeur par défaut vaut propriété "address". S'il est défini à vide, la propriété "address" ne sera pas utilisé comme synckey.
update_cfg_with_staging_se_uuid	1	Ce paramètre permet de vérifier, avant l'import, si un élément (quelque soit son type) existe déjà en Staging. Ce sont les clés de synchronisation (sync_keys) qui sont utilisées pour déterminer si l'élément existe en Staging. Si c'est le cas et que cet élément ne dispose pas de SE_UUID dans la source, alors le fichier sera édité afin d'insérer le SE_UUID de l'élément en Staging. Cela permet de s'assurer que l'élément soit toujours calculé correctement, même si son nom ou son adresse change. *Note : pour cela il faut que l'utilisateur Shinken est les droits d'écriture sur les fichiers .cfg.*
create_and_write_SEUUID_in_file	0	Ce paramètre permet de créer et d'écrire un SE_UUID dans le fichier de la source, avant l'import, sur tous les éléments qui ne possèdent pas de SE_UUID, quelque soit leur type. Si l'élément existe déjà dans l'interface de configuration alors il pourra avoir conflit, il faut donc activer ce paramètre que si vous êtes sûr que les éléments n'existent pas déjà dans l'interface. *Note : pour cela il faut que l'utilisateur Shinken est les droits d'écriture sur les fichiers .cfg.*

Si le paramètre **create_and_write_SEUUID_in_file** est utilisé, le SE_UUID sera créé même si un élément correspondant existe déjà dans staging. Le paramètre **update_cfg_with_staging_se_uuid** sera donc inutile.

Création du répertoire de données de la source

Pour créer son propre répertoire dans le répertoire des sources de shinken-user :

```
cd /etc/shinken-user/source-data
cp -r source-data-cfg-sample/ source-data-cfg-ma-source/
mv source-data-cfg-ma-source/definition-source-data-cfg-sample.cfg source-data-cfg-ma-source/definition-source-data-cfg-ma-source.cfg
```

Rajouter les droits sur le répertoire :

```
cd /etc/shinken-user/source-data
chown -R shinken:shinken source-data-cfg-ma-source/
```

Astuce

Une fois le répertoire créé, les fichiers cfg pourront alors être placés directement dans le répertoire source-data-cfg-ma-source/elements ou source-data-cfg-ma-source/packs
Dans tous les cas, tous les fichiers cfg dans le répertoire et sous répertoires de source-data-cfg-ma-source seront traités.

Configuration du démon Synchronizer pour déclarer la source

Modifiez le fichier **/etc/shinken/synchronizers/synchronizer-master.cfg**

A la fin de la ligne "sources", rajoutez votre source.

Exemple :

```
    sources              syncui,cfg-file-shinken,active-dir-example,sync-vmware,cfg-file-nagios,discovery,
openldap-example,cfg-file-ma-source
```

Redémarrez le synchronizer pour qu'il prenne acte de la nouvelle source :

```
service shinken-synchronizer restart
```

Affichage de la source dans l'UI de configuration

Une fois le démon redémarré, votre nouvelle source personnalisée doit apparaître :

Son paramétrage sera alors en place : son état (activée ou désactivée), son ordre, sa fréquence en automatique ou en manuel.

Import des objets

Depuis la page d'accueil de l'interface de configuration, vous pouvez alors utiliser le bouton "**Force import**" en cliquant sur le bouton 

Dans le panel des éléments, vos nouveaux éléments doivent apparaître. Vous pourrez alors les importer dans votre base.

Depuis une page spécifique de source, vous pouvez également appuyer sur le Bouton  , (il ne sera disponible que si la source est active). Une pop-up sera lors affichée jusqu'à la fin de l'import, ensuite la page sera rechargée pour être mise à jour.

Données globales

Les données globales peuvent être configurées dans ***/etc/shinken/resources.d/***.

- Ce dossier contient des fichiers définissant des données globales pour différents sujets (email, mysql, oracle, ...) accessibles par vos objets globalement dans Shinken.
- Vous trouverez plus de détail sur l'utilisation des données globales dans le chapitre LES VARIABLES (Remplacement dynamique de contenu - Anciennement les MACROS)

Pour une organisation plus facile et une configuration plus propre, les données globales peuvent également être configurées dans les sources.

- Le dossier d'exemple d'une source cfg est situé dans ***/etc/shinken-user/source-data/source-data-cfg-sample*** et contient un dossier global-data dans lequel les définitions des données globales peuvent être placées (un fichier échantillon est également présent dans ce dossier)
- Lorsque la source est importée, ces fichiers de données globales seront copiés dans ***/etc/shinken/resources.d/module-source_name***

Ce mécanisme vous permet également de surcharger les valeurs définies dans ***/etc/shinken/resources.d*** avec des fichiers définis dans les sources.

Par contre ces fichiers contenant des définitions de donnée globale ne sont pris en compte **qu'après le rechargement du Synchronizer**.

Précisions techniques

Clés de synchronisation

Les clés utilisées pour la synchronisation entre les sources sont des chaines de caractères accrochées sur les éléments. Elles permettent de considérer un élément comme identique provenant de plusieurs sources, même si l'élément n'est pas identique dans les sources. s

La source choisie 1 ou plusieurs propriétés de l'élément importé comme clé de synchronisation (différent suivant les sources et leurs paramétrages).

- Le fonctionnement et l'utilité des clés de synchronisation sont décrits de manière plus détaillée dans la page de documentation dédiée: Précisi on technique sur le fonctionnement de l'import des sources (**il importait que vous lisiez cette page avant d'aller plus loin dans ce chapitre**).

Le module d'import de fichier cfg utilise le paramètre properties_used_as_synckey de son fichier de configuration pour lister les propriétés de l'objet qui seront accrochées à un élément importé en tant que clés de synchronisation.

- Si un élément importé a une ou plusieurs propriétés remplies lors de l'import, le Synchronizer les rajoutera à la liste des valeurs de synchronisation de cet objet.

Important

Les éléments d'une même source qui ont au moins une clé de synchronisation commune doivent avoir toutes leurs clés de synchronisation identique (Exemple pour un hôte:même nom et même adresses) pour être fusionnés

- Si le paramètre properties_used_as_synckey
  - n'est pas défini,
    - sa valeur par défaut est à *address. Tout élément ayant la propriété address aura la valeur de son "address" ajouté dans les clés de synchronisation.*
  - est vide,
    - aucune propriété ne sera ajoutée dans les clés de synchronisation.
  - est redéfini à *display_name*
    - seul la valeur de cette propriété sera ajoutée dans les clés de synchronisation.
    - Remarque: plusieurs propriétés peuvent être définies comme clef de synchronisation (cela dépendra de vos besoins)

```
properties_used_as_synckey host_name, display_name, address
```

- Si dans une même source, deux éléments importés ont toutes leurs clés de synchronisation sont communes, les deux éléments seront fusionnés.
  - Par exemple, pour des clés de synchronisation sur le nom et l'adresse :
    - Si Hôte 1 et Hôte 2 ont le même nom et la même adresse, ces deux éléments seront fusionnés.
    - Par contre si Hôte 1 et Hôte 2 ont le même nom, mais pas la même adresse, ces éléments ne seront pas fusionnés, car toutes les clés ne sont pas communes

Remarque: le Synchronizer ajoute automatiquement le nom de l'élément comme clé de synchronisation.