

# Comment appliquer un modèle d'hôte à un hôte - Pack Switch-SNMP

## Sommaire

[Application du modèle via l'interface de Configuration](#)  
[Application du modèle via un collecteur d'import de fichiers au format .cfg](#)

## Intérêt global de la commande

Sur les grands environnements, la majorité de la charge CPU va être consommée par les sondes de supervision. Quand elles sont nombreuses et qu'elles sont ordonnancées avec des intervalles de temps différents, il peut être difficile :

- d'extrapoler, par rapport au nombre d'hôtes déjà présent, de combien de CPU sur ses Pollers une infrastructure à besoin,
- sur une infrastructure, savoir qu'elles sont les sondes les plus consommatrices en termes de CPU :
  - une fois identifié, il est intéressant de passer du temps pour optimiser ces sondes afin de diminuer le nombre de CPU nécessaire.

C'est pour répondre à ces problématiques que nous avons créé cette commande.

Vous trouverez des exemple d'analyses de résultat de la commande dans les pages suivantes :

- Vous avez un environnement avec un certain nombre d'hôtes, vous souhaitez dimensionner vos Pollers si vous rajoutez 10, 20 ou 50% d'hôtes en plus :
  - dans ce cas, il faut se référer à la page [FOR SHINKEN TEAM - Dimensionnement des Pollers à l'aide de la commande shinken-scheduler-export-data](#).
- Si vous souhaitez diminuer le nombre de CPU de vos Pollers en identifiant, et optimisant, les sondes les plus consommatrices :
  - dans ce cas, il faut se référer à la page [FOR SHINKEN TEAM - Identification des sondes les plus consommatrices à l'aide de la commande shinken-scheduler-export-data](#).

Cette commande interroge les Schedulers, car ce sont eux qui centralisent les informations de temps d'exécution des sondes de supervision.

## Fonctionnement général de la commande

### Récupération des données des checks auprès des Schedulers

La commande "shinken-scheduler-export-data" est seulement utilisable sur le serveur de l'Arbiter. En effet, elle se connecte sur tous les Schedulers, et c'est le seul démon habilité à cela.

La commande va lire la configuration depuis /etc/shinken et va pouvoir se connecter sur tous les Schedulers qui sont définis dans l'architecture. La récupération peut se faire de deux manières différentes :


- anonyme ( *aucun nom ne sera renseigné dans le fichier généré* ),
  - L'extraction sera anonymisée ( *seule un hash des noms sera lisible* ) sauf si l'argument : --full est ajouté à la commande.
- avec les noms des éléments ( *voir l'option --full* )
  - Le mot de passe est celui défini dans le fichier de configuration de Shinken ( /etc/shinken/shinken.cfg ) dans la clé : daemon\_\_export\_\_data\_\_password. ( *Voir la page [Fichier de configuration \(shinken.cfg\)](#)* ).

Il est possible de désactiver l'extraction pour un Scheduler via l'option défini dans son fichier de configuration dans la clé : scheduler\_\_export\_data\_\_enabled. ( *Voir la page [Le Scheduler](#)* )

### Lancement de la commande

La commande accepte les options suivantes :

Nom	Type	Unité	Défaut	Commentaire
<code>--timeout</code>	Entier	sec	<b>10</b>	Permet de configurer le temps d'attente acceptée sur les appels des Schedulers avant de retourner une erreur.
<code>--full</code>	Booléen	---	<b>False</b>	L'export se fait de façon anonyme, c'est-à-dire que les noms des éléments ne seront pas visibles. Si activé, l'export demandera également les noms des éléments ( <i>Pour ce faire, il utilisera la variable scheduler__export_data__password qui devra être configurée dans les .cfg des Schedulers, voir la page <a href="#">Le Scheduler</a></i> ).
<code>--realm-filter</code>	Texte	---		Prends le nom du royaume que l'on souhaite utiliser comme filtre afin de n'avoir que des Schedulers de ce royaume dans notre extraction de données.  ( <i>on ne peut mettre qu'un seul royaume dans le filtre</i> )

<code>--simulation-extra-period</code>	Entier	sec	0	<p>Cette option permet à partir de la date du check suivant et du <code>check_interval</code> d'un élément d'extrapoler sur la période fourni l'exécution des checks.</p> <p>Cela permet de simuler l'exécution des checks sur une période.</p> <ul style="list-style-type: none"> <li>• Une ligne de donnée sera donc ajoutée pour chaque exécution simulée.</li> <li>• Le nombre de lignes ajoutées correspond à la durée de la simulation divisée par l'intervalle de vérification de l'élément.</li> </ul> <p><i>Par exemple avec <code>--simulation-extra-period</code> à 3600 un check avec un intervalle de vérification à une minute générera 60 lignes.</i></p> <p><i>Avec une ligne qui représente une exécution de check.</i></p> <div style="border: 1px solid red; padding: 5px; margin-top: 10px;"> <p> Cette simulation ne prend pas en compte :</p> <ul style="list-style-type: none"> <li>• les <b>périodes de vérification</b> ( <i>prévu pour une futur version</i> ),</li> <li>• les <b>intervalles de nouvelles tentatives de confirmations d'état</b> ( <code>retry_interval</code> ) =&gt; Difficile de définir de manière pertinente combien d'incidents pourraient apparaître et engendrer une activité supplémentaire sur les Pollers.</li> </ul> <p>Donc les valeurs données par cette simulation ne seront pas forcément exhaustive.</p> </div>
<code>--skip-errors</code>	Booléen	---	False	Si activé, la commande ne s'arrêtera plus si un ou plusieurs Scheduler est éteint/mal configuré, ou s'il ne répond pas ( <i>time out</i> ).
<code>--debug</code>	---	---	---	Rajout des données exportées dans le terminal.

## Résultat du lancement sur le terminal

Le lancement de la commande fera deux choses :

- un retour dans le terminal pour voir le retour global de la commande,
- la génération d'un fichier .csv avec l'ensemble des données.
  - Les lignes du fichier csv représente chaque occurrence d'exécution de chaque check.

Deux exemples seront fournis pour agrémenter la documentation:

- Exemple 1 : exécution avec toutes les données ( `--full` ) et sans simulation dans le futur.

```
shinken-scheduler-export-data --full
```

- Exemple 2 : exécution anonyme avec une demande de simulation d'une heure supplémentaire ( `--simulation-extra-period=3600` )

```
shinken-scheduler-export-data --simulation-extra-period=3600
```

## Le retour du terminal

Le retour donne :

- la liste des Schedulers,
- l'adresse du fichier ou le CSV a été généré,
- le nombre de ligne générer ( *qui correspond aux nombres d'éléments et simulé si l'option `--simulation-extra-period` a été utilisé* )

### Exemple 1

? Unknown Attachment

## Exemple 2

? Unknown Attachment

À noter que dans l'exemple 2, le nombre de lignes est bien plus élevée à causes des nombreuses lignes de la simulation ( 3163 au lieu de 55 dans notre exemple ).

## CSV des données

Elle va générer un fichier **csv** dans le répertoire **/tmp** du serveur dont le nom va dépendre des options utilisées :

- Exemple 1 :

`/tmp/dump-schedulers--not-expanded--with-names--2023-01-31-14h32m54s.csv`

- Exemple 2 :

`/tmp/dump-schedulers--expanded-3600s--anonymous--2023-01-31-14h31m32s.csv`

## Contenu du fichier csv généré

Dans le fichier CSV, on a les données suivantes :

- host\_uuid** : UUID de l'hôte.
- host\_name** : Nom de l'hôte ( *présent si l'option `--full` est défini dans la commande* ).
- host\_name\_anonymous\_hash** : Hash du nom pour l'identifier sur un export sans l'option `--full`.
- check\_uuid** : UUID du check.
- check\_name** : Nom du check ( *présent si l'option `--full` est défini dans la commande* ).
- check\_name\_anonymous\_hash** : Hash du nom pour l'identifier sur un export sans l'option `--full`.
- command\_name** : Nom de la commande ( *présent si l'option `--full` est défini dans la commande* ).
- command\_name\_anonymous\_hash** : Hash du nom pour l'identifier sur un export sans l'option `--full`.
- realm** : Nom du royaume ( *présent si l'option `--full` est défini dans la commande* ).
- realm\_anonymous\_hash** : Hash du nom pour l'identifier sur un export sans l'option `--full`.
- check\_interval** : Intervalle entre les vérifications.
- retry\_interval** : Intervalle de nouvelles tentatives de confirmations d'état.
- cpu\_time** :
  - Temps CPU utilisé par la sonde.
  - Format chaîne de caractères représentant un float avec un ".".
- scheduled\_epoch** : seconde où le check est planifié ( *en epoch, depuis le 1er janvier 1970* ).
- scheduled\_date** :
  - Format YYYY-MM-DD hh:mm:ss :
    - Exemple : 2023-01-31 10:49:1.
    - Importable dans Excel :
      - Il faut juste penser à lui demander d'afficher les secondes sur la colonne.
- is\_simulated** :
  - Booléen permettant de savoir si le check est issu d'une vraie exécution prévue dans le Scheduler, ou s'il est issu d'une simulation du check dans le futur.

## Chargement et analyse des données

### Chargement du fichier csv

Les fichiers CSV générés peuvent être importés dans un tableur comme Excel.

- L'import dans ce dernier se fait simplement, dans les autres tableurs l'import suivra un mécanisme similaire.
- Depuis un tableur vide, il faut passer par DonnéesFichier texte.

? Unknown Attachment

Vu que le fichier a une première ligne avec la description des champs, il faut bien activer "Mes données ont des en-têtes".

? Unknown Attachment

Sur la phase suivante, il faut bien placer le séparateur sur "Virgule".

- L'Aperçu de données permet de s'apercevoir que la séparation des champs est bien prise en compte.

? Unknown Attachment

Si la majorité des champs sont des formats simples ( *chaines de caractères ou entier* ), deux colonnes sont particulières et on doit indiquer à Excel comment les interpréter :

- **cpu\_time** : Il faut lui indiquer via "Avancé" que le séparateur de décimale n'est pas "," mais "." ( *à faire que dans un Excel en Français, car dans le Excel en Anglais le séparateur par défaut est déjà le point* ).

? Unknown Attachment

- **scheduled\_date**: Il faut lui indiquer que le champ est un format Date.

? Unknown Attachment

Excel nous demande alors où charger ces données. Il est plus simple et lisible de le charger dans une "Nouvelle feuille de calcul" que l'on pourra renommer facilement.

? Unknown Attachment

Voici un exemple de ce qu'on obtient finalement :

? Unknown Attachment