

(PROCEDURE) Passer de Centos 7.9 à RedHat 7.9

Sommaire

[Intérêt de passer de Centos à RedHat](#)
[Convertir un serveur Centos7.9 en RedHat7.9: officiellement supporté par RedHat, mais complexe et risqué d'après notre expérience](#)
[Opération concrète de conversion](#)
 [Sauvegarde intégrale de votre serveur](#)
 [Mettre à jour le système vers un Centos 7.9 à jour](#)
 [Arrêter les démons Shinken et MongoDB](#)
 [Récupération l'outil convert2rhel](#)
 [Lancement du processus de conversion](#)
 [Erreurs possibles](#)
 [Erreur de licence](#)
 [Problème avec grub/uefi](#)

Connaitre les consommation des sondes afin de pouvoir dimensionner ses Pollers quand on rajoute des hôtes

Quand on a mis en place un échantillon représentatif de son infrastructure en supervision, il est utile de savoir quelles va être la consommation CPU quand on mettra l'ensemble de son infrastructure en place. Si l'on part du principe que vous avez mis en supervision environ 10% du parc d'un client et 20% d'un autre, il est utile de calculer combien chacun consomme en terme de sondes de supervisions sur les Pollers afin de pouvoir prévoir l'allocation finale des Pollers quand 100% du parc sera supervisé.

Pour cela, nous allons extraire des informations issues de la commande **shinken-scheduler-export-data** afin d'avoir:

- **le nombre de CPU utilisés sur les pollers par chaque royaume**

Pour cela, nous allons avoir besoin d'un dump de données avec en option une prévision de la charge sur une période, disons par exemple 1 heure. Il suffit alors de lancer la commande comme ceci:

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



Avec ce lancement, les noms (hôte, check, commandes et royaumes) ne seront pas présents dans l'export.

Il est tout à fait possible de faire l'analyse sur un export avec les noms en ajoutant l'option `--full` si vous voulez faire une autre analyse avec le même fichier .

L'importation du fichier .csv généré est décrit dans [FOR SHINKEN TEAM - shinken-scheduler-export-data : export des données du Scheduler](#)

Création du tableau récapitulatif sur la consommation totale des temps CPU des royaumes respectifs

Consolidation des données: utilisation d'un tableau croisé dynamique

A partir des nombreuses données de checks que nous avons, nous allons devoir procéder à une consolidation afin d'avoir notre résultat facilement exploitable. Pour cela nous allons utiliser un tableau croisé dynamique.

Un tableau croisé dynamique dans un tableur est un outil de analyse de données qui vous permet de créer une vue synthétique et facile à lire d'une grande quantité de données (ici nos exécutions de checks). On y choisi les données à inclure, choisi comment les organiser et comment les synthétiser, puis filtrer, classer et totaliser les données en fonction de nos besoins.

Création du tableau croisé dynamique

La création du tableau récapitulatif passe par la création d'un Tableau croisé dynamique. Depuis votre Feuille d'importation des données, il faut cliquer sur Insertion Tableau croisé dynamique, et valider:

? Unknown Attachment

Sélection des royaumes en tant que lignes de notre tableau

Arrivé sur la nouvelle feuille, Excel demande quelles lignes sélectionner, dans le bloc de droite.

Dans notre cas, il faut faire glisser le champ **realm** (ou bien **realm_anonymous_hash** si on a une version anonyme de l'export) vers le bloc "**Lignes**":

? Unknown Attachment

Sélection des "Valeurs": le `cpu_time`, le temps consommé par les sondes

A chaque royaume, il faut lui assigner une (ou plusieurs) "Valeurs". Pour cela, on fait glisser le champ "`cpu_time`" vers le bloc "Valeurs" afin d'avoir pour chaque royaume son champs `cpu_time` associé.

Par contre par défaut Excel prends le nombre d'occurrences du champ `cpu_time` comme "Valeur", ce qui n'est pas ce qui est souhaité:

? Unknown Attachment

Passer du nombres de lignes avec "`cpu_time`" à une vrai somme des temps CPU

Une modification des "Paramètres des champs de valeurs" est nécessaire sur "Nombre de `cpu_time`" qu'il faut changer en Somme pour donner au final "Somme de `cpu_time`":

? Unknown Attachment

Passer du nombre total de temps CPU consommé au nombre de CPU nécessaires

Ceci fait, le tableau obtenu permet d'avoir le temps CPU utilisé pour l'ensemble des sondes, par royaume, sur la période utilisée pour l'export (ici 1 heure = 3600 secondes).

Afin d'avoir le nombre de CPU nécessaire, il faut revenir à l'échelle d'une seconde, et donc rajouter une nouvelle colonne avec comme valeur la "Somme de `cpu_time`" divisée par **3600**:

? Unknown Attachment

Dans notre cas d'exemple, l'environnement a beaucoup de marge et ne consomme largement moins qu'un seul CPU.