

Shinken-backup et Shinken-restore, les commandes de sauvegarde et de restauration

Sommaire

- Contexte
- Commande de sauvegarde (shinken-backup)
 - Utilisation
 - Les options disponibles
 - Options de connexion à la base MongoDB
 - Options génériques
 - Options de connexion SSH
 - Options d'authentification
 - Options SSL/TLS
 - Sauvegarde de données de configuration chiffrées
 - Sauvegarde d'une configuration anonymisée
 - Exemples
 - Sauvegarde complète d'un serveur hébergeant l'ensemble des démons
 - Sauvegarde sur un serveur hébergeant le démon Synchronizer
 - Erreur lors du Backup
- Commande de restauration (shinken-restore)
 - Utilisation
 - Les options disponibles
 - Options de connexion à la base MongoDB
 - Options génériques
 - Options de connexion SSH
 - Options d'authentification
 - Options SSL/TLS
 - Restauration d'une base de données de configuration dont les données sensibles sont chiffrées
 - Exemples
 - Restauration d'une sauvegarde complète de Shinken depuis le dossier ~/shinken-backups
 - Restauration d'une sauvegarde de la configuration de Shinken, lancé depuis le serveur hébergeant l'Arbiter /Synchronizer :
 - Restauration d'une sauvegarde des données de l'utilisateur "monutilisateur" de Shinken (portails, listes, favoris, tableaux de bord), lancé depuis le serveur hébergeant le Broker
 - Restauration avec une sauvegarde contenant seulement certaines parties
 - Erreurs lors de la restauration

Contexte

Afin de prévenir d'une perte de données après un crash d'un ou de plusieurs de vos serveurs utilisés dans votre architecture Shinken, nous vous conseillons d'utiliser les commandes shinken-backup et shinken-restore.

Ces deux commandes vous permettront de sauvegarder ou de restaurer tout ou une partie d'un serveur de votre architecture Shinken.

Les commandes **shinken-backup** et **shinken-restore** permettent de restaurer automatiquement les données suivantes :

- Les données sur les add-ons (*par exemple l'add-on Nagvis*)
- Les données de configuration des éléments de Shinken comme les hôtes, clusters, checks ... (*il est possible d'anonymiser ces données*)
- Les données du bac à événements
- Les logs
- Les données des modules **NON fournis** par Shinken Enterprise
- Les données de la métrologie
- Les données des SLA
- Les données des utilisateurs



La rétention effectuée par le Scheduler n'est **PAS** sauvegardée et restaurée automatiquement par ces scripts.

Il est tout de même possible de le faire manuellement en suivant cette page de documentation : [Sauvegarde et restauration manuelle de la rétention](#)

Commande de sauvegarde (shinken-backup)

Utilisation

La commande de base pour la sauvegarde complète d'un serveur shinken est la suivante :

shinken-backup

Les différentes options possibles :

Option courte	Option	Description	Démon sur lequel se trouve les données à sauvegarder
-h	--help	Affiche l'aide de la commande	Tous
-od [dir]	--output-directory [dir]	Permet de choisir le répertoire où sera enregistrée la sauvegarde. Les répertoires nécessaires seront créés : • Soit le chemin fourni est absolu (<i>/root/backups/example</i>) • Soit le chemin relatif, dans ce cas le répertoire fourni sera créé à partir du répertoire où sera exécutée la commande shinken-backup. <i>(Par défaut, la sauvegarde se fait dans le répertoire où est lancée la commande)</i>	Tous
-on [name]	--output-name [name]	Permet de choisir le nom du répertoire contenant la sauvegarde <i>(par défaut il sera YYYY-MM-DD__HH-MM-SS__ShinkenVersion)</i>	Tous
-a	--addons	Sauvegarde les configurations et les données des addons (<i>exemple : Nagvis</i>) • Les données de nagvis dans /opt/nagvis et /etc/httpd/conf.d/ • Les données de nagvis-shinken-architecture dans /var/lib/shinken-nagvis et /etc/httpd/conf.d	Tous
-c	--configuration	Sauvegarde les données de configuration : • Les données du Synchronizer dans MongoDB • Les fichiers de configuration dans /etc/shinken/ • Les fichiers de configuration dans /etc/shinken-user • Les sondes utilisateurs dans /var/lib/shinken-user/libexec ( <i>Si vos sondes utilisent des dépendances systèmes, il faudra aussi qu'elles soit présentes lors de la restauration pour que les sondes soient opérationnelles</i>). • La rétention de l'Arbiter	Synchronizer, Arbiter
-ca	--configuration-anonymous	Sauvegarde les données de configuration anonymisées • Les données du Synchronizer dans MongoDB anonymisées Voir le chapitre : Sauvegarde d'une configuration anonymisée	Synchronizer
-e	--events	Sauvegarde les données du bac à événements • Les données mongo du bac à événements	Broker
-l	--log	Sauvegarde les logs • Tous les logs du dossier /var/log/shinken	Tous
-M	--modules	Permet de sauvegarder les données de tous les modules NON fournis par Shinken Enterprise (<i>exemple : le module mssql_export</i>) • Les données des modules dans MongoDB	Tous
	--module-type [TYPE]	Permet de sauvegarder les données d'un seul type de module NON fourni par Shinken Enterprise. Uniquement utilisable avec l'option --modules/-M.	Tous
-L	--list-modules	Permet de lister tous les types de modules NON fournis par Shinken Enterprise qui peuvent être sauvegardés.	
-m	--metrology	Sauvegarde les données de métrologie • Les fichiers du dossier /opt/graphite/storage/whisper	Broker
-s	--sla	Sauvegarde les données SLA • Les données mongo des SLA	Broker

-u	--user	Sauvegarde les données des utilisateurs de l'interface de visualisation (<i>portails, listes, favoris, tableaux de bord...</i>) - Les données mongo des dashboard utilisateurs - Les données mongo des listes utilisateurs - Les données mongo des préférences utilisateurs	Broker
	--mongo-host	Nom ou IP du serveur MongoDB à contacter (<i>défaut: localhost</i>)	Tous
	--mongo-port	Port du serveur MongoDB à contacter (<i>défaut: 27017</i>)	Tous
	--mongo-use-ssh	Utiliser un tunnel SSH pour accéder au serveur MongoDB (<i>défaut: non</i>)	Tous
	--mongo-ssh-key	Clé privée à utiliser pour créer le tunnel SSH vers le serveur MongoDB (<i>défaut: /var/lib/shinken/.ssh/id_rsa</i>)	Tous
	--mongo-ssh-user	Utilisateur pour la connexion SSH sur le serveur MongoDB (<i>défaut: shinken</i>)	Tous



La commande **shinken-backup** ne peut pas être exécutée dans les dossiers **/etc/shinken** et **/etc/shinken-user**

Attention, pensez bien à utiliser cet outil générique sur le bon serveur.

Par exemple, sauvegarder les SLA depuis un serveur Poller ne sera pas cohérent. Ou encore, pour sauvegarder la configuration de Shinken, placez-vous sur le serveur hébergeant le duo Arbiter / Synchronizer.

Sauvegarde de données de configuration chiffrées

Lorsque la sauvegarde de données de configuration avec des données protégées est effectuée, **shinken-backup** affiche un avertissement si la clé n'a pas été sauvegardée.

La sauvegarde est tout de même effectuée, mais vous devez effectuer une sauvegarde de la clé en utilisant la commande [shinken-protected-fields-keyfile-export](#) avant toute autre opération impliquant un changement de clé.

Le message suivant apparaîtra si vous n'avez pas effectué de sauvegarde de la clé avant la sauvegarde de la configuration :

The protected fields key from this backup looks like it has never been saved

Si la sauvegarde de la clé a été effectuée par la suite, vous pouvez ignorer ce message et restaurer la clé selon [la procédure de restauration habituelle](#).

```
$ shinken-backup -c -od . --on crypted-backup2
Saving Configuration
Configuration save size: 7.0M
Done : your backup directory is /root/shinken-enterprise_V02.08.02-RC001_BUILD066-FR/2020-10-27__10-47-43__V02.08.02-001_BUILD066.fr__crypted-backup2
```

Sauvegarde d'une configuration anonymisée

Afin de transmettre une configuration au support, il est possible d'anonymiser la configuration lors de la sauvegarde.

L'anonymisation est réalisée pour les données présentes dans la base de données du Synchronizer. Si l'option **--configuration-anonymous** est utilisée, les autres options (**--sla** par exemple) sont ignorées. Les fichiers de configuration des démons et de l'architecture de Shinken ne sont pas sauvegardés.

Cette étape est réalisée entre la récupération des données dans la base et l'écriture des fichiers de sauvegarde, la base de données n'est donc pas impactée par ce fonctionnement. Le tableau suivant présente les propriétés qui seront remplacées avant d'être écrites dans les fichiers de sauvegarde.

Type de donnée	Valeur remplacée	Description
address	clean address	Permet de masquer les adresses des équipements

check_command	clean_command	Permet de masquer les commandes : une fois restaurée, la configuration pourra démarrer sans effectuer de check
poller_tag		La valeur est supprimée pour permettre à la configuration de démarrer
reactionner_tag		La valeur est supprimée pour permettre à la configuration de démarrer
realm		La valeur est supprimée pour permettre à la configuration de démarrer
password	clean	Les mots de passe sont effacés

Toutes les données utilisateur, chiffrées ou non, présentes dans le système de champs protégé sont également remplacées par la valeur "clean". Pour connaître la liste des champs protégés, utilisez la commande **shinken-protected-fields-data-manage**.

Le nom de la sauvegarde généré sera précédé de "--anonymous" afin de le distinguer des sauvegardes par défaut.



Exemple de nom de sauvegarde anonyme : 2020-10-27__10-49-55__V02.08.02-001_BUILD066.fr--anonymous



Les données ne sont ni effacées ni modifiées sur le serveur sur lequel le **shinken-backup** est effectué. Elles sont simplement remplacées avant d'être écrites dans les fichiers de sauvegardes.



La sauvegarde réalisée n'est pas chiffrée, car toutes les données présentant un risque ont été remplacées.

Exemples

Sauvegarde complète d'un serveur hébergeant l'ensemble des démons

```
root@vm-shinken: ~
$ shinken-backup
Saving Events
  Events save size: 51M
Saving Sla
  Sla save size: 147M
Saving User
  User save size: 136K
Saving Configuration
  Configuration save size: 17M
Saving Metrology
  Metrology save size: 47M
Saving Logs
  Logs save size: 1.1G
Saving addons data
  Addons save size: 4.8M
Saving custom modules data
  There is no custom modules in database to backup.

Done : your backup directory is /root/shinken-backups/2021-02-02__08-22-45__V02.08.02-002.fr
```

Sauvegarde sur un serveur hébergeant le démon Synchronizer

Sauvegarde de la configuration :

```
[root@bmar-dev2 ~]# shinken-backup --configuration
Saving Configuration
  Configuration save size: 12M

Done : your backup directory is /root/shinken-backups/2021-03-27__01-10-28__V02.07.06-release.fr-Patched-09
```

Sauvegarde complète :

```

[root@bmar-dev2 ~]# shinken-backup
Saving Sla
  Skipped : No sla data in database
Saving User
  Skipped : No user data in database
Saving Configuration
  Configuration save size: 16M
Saving Metrology
  Metrology save size: 127M
Saving Logs
  Logs save size: 8.0K
Saving addons data
  Addons save size: 4.6M
Saving custom modules data
  Skipped : There is no custom modules in database to backup.

Done : your backup directory is /root/shinken-backups/2021-03-26__21-10-05__V02.07.06-release.fr-Patched-09

```

Erreur lors du Backup

Lorsqu'une erreur arrive pendant le backup, on a un message d'erreur nous indiquant la section où l'erreur est survenue, le dossier qui aurait du contenir le contenu du backup va contenir les logs d'erreurs.

Dans ce message est précisé le nom du dossier créé :

```

$ shinken-backup
Saving Events
  Events save size: 392K
Saving Sla
Mongodb database dump failed, please look at /root/shinken-backups/2022-06-30__14-01-11__V02.08.02-B0007.06-release.fr/step2-Backup02.081.fr/sla/backup-sla.

```



La partie floutée du chemin dans l'image suivante correspond au dossier créé par le shinken-restore. Ce nom de dossier est créé à partir de la date, du nom de la version et de la langue.

Commande de restauration (shinken-restore)

Utilisation

```
shinken-restore DIRECTORY-TO-RESTORE
```

Le dossier "DIRECTORY-TO-RESTORE" doit contenir les dossiers de sauvegardes comme :

- configuration/
- context/
- events/
- logs/
- metrology/
- sla/
- user/

Les différentes options possibles :

Option courte	Option	Description	Démon sur lequel se trouve les données à restaurer
-h	--help	Affiche l'aide de la commande	Tous
-a	--addons	Restaure les configurations et les données des addons (<i>exemple : Nagvis</i>) présent dans l'archive.	Tous
-c	--configuration	Restaure les données de configuration	Synchronizer

	--with-key-backup [HASH]	Pour une sauvegarde contenant des données protégées, restaure également le hash de la clé de chiffrement spécifiée. Il s'agit du résultat de la commande <code>shinken-protected-fields-keyfile-export</code> .	Synchronizer
--rocd	--restore-only-configuration-database	Cette option restaure uniquement les données de la configuration, sans restaurer les fichiers de configuration (<code>.cfg</code>). Cette option est pratique pour transférer la politique de supervision d'une préproduction vers une production (<i>ou inversement</i>), sans changer les fichiers de définition des démons.	Synchronizer
-e	--events	Restaure les données du bac à événements	Broker
-l	--log	Restaure les logs	Tous
-m	--metrology	Restaure les données de métrologie	Broker
-M	--modules	Restaure les données de tous les modules NON fournies par Shinken Enterprise présent dans l'archive. (<i>exemple : le module <code>mssql_export</code> développé par un tiers</i>)	Broker
	--module-type [TYPE]	Restaure les données d'un type de module NON fourni par Shinken Enterprise. Uniquement utilisable avec l'option <code>--modules/-M</code> .	Broker
-L	--list-modules	Permet de lister tous les types de modules NON fournis par Shinken Enterprise présent dans l'archive de sauvegarde et qui peuvent être restaurés.	
-s	--sla	Restaure les données SLA.	Broker
-u	--user	Restaure les données des utilisateurs de l'interface de visualisation (<i>portails, listes, favoris, tableaux de bord...</i>).	Broker
	--restore-only-user [USER]	Restaure les données de l'interface de visualisation (<i>portails, listes, favoris, tableaux de bord...</i>) pour l'utilisateur spécifiée. La valeur à utiliser est le Nom (<code>contact_name</code>) de l'utilisateur - à utiliser avec l'option -u 💡 Cette option ne demande pas le redémarrage de Shinken.	Broker
	--mongo-host	Nom ou IP du serveur MongoDB à contacter (défaut: localhost)	Tous
	--mongo-port	Port du serveur MongoDB à contacter (défaut: 27017)	Tous
	--mongo-use-ssh	Utiliser un tunnel SSH pour accéder au serveur MongoDB (défaut: non)	Tous
	--mongo-ssh-key	Clé privée à utiliser pour créer le tunnel SSH vers le serveur MongoDB (défaut: /var/lib/shinken/.ssh/id_rsa) (active la connexion par tunnel SSH)	Tous
	--mongo-ssh-user	Utilisateur pour la connexion SSH sur le serveur MongoDB (défaut: shinken) (active la connexion par tunnel SSH)	Tous

Restauration d'une base de données de configuration dont les données sensibles sont chiffrées

Si une clé est déjà présente sur le serveur et qu'elle est identique à celle de la sauvegarde, **shinken-restore** restaurera cette sauvegarde en utilisant la clé.

Si la clé n'a pas été exportée, **shinken-restore** affichera un avertissement vous signalant qu'elle est automatiquement exportée dans un fichier temporaire, en vous enjoignant de le déplacer en lieu sûr.

```

$ shinken-restore /root/shinken-backups/2019-01-16__11-58-28
Restoring from 02.06.00-001_BUILD67.fr to 02.06.00-001_BUILD67.fr
-Restoring Configuration
Checking encryption status for your current database : WARNING

The current key was never exported
  I'm exporting it to /root/protected_fields_key-export-2019-01-16__12-00-47
  The current key will be replaced by the one from the backup.
  => Remove this file as soon as possible to prevent security holes

  The keyfile used in the backup is the same as the one from the current configuration.
Stopping Shinken before restoring
The backup you are restoring contains a encrypted Synchronizer database.

```

Si les deux clés sont différentes vous devez spécifier l'option **--with-key-backup** suivi de l'export de la clé, qui vous permet de faire la restauration et de placer automatiquement la clé fournie.

En suivant, vous devrez redémarrer le Synchronizer.

- **Le Synchronizer refusera de démarrer tant que la clé n'est pas restaurée.**

Si vous avez égaré votre clé, nous vous conseillons de lire la page de la documentation "[shinken-protected-fields-keyfile-rescue-from-backup](#)". Cette commande va vous permettre de restaurer votre clé par l'intermédiaire du support Shinken.

```

$ shinken-restore --with-key-backup dmFsaWRhdGlvdjJBM3c0ZVRRMEQ2STVzVVVhWkxMRcUdEhrZ0d1b2JZZTY2YTZ2REVKQVVKaz0K /root/shinken-backups/2019-01-16__11-58-28
Restoring from 02.06.00-001_BUILD67.fr to 02.06.00-001_BUILD67.fr
-Restoring Configuration
Checking encryption status for your current database : ENCRYPTED

Stopping Shinken before restoring
Checking protected fields key : OK

The backup you are restoring contains a encrypted Synchronizer database.

Setting addons state
nagvis : ENABLED
nagvis-shinken-architecture : ENABLED
=> Restoring configuration DONE

```

Exemples

Restauration d'une sauvegarde complète de Shinken depuis le dossier ~/shinken-backups

```
root@vm-shinken: ~/shinken-backups

$ shinken-restore 2020-10-27__10-47-43__V02.08.02-001_BUILD066.fr__crypted-backup2
```

```
Stopping Shinken before restoring
Restoring from 02.08.02-001_BUILD066.fr to 02.08.02-001_BUILD066.fr
-Restoring Sla                DONE
-Restoring User               DONE
-Restoring Configuration      DONE
-Restoring Metrology          DONE
-Restoring Logs               DONE
```

```
Sanatizing your restored data
  fix_double_link              : skip (unecessary)
  fix_double_sync_keys         : skip (unecessary)
  fix_default_item_se_uuid     : skip (unecessary)
  fix_remove_shinken_core      : skip (unecessary)
  fix_remove_deprecated_check  : skip (unecessary)
  fix_remove_undefined_aix_templates : skip (unecessary)
  fix_flapping_thresholds      : skip (unecessary)
  fix_business_impact          : skip (unecessary)
```

Done. You can restart your shinken with `/etc/init.d/shinken start`



Après la restauration des données, des scripts de "Sanatize" sont lancés. Ces scripts permettent de réparer certaines incohérences dans vos données en cas de besoin.

Une fois la restauration terminée, vous devez démarrer Shinken:

```
service shinken start
```

Restauration d'une sauvegarde de la configuration de Shinken, lancé depuis le serveur hébergeant l'Arbiter /Synchronizer :

```
root@vm-shinken: ~/shinken-backups

$ shinken-restore --configuration 2020-10-27__10-47-43__V02.08.02-001_BUILD066.fr__crypted-backup2
```

```
Stopping Shinken before restoring
Restoring from 02.08.02-001_BUILD066.fr to 02.08.02-001_BUILD066.fr
-Restoring Configuration      DONE
```

```
Sanatizing your restored data
  fix_double_link              : executed [OK]
  fix_double_sync_keys         : skip (unecessary)
  fix_default_item_se_uuid     : skip (unecessary)
  fix_remove_shinken_core      : skip (unecessary)
  fix_remove_deprecated_check  : skip (unecessary)
  fix_remove_undefined_aix_templates : skip (unecessary)
  fix_flapping_thresholds      : skip (unecessary)
  fix_business_impact          : skip (unecessary)
```

Done. You can restart your shinken with `/etc/init.d/shinken start`

Restauration d'une sauvegarde des données de l'utilisateur "monutilisateur" de Shinken (portails, listes, favoris, tableaux de bord), lancé depuis le serveur hébergeant le Broker

```

root@vm-shinken: ~/shinken-backups

$ shinken-restore -u --restore-only-user monutilisateur 2020-10-27__10-47-43__V02.08.02-001_BUILD066.
fr__crypted-backup2
Restoring from 02.08.02-001_BUILD066.fr to 02.08.02-001_BUILD066.fr
-Restoring User
  Restoring only the user monutilisateur

Restore of the user monutilisateur data is OK

```

Restauration avec une sauvegarde contenant seulement certaines parties

```

root@vm-shinken: ~/shinken-backups

$ shinken-restore 2021-04-13__12-18-24__V02.08.02-004_BUILD08.fr/
Restoring from 02.08.02-004_BUILD08.fr to 02.08.02-004_BUILD08.fr
=> Restoring Configuration      :
    => Checking encryption status for your current database : NOT ENCRYPTED
    => The keyfile used in the backup is the same as the one from the current configuration.
    => Stopping Shinken before restoring : DONE
    => Restoring configuration database : DONE
The backup you are restoring contains a encrypted Synchronizer database.
    => Setting addons state
nagvis                          : ENABLED
- [ WARNING ] Cannot find local broker daemon configuration in /etc/shinken/brokers. The 'Livestatus'
module must be enabled manually in your broker configuration file for this addon to work properly.

nagvis-shinken-architecture    : ENABLED
- [ WARNING ] Cannot find local broker daemon configuration in /etc/shinken/brokers. The 'Livestatus'
module must be enabled manually in your broker configuration file for this addon to work properly.
- [ WARNING ] Cannot find local arbiter daemon configuration in /etc/shinken/arbiters. The
'architecture-export' module must be enabled manually in your arbiter configuration file for this addon to
work properly.

    => Restoring configuration files      : DONE
    => Restoring configuration            : DONE
=> Restoring Sla                  : SKIPPED
No sla data to restore in /tmp/shinken-backups/2021-04-13__12-18-24__V02.08.02-004_BUILD08.fr
=> Restoring Events              : SKIPPED
No events data to restore in /tmp/shinken-backups/2021-04-13__12-18-24__V02.08.02-004_BUILD08.fr
=> Restoring User                : SKIPPED
No user data to restore in /tmp/shinken-backups/2021-04-13__12-18-24__V02.08.02-004_BUILD08.fr
=> Restoring Metrology           : SKIPPED
No metrology data to restore in /tmp/shinken-backups/2021-04-13__12-18-24__V02.08.02-004_BUILD08.fr
=> Restoring Logs                : DONE
=> Restoring Addons              : DONE
=> Restoring custom modules data : SKIPPED
No custom modules data to restore in /tmp/shinken-backups/2021-04-13__12-18-24__V02.08.02-004_BUILD08.fr

Sanatizing your restored configuration for this version of Shinken
fix_double_sync_keys           : skip (unecessary)
fix_default_item_se_uuid       : skip (unecessary)
fix_remove_shinken_core        : skip (unecessary)
fix_remove_deprecated_check    : skip (unecessary)
fix_remove_undefined_aix_templates : skip (unecessary)
fix_flapping_thresholds        : skip (unecessary)
fix_business_impact            : skip (unecessary)
fix_host_name_inheritance      : skip (unecessary)
fix_bp_rule_args               : skip (unecessary)
fix_uuid_in_sla_info           : skip (unecessary)
fix_synchronizer_install_missing_addons : skip (unecessary)
fix_missing_skeletons          : skip (unecessary)
fix_missing_sources            : executed [OK]
fix_update_protected_fields_parameters : skip (unecessary)
fix_all_move_overload_files    : skip (unecessary)
remove_hostgroups_in_working_area : skip (unecessary)
make_index_for_sla_module      : skip (unecessary)
migrate_links_name_into_id     : skip (unecessary)

```

Restoration done. You can restart your Shinken with `/etc/init.d/shinken start`

Lorsqu'une erreur arrive pendant le restore, on a un message d'erreur nous indiquant qu'une erreur est survenue dans une section en particulier.

Un log d'erreur est alors créé dans le dossier du backup utilisé par le script.

La partie floutée du chemin de la capture d'écran correspond au dossier que l'on donne en paramètre au shinken-restore.