shinken-protected-fields-encryption-enable

Sommaire

Concept

Cette commande permet d'activer le chiffrement sur le Synchronizer.

- Elle affichera un texte explicatif des possibilités de protection des champs et demandera ensuite une confirmation pour être sûr que l'activation du chiffrement est bien souhaité.
  - Si aucune clé de chiffrement n'existe, elle est générée.
  - Si une clé a déjà était mise en place (*par l'activation puis désactivation*), elle sera réutilisée.
- Lors de l'activation :
  - Si le Synchronizer est démarré, il sera redémarré et le chiffrement activé.
  - S'il est stoppé, le chiffrement prendra effet au prochain redémarrage.

Options

Options génériques

Option	Valeur par défaut	Description
-h	---	Affiche l'aide de la commande.

Options d'activation du chiffrement

Option	Valeur par défaut	Description
-q	---	Mode silencieux : n'affiche que le minimum d'informations nécessaires.
-y	---	Force l'activation du chiffrement sans demander la confirmation (*utile lors de l'automatisation d'une installation*).

Options de connexion à la base MongoDB

Cette commande récupère les paramètres de connexion à la base MongoDB depuis la configuration.

- Il est nécessaire d'utiliser les options de la ligne de commande que si les fichiers de configuration ne correspond pas à la base MongoDB sur laquelle, la commande doit être exécutée (

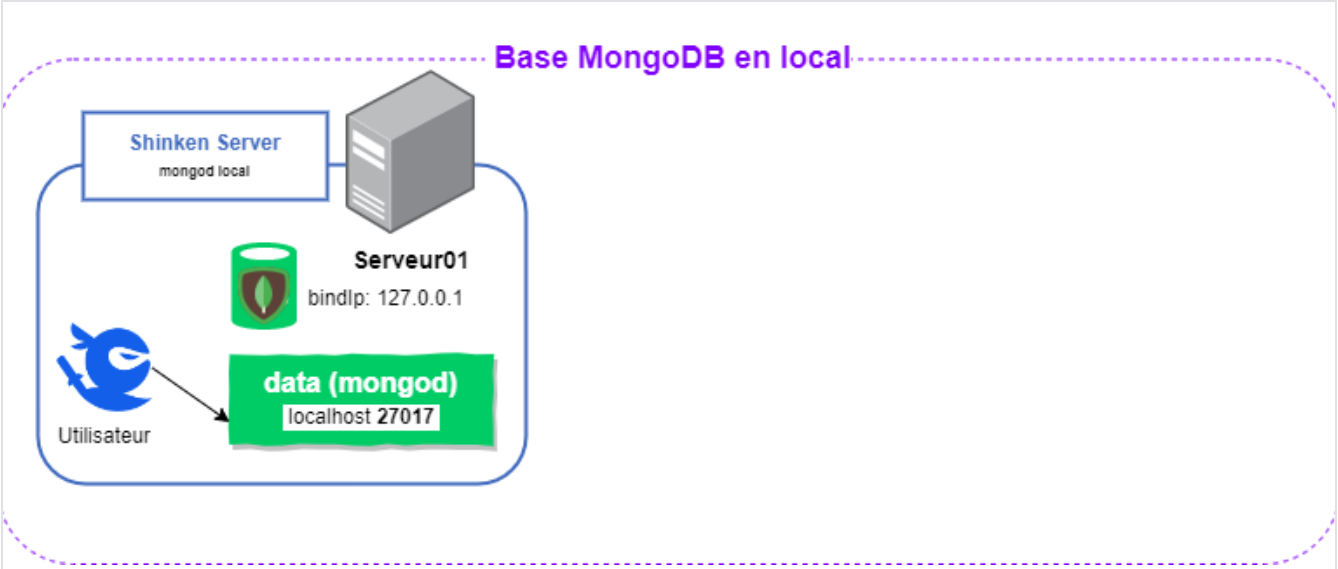
migration de base, test sur une préprod ...).

La commande dispose d'options de connexion à la base MongoDB qui peuvent être utilisés dans les cas suivants :

- La base de données MongoDB ne se trouve pas sur la machine qui exécute la commande.
- L'authentification par mot de passe à la base MongoDB est activée.
- Le port de MongoDB n'est pas celui par défaut (*défaut : 27017*).

La combinaison des options de connexion à MongoDB peut rapidement devenir complexe ; voici des paramètres adaptés aux cas les plus courants.

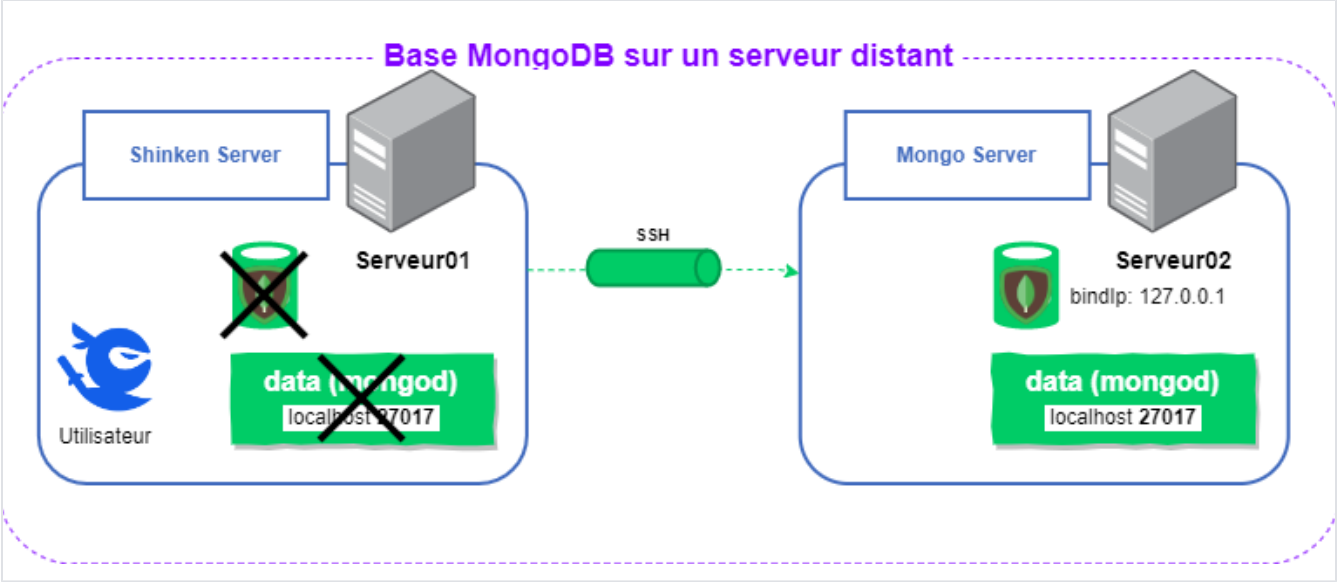
Options génériques




```
[root@serveur01 ~] shinken-commande --mongo-host 127.0.0.1 --mongo-port 27017 --mongo-database shinken
```

Option	Valeur par défaut	Description
--mongo-host *ARG*	**localhost**	Nom ou IP du serveur MongoDB.
--mongo-port *ARG*	**27017**	Port de la base MongoDB.
--mongo-database *ARG*	**shinken** (*ou* **synchronizer** *si la commande concerne la base du Synchronizer*)	Nom de la base de données à utiliser dans MongoDB. À n'utiliser que si la configuration du module ou du démon a changé la base utilisée par défaut.

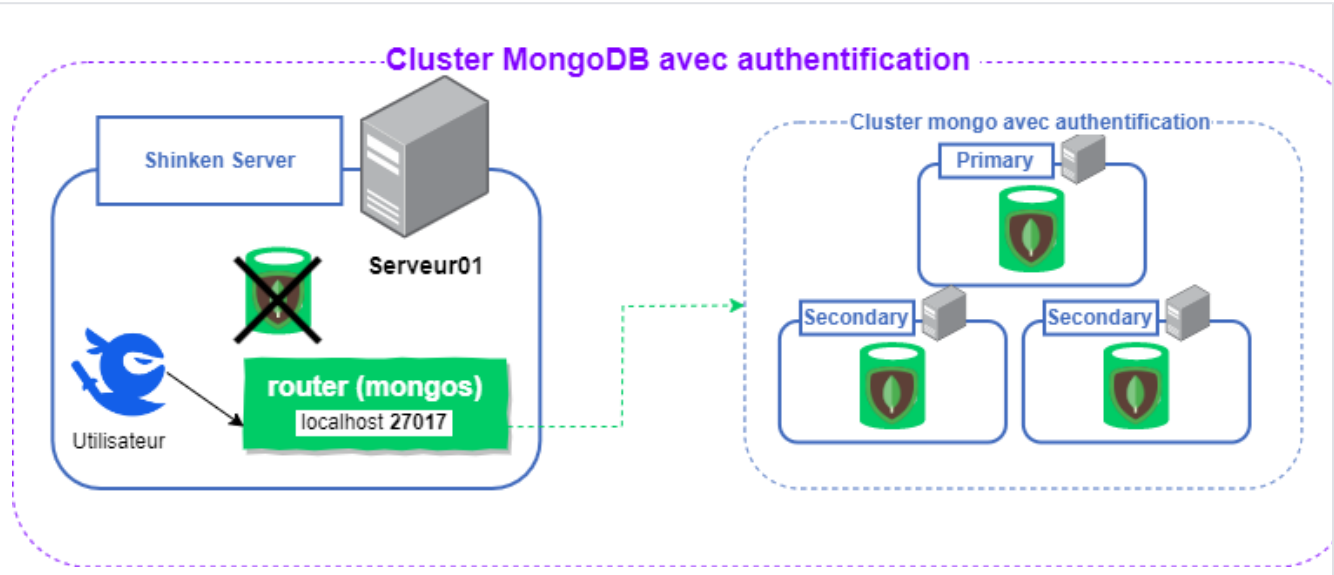
Options de connexion SSH




```
 [root@serveur01 ~] shinken-command --mongo-host serveur02 --mongo-port 27017 --mongo-use-ssh --mongo-ssh-key /var/lib/shinken/.ssh/id_rsa --mongo-ssh-user shinken
```

Option	Valeur par défaut	Description
--mongo-use-ssh	---	Active la connexion SSH au serveur MongoDB.
--mongo-ssh-key *ARG*	**/var/lib/shinken/.ssh/id_rsa**	Clé privée SSH pour la connexion au serveur MongoDB. À utiliser en complément de l'option --mongo-use-ssh.
--mongo-ssh-user *ARG*	**shinken**	Utilisateur à utiliser pour la connexion SSH. À utiliser en complément de l'option --mongo-use-ssh.

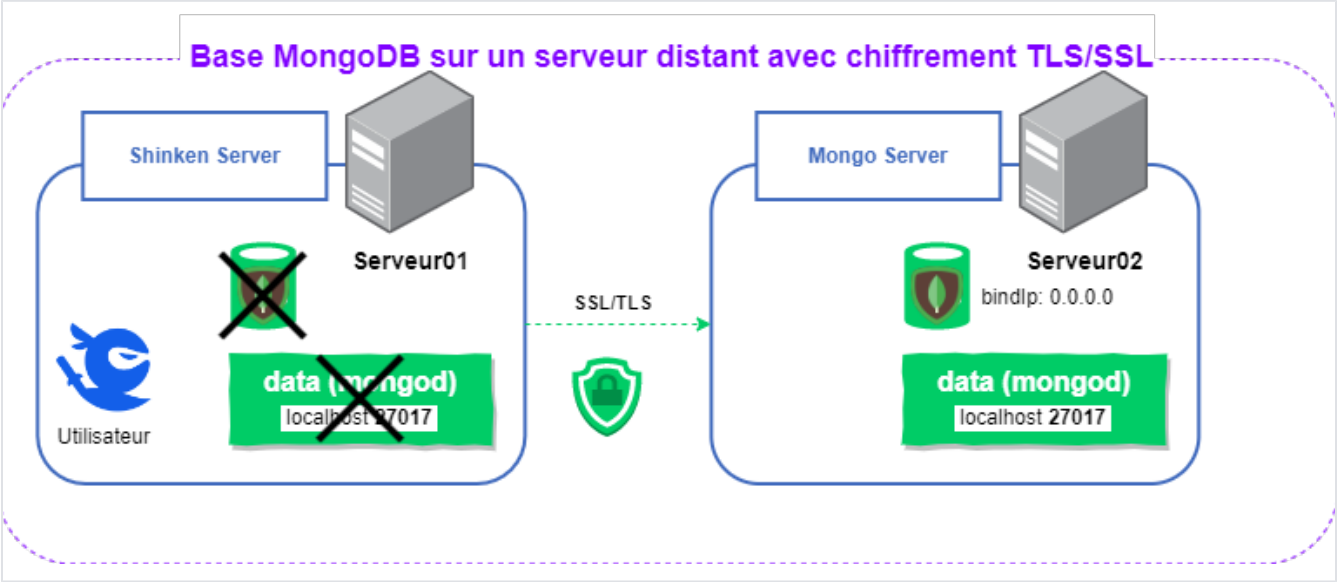
Options d'authentification



```
[root@serveur01 ~] shinken-command  --mongo-host 127.0.0.1  --mongo-port 27017  --mongo-username shinken --mongo-password shinken
```

Option	Valeur par défaut	Description
--mongo-username *ARG*	---	Utilisateur pour l'authentification avec mot de passe.
--mongo-password *ARG*	---	Mot de passe de l'utilisateur pour l'authentification avec mot de passe. À utiliser en complément de l'option --mongo-username. Si l'option **--mongo-password** est utilisée, le mot de passe risque d'être visible dans l'historique des commandes (*via la commande* ***history***). Pour éviter d'exposer le mot de passe, il est possible d'utiliser cette commande uniquement avec l'option **--mongo-username**. Un prompt interactif apparaîtra alors pour demander le mot de passe. Pour automatiser les commandes dans un script, il est possible de rediriger le contenu d'un fichier contenant le mot de passe *(par exemple : --mongo-password $(cat my_file_with_password)*).

Options SSL/TLS




```
[root@serveur01 ~] shinken-command  --mongo-host serveur02  --mongo-port 27017  --mongo-ssl-ca-file /etc/shinken/certs/mongo/ca.pem --mongo-ssl-pem-key-file  /etc/shinken/certs/mongo/client.pem
```

Option	Valeur par défaut	Description
--mongo-ssl	---	Active SSL/TLS pour les communications avec la base MongoDB.

	--mongo-ssl-ca-file *ARG*	---	Chemin vers le fichier de l'autorité de certification (*CA*) utilisé pour vérifier le certificat SSL de MongoDB. À utiliser en complément de l'option --mongo-ssl.
	--mongo-ssl-pem-key-file *ARG*	---	Chemin vers le fichier contenant le certificat SSL du client. À utiliser en complément de l'option --mongo-ssl.
	--mongo-ssl-pem-key-password *ARG*	---	Mot de passe du certificat SSL du client. À utiliser en complément de l'option --mongo-ssl.
	--mongo-ssl-crl-file *ARG*	---	Chemin vers le fichier CRL (*liste de révocation*) des certificats SSL à rejeter. À utiliser en complément de l'option --mongo-ssl.
	--mongo-ssl-allow-invalid-hostnames	---	Accepter le certificat SSL de MongoDB même si le nom d'hôte du certificat ne correspond pas à celui du serveur. À utiliser en complément de l'option --mongo-ssl.
	--mongo-ssl-allow-invalid-certificates	---	Accepter le certificat SSL de MongoDB même s'il est invalide, par exemple expiré. À utiliser en complément de l'option --mongo-ssl.

Exemples

```
$ shinken-protected-fields-encryption-enable

This command will enable encryption and restart the synchronizer to encrypt the protected fields.

Checking consistency between the synchronizer configuration file and the currently running configuration... DONE

You can review the list of Shinken properties which will be protected using the following command :

        shinken-protected-fields-data-manage

You can remove or add substrings to that list and review which Shinken properties would become protected or unprotected
before committing using the following commands or a combination :

        shinken-protected-fields-data-manage --add substring1 --add substring2 ...

        shinken-protected-fields-data-manage --remove substring1 --remove substring2 ...


        Are you sure you want to proceed and enable encryption (if you answer negatively you can still re-run this command later) ? (y/N) y

In order to enable encryption, an secret key is required.
It will be generated now, but it can be changed at a later stage, assuming this one is still available.

This key must have a name in order to identify it easily if you have several of them.
Enter your key name: secret key

Enabling encryption with key named 'secret key'...


Now stopping the Synchronizer...  OK

Encryption enabled

Now restarting the Synchronizer...  OK

Your protected data is now encrypted, using the key generated with the key name you provided : secret key

 * You now need to make a backup of your key using the following command command :

      shinken-protected-fields-keyfile-export

 * NOTE : If you lose your key, you won't be able to restore a backup and you will have to contact your support.
```

Si le chiffrement est déjà activé, le lancement de la commande **shinken-protected-fields-encryption-enable** expliquera qu'une clé est déjà activée et comment migrer la base vers une nouvelle clé de chiffrement.

```
$ shinken-protected-fields-encryption-enable

This command will enable encryption and restart the synchronizer to encrypt the protected fields.

Checking consistency between the synchronizer configuration file and the currently running configuration... DONE
Encryption already enabled with the key named secret key
Nothing to do.

However if you want to encrypt your protected fields with a new key, please use the following command :
        shinken-protected-fields-keyfile-migrate
```