

Pack MongoDB

Sommaire

- [Introduction](#)
- [Configuration de la connexion SSH](#)
 - [Dans la configuration de l'hôte](#)
 - [Côté client \(serveur supervisé\)](#)
 - [Côté Shinken \(serveur Poller\)](#)
- [Comment utiliser le pack MongoDB](#)
 - [Modèles d'hôtes fournis par le pack](#)
 - [Via l'interface de configuration](#)
 - [Via un fichier de configuration](#)
- [Authentification de la base Mongo](#)
- [Résumé des checks](#)
- [Personnaliser les seuils de Warning et Critique](#)
 - [Changer les seuils pour un seul hôte](#)
 - [Changer les seuils pour tous les hôtes qui utilisent l'un des modèles du pack MongoDB](#)
- [Version des scripts livrés](#)

Contexte

Ce guide vous permettra d'installer Shinken Entreprise sur un serveur Linux.

Une fois le guide d'installation suivi, vous aurez rapidement accès à l'interface de Configuration et de Visualisation de Shinken dans une architecture par défaut, c'est-à-dire sur un serveur simple, sur lequel tous les démons seront activés.

Si votre serveur fait parti d'une architecture distribuée, il vous faudra passer à la phase de configuration de vos démons

- noms et IP des serveurs,
- royaume,
- spare,
- Tag des Pollers,
- rétention de données...

En ce qui concerne la procédure de mise à jour, le script "d'update" vous permettra de mettre à jour votre serveur Shinken de manière complète, même si quelques démons sont seulement activés.

La configuration de votre serveur Shinken ne sera pas modifiée.



Important

Lors de l'installation de Shinken Entreprise, le système de gestion de base de données orientée documents **MongoDB** est mis en place avec la version **v3.0.15**. Ce système de base de données permettra le bon fonctionnement de l'interface de Configuration et de Visualisation. Utilisé avec une base MongoDB, **Graphite**, quant à lui, est un outil pour stocker les métriques de vos sondes.

Pour ne pas créer de dysfonctionnement, **nous vous demandons de ne pas mettre à jour MongoDB / Graphite**. Veuillez simplement laisser en place les versions fournies par nos services.

⚠ Afin de prévenir tout risque, les démons Shinken Entreprise refuseront de démarrer si la version installée de **MongoDB** n'est pas celle préconisée.

⚠ Si une version différente de **MongoDB** est déjà présente sur le serveur, l'installation sera interrompue

⚠ Si vous faites une mise à jour de Shinken Entreprise depuis une version antérieure à la 2.6.1 et que la version de **MongoDB** installée n'est pas la 2.6.9, la mise à jour sera interrompue

Historique de l'installeur

Concernant l'installeur à utiliser, il faut prendre le dernier en date.

Voici l'historique des installeurs de cette version:

Ajout (mots clefs)	Date	Nom de l'installeur	Modification par rapport à la version précédente
----------------------	------	---------------------	--

RC006.02	23 Mai 2022	shinken-enterprise_V02.08.02-RC006.02_US /FR_Linux_FULL_2022-04-14.tar.gz	Version d'origine (non finale pour l'instant)
RC007	29 Mai 2022	shinken-enterprise_V02.08.02_US/FR_Linux-PACKAGE-007_2022-06-15.tar.gz	Voir release note. Ajout du paramètre "--ignore-pre-setup-non-blocking-errors" (permet de passer outre les erreurs non importantes pour le bon fonctionnement de Shinken)

Installation de Shinken Entreprise

Prérequis

Concernant l'OS

Environnement requis : [RHEL/Centos 6.10, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9](#) [64bits]

Shinken Entreprise a choisi les distributions produites par Red Hat : **Red Hat Enterprise Linux (RHEL)** et **CentOS (Community enterprise Operating System)**. Ces distributions Linux, principalement destinées aux serveurs, sont stables, performantes et compatibles avec une très grande majorité des environnements professionnels.

- **Red Hat Enterprise Linux** est la distribution référente dans l'écosystème professionnel Linux
- **CentOS** est une distribution dont tous ses paquets, à l'exception du logo, sont des paquets compilés à partir des sources de la distribution **Red Hat Enterprise Linux**
 - Elle est donc quasiment identique à celle-ci et se veut 100 % compatible d'un point de vue binaire

Concernant le support de ces distributions: _

Distribution	Version distribution	Date support éditeur distribution	Gérée actuellement par Shinken	Sera gérée dans les prochaines versions de Shinken	Recommandations Shinken
RedHat	6.10	nov 2020 <i>(plus supportée)</i>	Oui	Non	Ne pas installer sur cet OS, et migrer les installations existantes en RedHat 7.
	7.2 7.9	juin 2024	Oui	Oui	Mettez à jour en RedHat 7.9 si possible.
	8	mai 2029	Pas encore	Oui	Sera gérée dans une prochaine version, utilisez RedHat 7.9 à la place.
CentOS	6.10	nov 2020 <i>(plus supportée)</i>	Oui	Non	Ne pas installer sur cet OS, et migrer les installations existantes en CentOS 7.
	7.2 7.9	juin 2024	Oui	Oui	Mettez à jour en CentOS 7.9 si possible.
	8	décembre 2021 <i>(fin de vie proche)</i>	Non	Non	La version 8 a été annoncée comme arrêtée fin 2021 (https://wiki.centos.org/About/Product) et ne sera donc pas gérée.

Concernant la transformation de la Centos en Centos Stream (Béta de la Redhat)

Redhat a changé sa politique concernant la Centos, qui devient maintenant une version Béta à la RHEL.

Là où précédemment elle était une recompilation à l'identique d'une RHEL, elle est désormais une distribution sans version fixe (dite "rolling release") en amont de RHEL :

- qui sert à RedHat afin de tester des nouvelles versions de paquets, avant leur sélection si les tests sont fonctionnels dans la RHEL.
- Elle récupère ainsi le rôle qu'avait la Fedora avant elle.
- Elle ne nous semble donc pas viable pour une utilisation professionnelle en production.

Il y a donc 2 axes possibles :

- Vous restez sur Centos 7, le temps qu'un remplaçant se démarque.
 - Le support de la Centos 7 va jusqu'en Juin 2024, ce que laisse une marge conséquente.
 - Dès qu'un remplaçant sera suffisamment stable, nous intégrerons cette OS dans nos mécanismes d'installation / mise à jour / patch
- Passer vos Centos en Redhat.

Notre recherche du remplaçant de Centos

Pour le remplacement de Centos 7, pour l'instant nous attendons qu'une distribution fasse consensus sur le marché afin de partir sur une distribution pérenne, pour les prochaines années. Actuellement, nous suivons de prêt l'évolution de deux distributions, clones de Centos:

- Rockylinux (par le créateur initial de Centos)
- AlmaLinux (par la société CloudLinux)

Transformer une Centos en Redhat

RedHat a mis à disposition un outil de conversion CentOS 7.9 vers RedHat 7.9 qui est [convert2rhel](#).

- Il vous faudra posséder un compte et une licences valide pour procéder à cette conversion.

Suite à nos tests, la conversion d'un serveur avec Shinken déjà installé est fonctionnelle et n'a aucun impact sur notre outil.

Concernant la Redhat



Attention - Enregistrement Redhat

Lors d'une installation de distribution Redhat Enterprise Linux (commerciale), il faut rattacher votre souscription Redhat à votre système.

Voici les commandes à utiliser depuis le serveur:

```
1/ subscription-manager register  
( -> Nom d'utilisateur / mot de passe )
```

et il faut également l'attacher à l'OS en cours:

```
2/ subscription-manager attach
```

Yum pourra alors être utilisé correctement car l'abonnement sera valide (et donc Shinken pourra être installé)

Prérequis CPU

La langage de programmation python n'ayant actuellement pas la possibilité d'utiliser plusieurs processus (*multithreading*), chaque démon utilise un ou plusieurs processus :

- Synchronizer
 - 1 pour l'UI de configuration
 - 1 pour le daemon
- Broker
 - 1 / Web UI
 - 1 pour le daemon
- Arbitrer
- Receiver
- Reactionner
- Scheduler
- Poller

Extraction du package et installation

Installation automatique :

Il faut être loggué en tant que root,

```
$id  
uid=0(root) gid=0(root)
```

Et que le umask du compte root soit à 0022

```
$umask 0022
```

« Décompresser » le package qui vous a été transmis :

- `tar xzvf shinken-enterprise_V02.08.XX- LANGUAGE .tar.gz`
- Cela vous créera un répertoire **shinken-entreprise** contenant le script d'installation et les dépendances nécessaires à l'installation.

Déplacez-vous à la base du répertoire **shinken-entreprise** (`cd shinken-entreprise_V02.08.XX- LANGUAGE`) et exécutez le script :

```
./install.sh
```

Il installera **Shinken Enterprise** et ses composants automatiquement.

- Le SDK VMWare nécessaire aux checks est maintenant installé automatiquement (*nécessitait une installation manuelle dans les versions précédentes*).

- Pour obtenir plus d'informations sur la consommation des démons et améliorer le fonctionnement de Shinken lorsqu'il est installé sur une machine virtuelle VMWare, le paquet "open-vm-tools" doit être installé manuellement :

```
yum install open-vm-tools
```

Vérification du bon fonctionnement

Vérification :

Pour vérifier que Shinken Entreprise est bien installé, configuré et fonctionnel, lancez la commande :

```
shinken-healthcheck
```

Votre licence Shinken Entreprise n'est pas installée, c'est normal, dirigez-vous dans la section [Clé de licence](#) plus bas sur cette page afin d'installer votre clé.



L'installation complète fera sur le même serveur :

- L'installation du moteur Shinken Entreprise, des modules et des dépendances.
- L'activation de **tous** les démons (*Synchronizer, Arbiter, Scheduler, Poller, Reactionner, Broker, Receiver*).

Pour une installation distribuée, voir la page [Architecture Distribuée](#)

Accès aux interfaces web

Interface Utilisateur (UI) de Configuration

Une fois Shinken Entreprise installé, pour accéder à l'UI de configuration, vous devez pointer votre navigateur Web vers l'adresse affichée durant l'installation.

- Par défaut, l'interface de configuration est accessible sur le port dédié **7766** (*via le protocole HTTP*). Par exemple : <http://192.168.0.1:7766>



L'adresse IP (*ou FQDN si votre résolution DNS est opérationnelle*) correspond à votre serveur hébergeant le démon **Synchronizer**.

Voir la page [Paramétrage de l'interface de Configuration](#) pour plus d'information.

Interface Utilisateur (UI) de Visualisation

Une fois Shinken Entreprise installé, pour accéder à l'UI de visualisation, vous devez pointer votre navigateur Web vers l'adresse affichée durant l'installation.

- Par défaut, l'interface de configuration est accessible sur le port dédié **7767** (*via le protocole HTTP*). Par exemple : <http://192.168.0.1:7767>



L'adresse IP (ou FQDN si votre résolution DNS est opérationnelle) correspond à votre serveur hébergeant le démon **Broker**.

Voir la page [Paramétrage de l'interface de Visualisation](#) pour plus d'information.

Présence du guide d'utilisateur dans le package

Le guide d'utilisateur (en français) est maintenant intégré au package d'installation.

Vous pouvez le retrouver à l'intérieur de shinken-enterprise_V02.08.XX- **LANGUAGE** .tar.gz dans le répertoire /tools/documentation/ui-visualisation/

Il vous suffira qu'un utilisateur, via son navigateur internet, ouvre le fichier "index.html" afin de pouvoir parcourir le guide d'utilisateur, contenant la documentation liée à l'UI de visualisation.

Installation du script dump_performance.sh

Ce script permet de stocker les informations systèmes de votre serveur Linux en local.

- Ces sont des informations utiles très utiles en cas de debug avec le support Shinken.
- Sa mise en place est décrite dans la page [dump_performance.sh \(top / ps / cpu / kernel / healthcheck \)](#)

Installation (Mode avancé)

Choisir les démons activés pendant le processus d'installation

Il faut être logué en tant que root,

```
$id
uid=0(root) gid=0(root)
```

Et que le umask du compte root soit à 0022

```
$umask 0022
```

« Dé-tarez » le package qui vous a été transmis :

- tar zxvf shinken-enterprise_V02.08.XX- **LANGUAGE** .tar.gz
- Cela vous créera un répertoire **shinken-entreprise** contenant le script d'installation et les dépendances nécessaires à l'installation.

Déplacez-vous dans le répertoire (**cd shinken-entreprise_V02.08.XX- LANGUAGE**) et lancez la commande **./ install.sh** mais avec des options basées sur les démons que vous souhaitez activer :

- **--pollernode** : active le démon Poller (*dédié au lancement des checks*)
- **--reactionnnode** : active le démon Reactionner (*dédié au lancement des notifications*)
- **--schedulernode** : active le démon Scheduler (*planificateur des checks*)
- **--arbiternode** : active le démon Arbiter (*rôle de distribution centrale*)
- **--receivernode** : active le démon Receiver (*reçois les checks passifs*)
- **--synchronizernode** : active le démon Synchronizer (*gère la configuration*)
- **--brokernode** : active le démon Broker (*export des données*)



Exemple

Vous pouvez par exemple installer Shinken Enterprise et activer directement le Scheduler et le Poller en même temps en tapant la commande

```
./install.sh --schedulernode --pollernode
```

Pour vérifier que les démons sélectionnés de Shinken Entreprise sont bien mis à jour, configurés et fonctionnels, lancez la commande :

```
shinken-healthcheck
```

Shinken-healthcheck vérifiera alors que Shinken Entreprise est bien configuré et en cours d'exécution (*seulement pour les démons installés*)

- **Après l'installation initiale, si vous souhaitez activer/désactiver des démons**, utilisez la commande d'activation détaillée dans le chapitre [Activation/Désactivation des démons](#) .
- **Lors d'une mise à jour, le script update.sh prend en compte les démons qui sont activés ou non.**

Les différents addons sont automatiquement activés lors de l'installation:

- **nagvis-shinken-architecture**: Activé lors d'une installation d'un Arbiter (*plus de détails dans la page de documentation dédiée: [Activation et configuration de la fonctionnalité](#)*)
- **nagvis**: Activé lors d'une installation d'un Broker (*plus de détails dans la page de documentation dédiée: [NagVis \(Addon \)](#)*)

Mise en place du chiffrement

Vous pouvez mettre en place le [Chiffrement des données sensibles](#) de façon automatique au moment de l'installation.



Si vous n'avez jamais activé le chiffrement des données sensibles, nous vous conseillons de procéder à l'installation sans activer le chiffrement et de découvrir la fonctionnalité par la lecture du chapitre [Chiffrement des données sensibles](#).

Une clé de chiffrement sera alors générée lors du processus d'installation et la base de données du Synchronizer sera chiffrée.

Pour cela, lancez la commande suivante :

```
./install.sh --activate-encryption <nom de clé> --disable-important-notices-user-input
```

- **--activate-encryption** permet d'activer le chiffrement. Le nom de la clé est optionnel toutefois il vous sera demandé lors de l'exécution du programme d'installation si vous ne le précisez pas.
- **--disable-important-notices-user-input** permet de désactiver les prompts vous demandant confirmation avant de continuer le processus.
⚠ **Il vous est cependant fortement conseillé de lire les informations fournies lors de l'installation.**



La mise en place automatique du chiffrement nécessite dans tous les cas d'effectuer l'export et la sauvegarde de la clé générée lors du processus. Veuillez consulter [shinken-protected-fields-keyfile-export](#) pour plus d'informations.

Shinken-healthcheck vous permettra de vérifier la bonne configuration des démons et du chiffrement.

Mise en place d'un serveur sans connexion internet

Dans le cas d'un serveur qui n'a pas de connexion internet, il faut lancer l'installateur avec le paramètre suivant:

- **--disable-add-public-epel** : permet de ne pas installer le repository epel sur le serveur
- **--disable-epel** (*DEPRECATED*): ancien nom du paramètre, sera supprimé dans le futur



Accès à un repository yum

Il est à noter que le serveur doit avoir un accès à un repository yum valide (ayant également les paquets présents dans epel) en cas de conflits de versions des rpm entre ce que propose l'installateur et ce qui est déjà installé sur le serveur.

Faire l'installation sur un serveur avec des repository internes (non publics) fixés sur une version précise

Dans le cas d'un serveur qui n'a accès qu'à des "**repository**" internes qui ne sont pas forcément synchronisés sur les dernières versions des "**repository**" centos/redhat officiel, le comportement de base de l'installateur et le script d'update sont de mettre à jour tous les packages si une mise à jour est possible, mais ceci peut entraîner des problèmes si l'installateur a une mise à jour à faire trop récente par rapport à ce qu'il a de disponible dans ses "**repository**".

Dans ce cas, il faut lancer avec l'option qui demande à ne pas mettre à jour les paquets s'ils sont déjà installés :

- **--package-update-only-on-conflict** : permet de ne pas chercher à mettre à jour les paquets déjà installés et ainsi tente d'éviter d'installer des paquets trop à jour par rapport au "**repository**" interne qui n'est pas à jour.



Accès à un repository yum

Il est à noter que le serveur doit tout de même avoir accès à un "**repository**" valide, et des conflits de paquets peuvent survenir dans le cas de nouveaux paquets installés et que dans ce cas seul yum requêtant les "**repository**" peut les résoudre (*arrive dans le cas de paquets de l'installateur trop à jour par rapport à ce qui est disponible dans le repository*).

Faire l'installation sur un serveur RedHat non enregistré sur les repository RedHat

Si un serveur RedHat a un accès uniquement à des "**repository**" locaux, il ne sera pas enregistré directement chez RedHat. La vérification de l'installateur et du script d'update sur les RedHat se base sur la vérification de cette connexion afin de déterminer si le serveur a bien accès aux "**repository**". Ici cette vérification va bloquer l'installation alors que le serveur a bien accès à des "**repository**" locaux. Il faut alors utiliser l'option suivante :

- **--skip-redhat-subscription-check**: permet de ne pas lancer la vérification de la souscription du serveur auprès de RedHat (*qui doit avoir tout de même accès à des repository locaux*).

Faire l'installation sur un serveur qui n'a accès à aucun repository (ni interne, ni public)

Dans le cas d'un serveur qui est totalement isolé du moindre "repository", il faudra passer par plusieurs tentatives d'installations afin d'avoir à chaque fois les erreurs "yum" qui donnent la liste des paquets manquants.

Le principe général sera:

- récupérer la liste des paquets manquants ou pas dans la bonne version depuis l'erreur de l'installateur
- depuis un serveur qui lui a accès à un repository, récupérer les paquets rpm manquants
- transférer les paquets rpm sur le serveur Shinken à installer
- installer les paquets rpm
- relancer l'installation, et en cas de problèmes de nouveaux paquets manquants, recommencer la procédure jusqu'à ce que l'installation finisse correctement

Dans le cas où l'installateur sorte l'erreur:

```
error: Failed dependencies:
  audit-libs = 2.4.1-5.e17 is needed by audit-libs-python-2.4.1-5.e17.x86_64
  libavahi-client.so.3()(64bit) is needed by cups-client-1:1.6.3-22.e17.x86_64
```

Alors il faut lancer sur un serveur ayant accès aux repository (centos/redhat et epel):

```
yum install yum-plugin-downloadonly
yum install -y --downloadonly --downloadaddr=. audit-libs-python cups-client
```

Ceci va télécharger localement plusieurs fichiers rpm qu'il faut transférer sur le futur serveur Shinken et lancer une simulation de l'installation des paquets:

```
rpm -Uvh --test *rpm
```

Si la commande ne retourne pas d'erreur, lancer l'installation des paquets rpm:

```
rpm -Uvh *rpm
```

Il faut alors relancer l'installation de Shinken qui devrait arriver à une étape suivante, et recommencer si certains paquets manquent encore.



Repository du serveur de téléchargement

Il est à noter que le serveur de téléchargement ne doit pas avoir de Shinken installé afin de ne pas avoir les paquets demandés déjà installés (sinon la commande yum ne les téléchargera pas).

Il doit également avoir accès aux repositories:

- standard centos/redhat
- EPEL

Clé de licence Shinken Enterprise

Une fois Shinken Enterprise installé, la commande **shinken-healthcheck** lancée depuis votre serveur Arbiter affichera un message d'erreur au sujet de la licence:

La licence par défaut installée est une licence d'essai. Vous ne pourrez placer en supervision qu'un très faible nombre d'hôtes.

Le service Commercial de Shinken Enterprise a dû vous envoyer une licence nominative vous permettant d'utiliser pleinement le produit.

? Unknown Attachment

La licence est un fichier qui a le nom suivant : **user.key** et cette licence est nominative et limitée dans le temps.

Pour l'installer, rien de plus simple, il suffit de :

- Placer ce fichier sur le serveur hébergeant l'Arbiter et sur les serveurs hébergeant le ou les UIs de Visualisation , dans le chemin suivant : **/etc/shinken/user.key**
- Redémarrez alors Shinken Enterprise via la commande : **service shinken restart**

Relancez alors la commande **shinken-healthcheck** le message d'erreur de licence doit avoir disparu et voici un exemple d'information de licence valide :

? Unknown Attachment

Si vous n'avez pas de clé de licence ou que celle-ci a expiré, contactez-nous : contact@shinken-solutions.com

Résolution des problèmes liés à l'installation

Lors de l'installation des dépendances, si une machine n'est pas connectée à internet ou connectée à un repository privé, il arrive que les scripts d'installation ou de mise à jour échouent.

Dans ce cas, des fichiers sont créés dans le "home" de l'utilisateur avec lequel est effectuée l'installation/mise à jour. Ces fichiers contiennent plus de détails sur les erreurs rencontrées et peuvent être envoyés à votre contact de support Shinken Enterprise pour la correction du problème.

skip-redhat-subscription-check

Pour chaque installation, un dossier est créé dans `~/shinken/versions_et_patch_installations/` et nommé comme suit:

```
YYYY-MM-DD-HHhMMmSS-install-VXX.XX.XX
```

Ce dossier contient les fichiers suivants:

- Affichage du script d'installation (*installation seulement*): `shinken.enterprise.install.log`
- Détails d'installation des paquets: `shinken.enterprise.install.detail.log`
- Nettoyage de la configuration: `sanatize.update.log`
- Affichage du script de mise à jour (*mise à jour seulement*): `shinken.enterprise.update.log`
- Backup de la configuration et données utilisateur (*mise à jour seulement*)

En cas de soucis avec les installations de packages via yum, les erreurs seront présentes dans les fichiers:

- `/tmp/install.txt`
- `/tmp/install_bogus.txt`

Cas spécifique de la mise à jour d'un cluster Mongo

Dans la version V02.07.00, la base MongoDB est mise à jour. Lorsque MongoDB a été configuré pour fonctionner en tant que cluster, le comportement du script de mise à jour de Shinken Enterprise a été modifié pour prendre en compte cette configuration particulière.

Des explications détaillées sont présentes dans la page de documentation dédiée: [Inférieur à V02.07.00 - Montée de version en MongoDB 3.0 \(réalisée automatiquement sous conditions \)](#)

Cas spécifique d'une machine n'ayant même pas accès à des repository redhat/centos/epel

Dans des cas bien précis un serveur Shinken peut ne pas avoir accès à des repository redhat/centos/epel. Il est important de noter qu'une connexion internet n'est pas nécessaire, juste l'accès à des repository locaux est totalement suffisant.

Si même l'accès à des repository locaux n'est pas possibles, l'installation peut échouer même avec l'usage des options **--disable-add-public-epel** et **-package-update-only-on-conflict** car il peut avoir besoin d'installer des nouveaux paquets dont les dépendances ne seront pas disponibles.

Il est cependant possible de contourner le problème en installant manuellement ces dépendances.

Attention: le non accès à des repository doit alerter sur la sécurité

Attention cependant: s'il est tout à fait possible de contourner le problème et d'installer Shinken Enterprise sur le serveur, ne pas avoir accès aux repository systèmes des bases de votre distribution doit être pesé avec beaucoup de soin:

=> ceci signifie que ce serveur ne pourra pas être maintenu à jour en terme de correction de faille de sécurité (comme par exemple les dernières failles sudo, ssh ou bash par exemple).

=> ceci doit être une politique connue et acceptée par l'équipe en charge du serveur, ainsi que les risques encourus.

Une fois une installation bloquée sur le serveur, vous allez avoir la liste des paquets manquants, comme par exemple:

```
-> Running transaction check
--> Package at.x86_64 0:3.1.13-20.el7 will be installed
--> Package audit-libs-python.x86_64 0:2.4.1-5.el7 will be installed
-> Processing Dependency: audit-libs = 2.4.1-5.el7 for package: audit-libs-python-2.4.1-5.el7.x86_64
```

Dans notre cas, il manque les dépendances **at** et **audit-libs-python** pour le paquet installé **audit-libs**. Il faut donc récupérer leurs **rpm**, ainsi que ceux de leurs dépendances :

- depuis un serveur ayant un accès aux repository (publiques ou internes).
- il ne faut pas que ce serveur ait déjà d'installé les paquets recherchés :
 - idéalement un serveur minimal sans Shinken d'installé dessus donc, sur ce serveur, il faut installer une extension à yum qui va permettre de juste télécharger les dépendances:

```
yum install yum-plugin-downloadonly
```

Puis nous demandons:

- les deux paquets que nous recherchons
- plus le paquet déjà installés (*audit-libs*), au cas où une mise à jour serait disponible.

Ils seront téléchargés localement:

```
yum install -y --downloadonly --downloadonly --downloadonly=. at audit-libs-python audit-libs
```

Dans notre cas, ceci télécharge deux paquets (*ils n'avaient pas de dépendances et le paquet audit-libs était déjà à jour*), *at-3.1.13-24.el7.x86_64.rpm* et *audit-libs-python-2.8.5-4.el7.x86_64.rpm*.

Il faut alors les envoyer sur le serveur qui n'a pas de repository et les installer localement:

```
rpm -Uvh at-3.1.13-24.el7.x86_64.rpm audit-libs-python-2.8.5-4.el7.x86_64.rpm
```

Il faut ensuite relancer l'installation, et recommencer l'opération si de nouveaux paquets manquants sont détectés.