



## Principes généraux

Shinken Enterprise comprend 7 démons.

Suivant les principes Unix : un outil, une tâche, Shinken Enterprise a une architecture où chaque partie est isolée et se connecte aux autres par une interface standard HTTP.

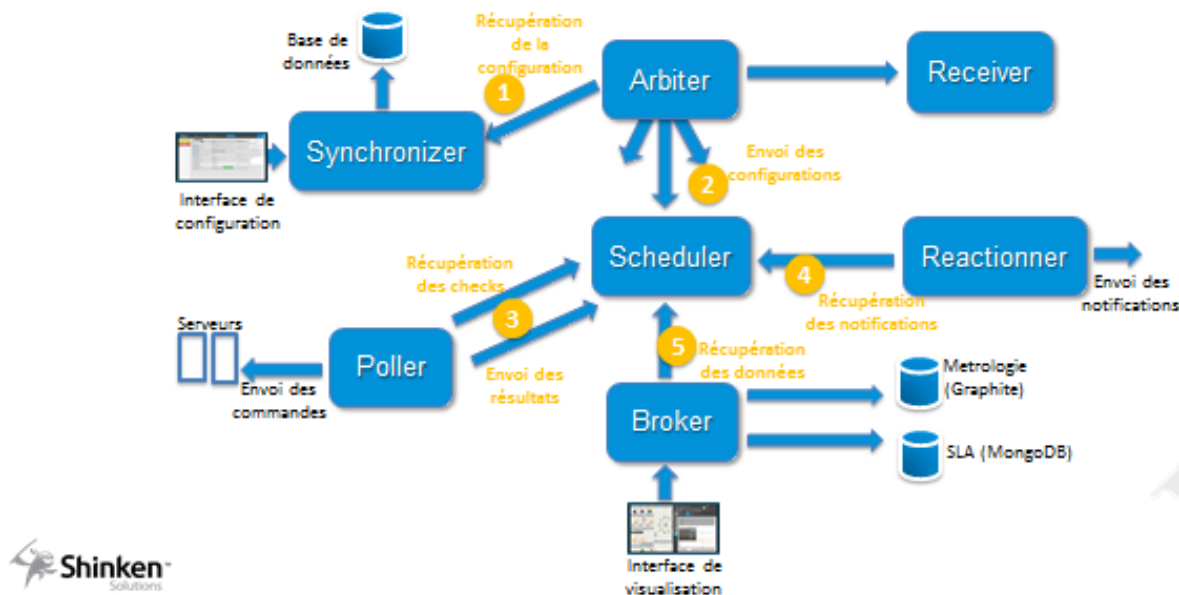
Basé sur un back-end HTTP, cela vous permettra de construire une architecture distribuée et hautement disponible très simplement.

Voici la table des différents démons, leurs ports et leur rôle respectif .

Daemon	Listening Port	Protocol	Role
Synchronizer	7765	HTTPS	Gère l'édition de la configuration
Arbiter	7770	HTTPS	Distribue la configuration à tous les démons
Poller	7771	HTTPS	Exécute les checks
Scheduler	7768	HTTPS	Analyse les statuts et contextes des éléments
Reactionner	7769	HTTPS	Envoie les notifications
Receiver	7773	HTTPS	Reçoit les résultats des checks externes
Broker	7772	HTTPS	Centralise et exporte les données

L'architecture est totalement flexible et scalable. Pour améliorer les capacités de Shinken Enterprise , augmenter le nombre de démons ayant le même rôle est la meilleure approche.

## Architecture



## Load balancing automatique

Distribution des hôtes à travers les schedulers

Shinken Enterprise est capable de couper la configuration en plusieurs parties et les distribuer aux Schedulers .

Le load balancing est fait automatiquement : l'administrateur n'a pas besoin de se souvenir quel hôte est lié à tel autre pour créer les packs

La répartition est basée sur les hôtes : cela veut dire que tous les checks associés à un hôte seront dans le même Scheduler que l'hôte . Cela signifie que l'administrateur n'a pas besoin de connaître toutes les relations entre éléments comme les parents,, dépendances d'hôtes ou dépendances de checks : Shinken Enterprise est capable de lire ses relations et de rassembler tous les éléments liés dans la même partition.

Cette action se fait en 2 parties :

- création de partitions indépendantes pour les éléments
- copie des partitions pour créer N configurations pour N Schedulers

## Création de partitions indépendantes

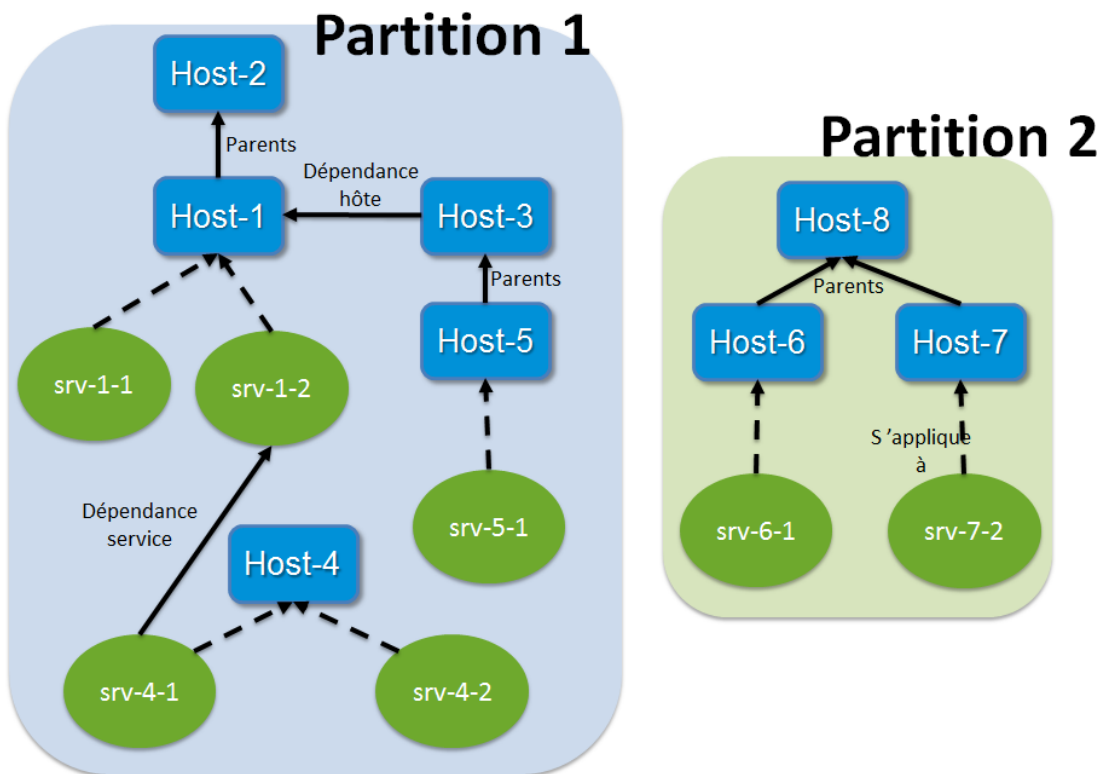
L'action de hachage se fait en se basant sur 2 éléments : les hôtes et les checks. Les checks sont liés à l'hôte donc ils seront dans la même partition.

D'autres relations sont prises en compte :

- Liaisons réseau pour un hôte (comme un serveur distant et son routeur).
- Dépendances logiques.

Shinken Enterprise regardent toutes les relations et crée un graphe avec. Un graphe est une partition de relations.

Illustration :



Dans cet exemple, nous avons 2 partitions:

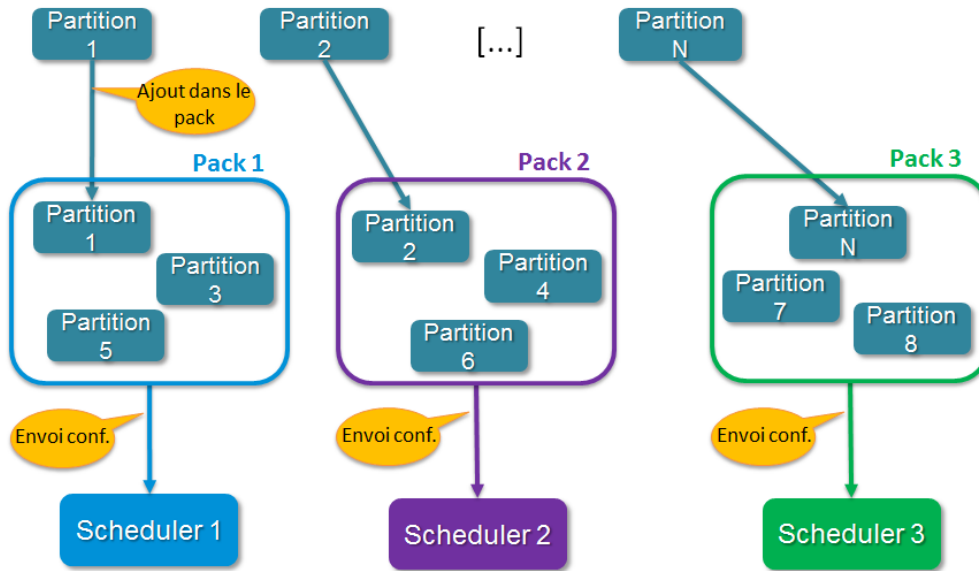
- Shard 1: Host-1 au host-5 et tous leurs checks
- Shard 2: Host-6 au Host-8 et tous leurs checks

## L'agrégation des partitions dans les schedulers

Quand toutes les partitions sont créées, l'Arbiter les agrègent dans N configurations si l'administrateur a défini N Schedulers actifs (sans spare).

La répartition se fait sur un critère de poids des Schedulers : plus le poids est élevé, plus il y a de packs .

Illustration :



## Envoi des configurations vers des satellites

Une fois que toutes les configurations sont créées, l'Arbiter les envoie aux N Schedulers actifs .

Un Scheduler peut commencer à lancer des checks une fois qu'il reçu et chargé sa configuration sans avoir à attendre que TOUS les Schedulers soient prêts.

Pour des configurations plus importantes, avoir plusieurs Schedulers (même sur un seul serveur) est fortement recommandé car ils chargeront leur configuration beaucoup plus vite (nouvelle ou modification)

L'Arbiter crée également les configurations pour ses satellites (pollers, reactionners et brokers) avec les liens permettant de savoir où réaliser les tâches .

Après avoir envoyé les configurations, l'Arbiter commence à traiter les ordres (appelées commandes externes) des utilisateurs et est responsable de vérifier la disponibilité des satellites.

## La haute disponibilité

L'architecture de Shinken Enterprise est hautement disponible.

Un serveur peut crasher, une application également. C'est pour cela que les administrateurs ont des back up : ils peuvent recharger la configuration des éléments tombés .

A ce jour, un seul démon n'est pas encore doublé : l'Arbiter. Cela sera fait dans le futur. L'Arbiter vérifie régulièrement que tous les démons sont disponibles. Si un Scheduler ou un autre satellite est tombé, l'Arbiter envoie sa configuration au nœud spare défini par l'administrateur .

- Tous les satellites sont informés de ce changement, de façon à recevoir leur tâches du nouvel élément sans essayer de joindre le démon tombé .
- Si un nœud est perdu du à une coupure réseau, puis revient , l'Arbiter en prend note et demande au système de supprimer son ancienne configuration .

Les critères de disponibilité peuvent être modifiés dans les paramètres par défaut lorsqu'il s'agit d'une grosse installation car les Schedulers et Brokers peuvent être surchargés et du coup avoir des temps de réponse de disponibilité plus longs.

Les délais sont volontairement très courts pour de petites installations ( Voir paramètres de configuration des [Démons](#) pour plus d'information ).

## Distribution par Commande Externe

L'administrateur doit envoyer des ordres aux Schedulers (comme par exemple un nouveau statut pour un check passif).

Dans Shinken Enterprise, l'administrateur envoie uniquement l'ordre à l'Arbiter, c'est tout. Les commandes externes sont de 2 types :

- commandes qui concernent tous les Schedulers.
- commandes qui sont spécifiques à un seul élément (hôte/check).

Pour chaque commande, Shinken Enterprise détecte si c'est global ou particulier:

- si global, il envoie les ordres à tous les Schedulers.
- Si particulier, il détecte quel Scheduler gère l'élément concerné par la commande (hôte/check) et envoie l'ordre au bon Scheduler.

Dès réception de l'ordre par les Schedulers, il est appliqué.