

Modules de Sources (imports) et de Taggers (qualification)

Sommaire

Concept
Fonctionnement de la collecte d'une source jusqu'à sa sauvegarde en base

Concept

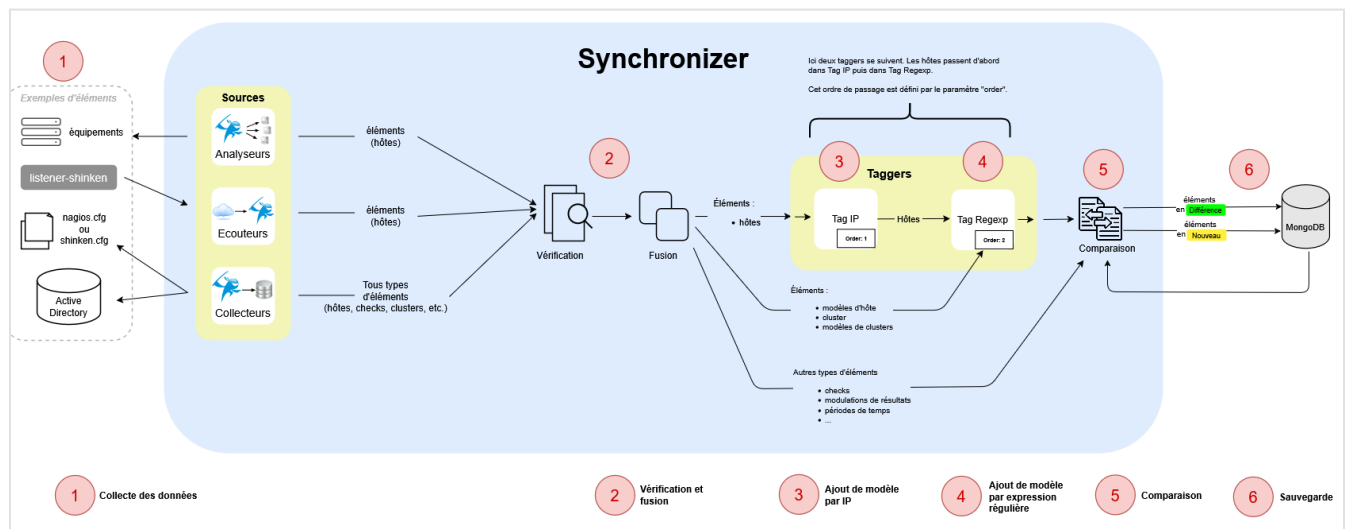
Le démon Synchronizer gère deux types de modules :

- Les Sources
- Les Taggers

Les modules de type Source permettent de détecter de nouveaux éléments. Ils servent à importer des éléments de différentes manières. Une liste d'exemple non exhaustive : import de fichiers de configuration (CFG), analyse d'un équipement, détection sur le réseau, interrogation d'un annuaire LDAP (voir la page [Les Sources](#)).

Les modules de type Tagger interviennent après l'import des Sources. Ils utilisent une règle (*plage d'IP, regex*) pour sélectionner des hôtes, clusters, modèles d'hôtes et modèles de clusters issu de l'import des Sources et modifier leurs propriétés (voir la page [Les Taggers](#)).

Fonctionnement de la collecte d'une source jusqu'à sa sauvegarde en base



Étape 1

Le **Synchronizer** collecte les éléments partiels des **Sources** suivant des intervalles de temps configurés.

Les sources "Analyseurs", "Ecouteurs" et "Collecteurs" vont interroger diverses applications et chargent les données détectées afin de générer des éléments partiels.

Étape 2

Le **Synchronizer** vérifie tous les éléments partiels importés. Il les fusionne s'ils possèdent au moins une clé de synchronisation commune.

En cas de conflit entre des éléments partiels, la propriété de la source ayant le numéro d'ordre le plus bas est utilisée (voir la page [Le mélange des sources & les clés de synchronisation \(sync-key\)](#)).

Étape 4

Les hôtes, clusters, modèles d'hôtes et modèles de clusters passent à travers un **Tagger par expression régulière**. Il compare une des propriétés de l'élément avec la règle `regex` définie dans sa configuration.

Si la propriété correspond, alors le Tagger modifie une propriété de l'élément.

Étape 5

Tous les éléments sont ensuite comparés avec ceux dans la base de données.

Les éléments détectés comme existants, mais dont certaines propriétés diffèrent, prennent le type `Différence`.

Étape 3

Les hôtes passent par le **Tagger IP**. Ce dernier compare l'adresse IP de l'hôte avec la plage IP définie dans le Tagger.

Si l'adresse IP correspond, le Tagger modifie une propriété de l'hôte.

Les éléments qui n'existent pas dans la base de données prennent le type `Nouveau`.

Étape 6

Tous les éléments sont **sauvegardés** dans la base de données.