

# Les Taggers

## Sommaire

### Concept

[Détails sur l'exécution des Taggers](#)  
[Définir ou configurer un Tagger](#)

## Concept

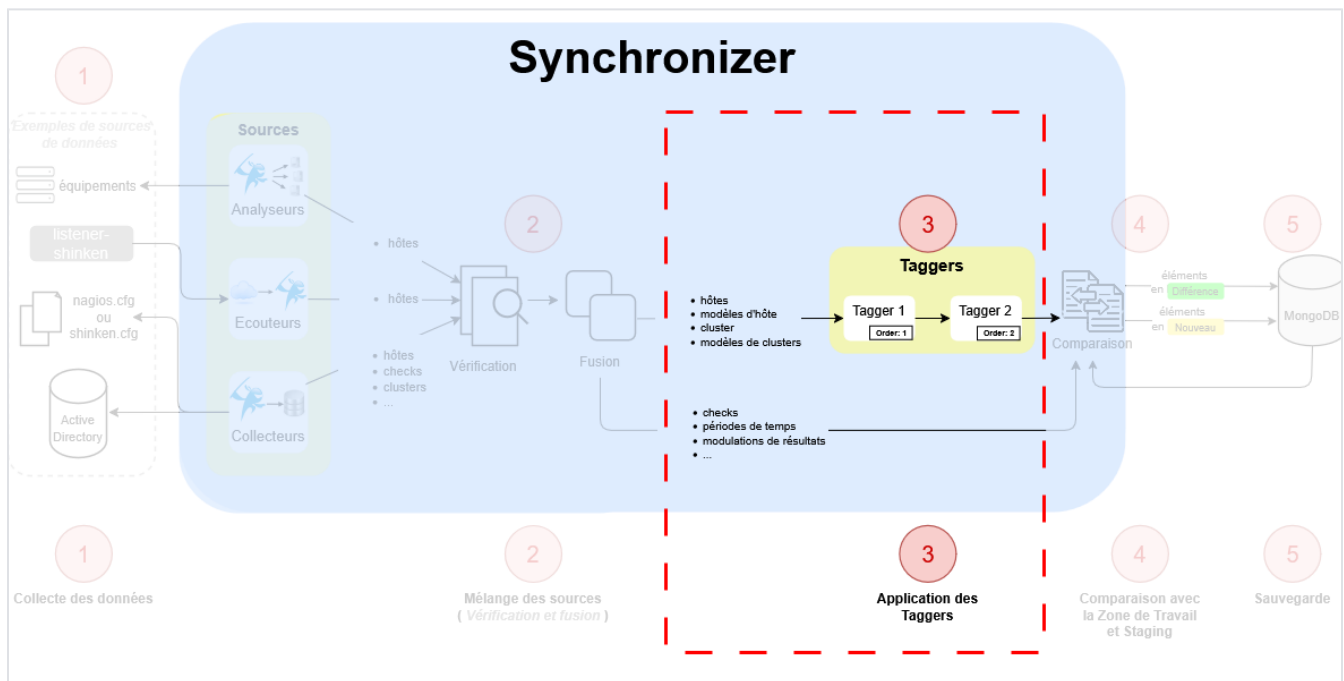
Les Taggers permettent d'effectuer des actions sur les éléments issus de l'import des Sources via la définition d'une règle.

Il est possible d'ajouter autant de Taggers que nécessaire.

- Les Taggers s'enchaînent les uns à la suite des autres dans l'ordre croissant de la propriété **order** ( définie dans la configuration de chaque Tagger ).
- Un Tagger peut surcharger les modifications effectuées par un Tagger précédent.
  - Exemple :
    - Tagger **ip-privé-to-ip-public**, basé sur des expressions régulières qui modifient l'adresse de l'hôte de 192.168.1.24 en 137.45.6.39 ( *IP de sous réseau en IP public* ).
    - Tagger **tag-ip-public**, basé sur des expressions régulières qui modifie la liste des templates des hôtes dont l'IP commence par 137 pour y mettre le modèle "réseau public".
      - Si l'ordre des Taggers est : Tagger **ip-privé-to-ip-public** puis Tagger **tag-ip-public**, un hôte avec l'IP 192.168.1.24 aura l'IP 137.45.6.39 et le modèle "réseau public".
      - Si l'ordre des Taggers est : Tagger **tag-ip-public** puis Tagger **ip-privé-to-ip-public**, un hôte avec l'IP 192.168.1.24 aura l'IP 137.45.6.39, mais n'aura PAS le modèle "réseau public" ( *car l'adresse de l'hôte a été changée après* ).

La page va se focaliser sur la partie "application des Taggers".

- Pour plus d'informations sur le mécanisme d'imports des éléments, voir la page [Modules de Sources \( imports \)](#) et de [Taggers \( qualification \)](#).

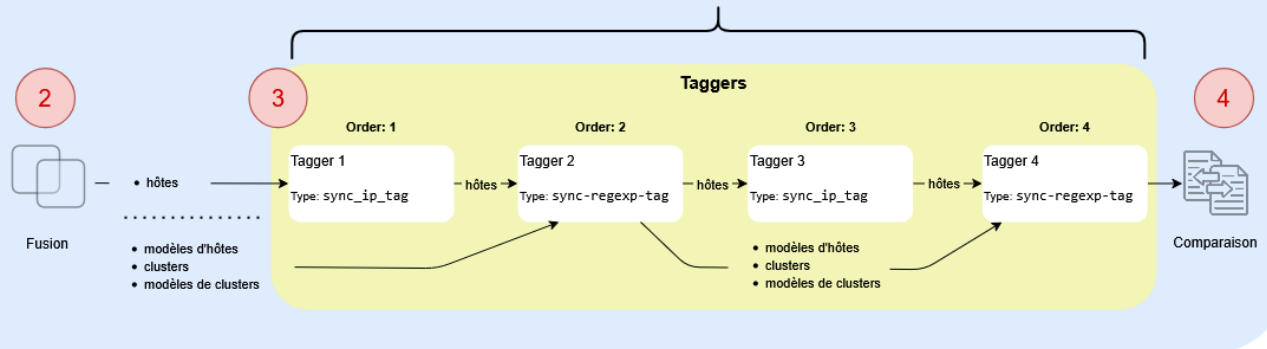


## Détails sur l'exécution des Taggers

# Synchronizer

( Détails sur l'exécution des Taggers )

L'ordre de passage est défini par le paramètre "order".



2

Mélange des sources  
( Vérification et fusion )

3

Application des  
Taggers

4

Comparaison avec  
la Zone de Travail  
et Staging

Dans cet exemple, quatre Taggers sont exécutés les uns après les autres dans l'ordre suivant :

1. Tagger 1, qui est un tagger basé sur les plages IP ;
2. Tagger 2, qui est un tagger basé sur des expressions régulières ;
3. Tagger 3, qui est un tagger basé sur les plages IP ;
4. Tagger 4, qui est un tagger basé sur des expressions régulières ;

( voir les pages [Tagger utilisant un module basé sur les plages IP](#) et [Tagger utilisant un module basé sur des expressions régulières](#) ).

Comme dans l'exemple précédent, seuls les hôtes, modèles d'hôtes, clusters et modèles de clusters passent dans les Taggers.

- Les hôtes passent dans tous les Taggers les uns à la suite des autres ;
- Les clusters, modèles de clusters et modèles de clusters passent uniquement dans Tagger 2 puis dans Tagger 4, car les Taggers de type `sync-ip-tag` n'acceptent que les hôtes ;
- Les autres éléments de Shinken ( *checks, périodes, modulations de résultats, contacts ...* ) ne passent pas dans les Taggers et vont directement à l'étape 4 - Comparaison avec Staging et la Zone de travail ;

## Définir ou configurer un Tagger

Regarder la page [Définition des taggers](#), pour mettre en place un tagger ou en modifier un.