

Module de type sync_ip_tag

Sommaire

- [Concept](#)
- [Définition du module](#)
 - [Exemple de fichier de configuration](#)
 - [Détails des sections composant le fichier de configuration](#)
 - [Identification du module](#)
 - [Définition des hôtes où l'action de modification sera appliquée \(plage d'adresses IP et exclusion \)](#)
 - [Édition des hôtes](#)
- [Déclarer le Module dans le Tagger](#)
- [Affichage des Taggers sur l'interface de Configuration](#)

Concept

Il est possible de définir un Tagger qui utilise une plage d'adresses IP afin d'éditer automatiquement une propriété des hôtes issus de l'import des sources en fonction de la valeur de leur IP ou nom DNS (*propriété "address"*).

L'intêret est d'ajouter le modèle d'hôte DMZ sur les hôtes avec des adresses comprises dans la plage IP 172.20.0.0/26 (voir la page [Tagger basé sur les plages IP](#)).

L'activation d'un Tagger se fait en trois étapes : :

1. Définir un Tagger qui utilise un module de type sync_ip_tag (*ex : /etc/shinken/taggers/ip-tags.cfg*).
2. Configurer les règles du module (*ex : /etc/shinken/modules/ip-tag-dmz.cfg*).
3. Déclarer le Tagger dans le Synchronizer (*/etc/shinken/synchronizers/synchronizer-master.cfg*).

Définition du tagger

La configuration des Taggers doit être placée dans le dossier **/etc/shinken/taggers/**.

Le Tagger basé sur les plages IP livré par Shinken : **/etc/shinken/taggers/ip-tags.cfg**.

Un exemple de fichier de configuration est disponible : **/etc/shinken-user-example/configuration/daemons/synchronizers/taggers/ip-tag/ip-tags-example.cfg**

Exemple de fichier de configuration

```
define tagger {
# Shinken Enterprise. Lines added by import core. Do not remove it, it's used by Shinken Enterprise to
update your objects if you re-import them.
    _SE_UUID          core-tagger-ed4200e85adc11e58e4d080027f08538
    _SE_UUID_HASH     0d3fd680dc3d526ffc3750cfa7ef49d1
# End of Shinken Enterprise part

    tagger_name       ip-tags
    order             1

    modules            ip-tag-dmz

    description       This tagger will tag hosts based on their ip range
}
```


Pour prendre en compte le changement de configuration, il faut redémarrer le Synchronizer :

```
service shinken-synchronizer restart
```

Détails des sections composant le fichier de configuration

Il est possible de définir plusieurs instances de Tagger s dans l'architecture Shinken .

Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Commentaire
tagger_name	Texte	---	---	Valeur obligatoire Nom du Tagger. Le nom doit être unique parmi tous les taggers existants. <u>Caractères interdits :</u> les signes inférieur ou supérieur (< ou >), les guillemets (' ou ").
order	Entier	---	---	Valeur obligatoire L'ordre d'application des Taggers. Les Taggers sont exécutés les uns à la suite des autres dans l'ordre croissant de la propriété. <div style="border: 1px solid red; padding: 5px; margin-top: 10px;"> Si plusieurs Taggers qui s'enchaînent modifient la même propriété, il est possible qu'un Tagger annule les modifications de précédents Taggers. Attention donc à l'ordre d'exécution des Taggers</div>
modules	Texte	---	---	Valeur obligatoire Nom du module utilisé par le Tagger.
description	Texte	---	---	Une description du Tagger affichée dans l'Interface de Configuration

Configuration du module

La configuration des modules doit être placée dans le dossier **/etc/shinken/modules/**

La configuration du module livré par Shinken se trouve dans le fichier **/etc/shinken/modules/ip-tag-dmz.cfg**

Un exemple de fichier de configuration est disponible ici : **/etc/shinken-user-example/configuration/daemons/synchronizers/modules/ip-tag-dmz/ip-tag-dmz-exemple.cfg**

```

=====
# ip-tag-*
=====
# Daemons that can load this module:
# - synchronizer (into a tagger object)
# This module add new templates into hosts objects based on their IP range. For example to set a DMZ
template is an host is in a DMZ ip range.
=====

define module{
# Shinken Enterprise. Lines added by import core. Do not remove it, it's used by Shinken Enterprise to
update your objects if you re-import them.
    _SE_UUID          core-module-07a50e365adc11e5a550080027f08538
    _SE_UUID_HASH     5d15556b0225fcc39cecf7d86d636120
# End of Shinken Enterprise part

    #==== Module identity =====
    # Module name. Must be unique
    module_name       ip-tag-dmz

    # Module type (to load module code). Do not edit.
    module_type       sync_ip_tag

    #==== Ip range definiton, and objects edition =====
    # IP range to detect. Should looks like:   XXX.XXX.XXX.XXX/YY
    ip_range          172.20.0.0/30

    # host_name listed will be skipped by tagger
    # ignore_hosts

    # property: which property to edit on the hosts. Default: use (templates definitions)
    property          use

    # method: how to edit the host "property". Several methods are available:
    # - replace = put the value if not another one is in place
    # - append  = add the value at the END
    # - prepend = add the value at the BEGINING
    # - set     = just the value, erase the previous value set by other tagger or source.
    method          append

    # value: which value to set on the property
    value           DMZ
}

```

Pour prendre en compte le changement de configuration, il faut redémarrer le Synchronizer :

```
service shinken-synchronizer restart
```

Détails des sections composant le fichier de configuration

Identification du module

```

#==== Module identity =====
# Module name. Must be unique
module_name       ip-tag-dmz

# Module type (to load module code). Do not edit.
module_type       sync_ip_tag

```

Il est possible de définir plusieurs instances de module de type `sync_ip_tag` dans l'architecture Shinken .

Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Commentaire
module_name	Texte	---	---	Valeur obligatoire Nom du module. Doit être unique parmi tous les modules.
module_type	Texte	---	<code>sync_ip_tag</code>	Ne pas modifier.



Définition de la plage d'adresses IP et de l'édition des objets

```

===== Ip range definition, and objects edition =====
# IP range to detect. Should looks like:  XXX.XXX.XXX.XXX/YY
ip_range          172.20.0.0/30

# host_name listed will be skipped by tagger
# ignore_hosts

# property: which property to edit on the hosts. Default: use (templates definitions)
property          use

# method: how to edit the host "property". Several methods are available:
# - replace = put the value if not another one is in place
# - append  = add the value at the END
# - prepend = add the value at the BEGINING
# - set     = just the value, erase the previous value set by other tagger or source.
method          append

# value: which value to set on the property
value           DMZ
    
```

Nom	Type	Unité	Défaut	Commentaire
ip_range	Texte	IP/masque	---	Valeur obligatoire Plage d'IP à détecter.
property	Texte	---	---	Valeur obligatoire Nom de la propriété à modifier sur l'hôte.
method	Texte	---	<code>replace</code>	Valeur obligatoire Méthode d'édition : <ul style="list-style-type: none"> • replace : ajoute la valeur si elle est absente. • append : ajoute la valeur à la fin. • prepend : ajoute la valeur au début. • set : remplace la valeur actuelle ou l'ajoute si absente.
value	Texte	---	---	Valeur obligatoire Valeur utilisée par la méthode.



Déclarer le Tagger dans le Synchronizer

L'activation des taggers se fait dans le fichier de configuration du Synchronizer :

- Il faut ajouter le nom du Tagger au paramètre `taggers` dans le fichier de configuration du démon Synchronizer (*/etc/shinken/synchronizers/synchronizer-master.cfg*) .

Exemple : Le Tagger livré par défaut dont le nom est "ip-tags":

```
define synchronizer {
    [...]
    #----- Taggers -----
    taggers          TAGGER_01, ip-tags
    [...]
}
```

Pour prendre en compte le changement de configuration, il faut redémarrer le Synchronizer :

```
service shinken-synchronizer restart
```

Affichage des Taggers sur l'interface de Configuration

Les Taggers apparaissent sur la page Principale de l'Interface de Configuration (*voir la page [Page Principale](#)*).