

# 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 sur un Tagger une action qui va utiliser 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"* ).

- Cette action est portée par un module de type `sync_ip_tag`, qu'il faudra rajouter sur le Tagger .
- L'intérêt est, par exemple, 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 utilisant un module basé sur les plages IP* ).

L'activation de ce module se fait en 3 étapes :

1. Définir le module de type `sync_ip_tag`.
2. L'ajouter au Tagger qui exécutera cette règle.
  - a. Si vous n'avez pas déjà de Tagger défini, il faut le mettre en place ( *voir la page Définition des taggers* ).
3. Redémarrer le Synchroniser pour que cette modification soit prise en compte.

## Définition du module

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

- Un exemple de fichier de configuration est disponible ici : **/etc/shinken-user-example/configuration/daemons/synchronizers/modules/sync\_ip\_tag/MY-MODULE-ip-tag.cfg**
- Copier cet exemple et adapter le au besoin ( *voir plus bas, la description des différents paramètres* ).

```
cp /etc/shinken-user-example/configuration/daemons/synchronizers/modules/sync_ip_tag/MY-MODULE-ip-tag.cfg /etc/shinken/modules/MY-MODULE-ip-tag.cfg
```

Shinken recommande d'intégrer, dans le nom du module, un rappel du type du module. C'est plus simple en gestion ( *ex : MY-MODULE-ip-tag.cfg* ).

- Renommer le module ( *paramètre `module_name`* ).
- Éditer les paramètres pour définir l'action du module :
  - Les hôtes à impacter.
  - La nature de l'action.

## Exemple de fichier de configuration

Cet exemple ajoute

- le modèle d'hôte DMZ, à la fin des modèles d'hôtes utilisés par chaque hôte
- avec des adresses comprises dans la plage IP 172.20.0.0/30.

```

#=====
# 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{
    #===== Module identity =====
    # Module name. Must be unique
    module_name          MY-MODULE-ip-tag

    # 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
}

```

## Détails des sections composant le fichier de configuration

### Identification du module

```

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

# 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	Description
-----	------	-------	--------	-------------

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

### Définition des hôtes où l'action de modification sera appliquée ( plage d'adresses IP et exclusion )

```

===== 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

```

Nom	Type	Unité	Défaut	Description
ip_range	Texte	IP/masque	---	<b>Valeur obligatoire</b> Plage d'IP à détecter.
ignore_hosts	Liste	---	---	Liste de nom d'hôtes ( <i>et pas de leurs adresses</i> ) qui seront ignorés par ce module de Tagger.

### Édition des hôtes

```

# 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	Description
property	Texte	---	---	<b>Valeur obligatoire</b> Nom de la propriété à modifier sur l'hôte. Ex : use pour les modèles de l'hôte.
method	Texte	---	append	<b>Valeur obligatoire</b> Méthode d'édition : <ul style="list-style-type: none"> <li>• <b>replace</b> : ajoute la valeur si elle est absente.</li> <li>• <b>append</b> : ajoute la valeur à la fin.</li> <li>• <b>prepend</b> : ajoute la valeur au début.</li> <li>• <b>set</b> : remplace la valeur actuelle ou l'ajoute si absente.</li> </ul>
value	Texte	---	---	<b>Valeur obligatoire</b> Valeur utilisée lors de l'application de la règle.

## Déclarer le Module dans le Tagger

Pour qu'un Tagger utilise les règles définies dans un module, il faut ajouter le nom du module au paramètre `modules` dans le fichier de configuration du Tagger ( *ex : `/etc/shinken/taggers/MY-TAGGER.cfg`* )

Exemple avec un Tagger appelé MY-TAGGER :

```
define tagger {
    tagger_name      MY-TAGGER
    order            1

    modules          TAGGER-MODULE-01, TAGGER-MODULE-02, MY-MODULE-ip-tag

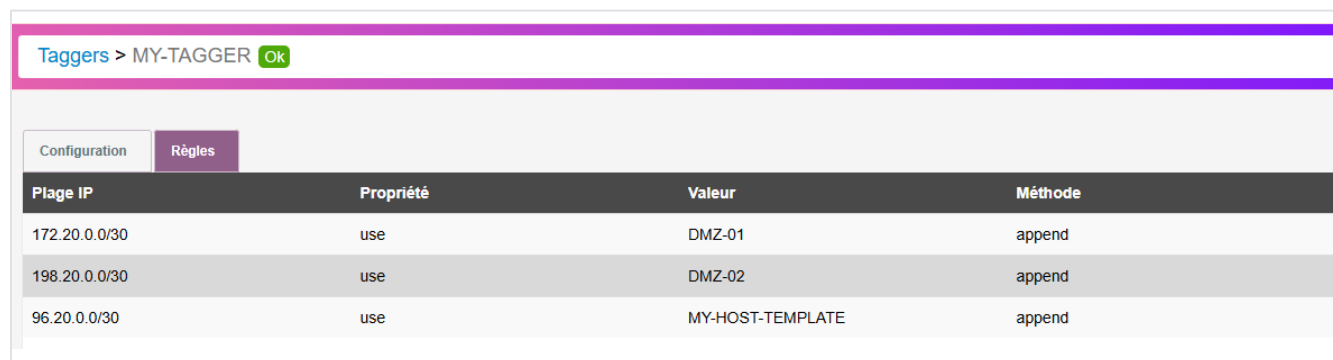
    description
}
```

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 règles apportées par chaque module apparaissent à l'intérieur du Tagger, dans l'onglet **Règles** :



Taggers > MY-TAGGER OK

Plage IP	Propriété	Valeur	Méthode
172.20.0.0/30	use	DMZ-01	append
198.20.0.0/30	use	DMZ-02	append
96.20.0.0/30	use	MY-HOST-TEMPLATE	append