

Edition d'un utilisateur

Sommaire

[Onglet générique](#)
[Données](#)
[Autorisations](#)
 [UI Visualisation \[Actions \]](#)
 [UI Visualisation \[Affichage des zones \]](#)
 [UI Visualisation \[Les favoris \]](#)
 [UI Configuration \[Essais des checks \]](#)
[Configuration directe](#)

Contexte

Le modèle **switch-SNMPv1v2-detailed** mets à disposition les 4 checks suivant :

Le but de ce modèle est de donner un point de vue plus détaillé sur chaque interface du switch, et donner des informations plus précises comme la mémoire RAM disponible et l'état du CPU.

Comment appliquer le modèle à un hôte

En utilisant l'interface de Configuration

Dans l'interface de Configuration,

- créez ou [éditez un Hôte](#),
- et ajoutez le modèle "**switch-SNMPv1v2-detailed**" dans la propriété "**Modèles d'hôte hérités**" à l'aide du menu déroulant.

? Unknown Attachment

Via un collecteur d'import de fichiers au format .cfg

Dans un fichier de configuration,

- créez ou éditez votre définition d'hôte
- en ajoutant, dans la propriété "**use**", la valeur "**switch-SNMPv1v2-detailed**" selon les besoins.

Puis, le contenu du fichier devra alors être importé avec une source d'import de fichier de configuration ([plus d'information sur cette page](#) : [Collecteur de type \(cfg-file-import \) - Import depuis des fichiers au format .cfg](#)).

```
define host {
    host_name    Switch
    use switch-SNMPv1v2-
detailed
}
```

Les donnée du modèle

Configuration de la connexion SNMP

Dans chaque hôte héritant du modèle d'hôte "**switch-SNMPv1v2-detailed**", vous aurez la donnée SNMPCOMMUNITY concernant la communauté SNMP interrogé :

Description	Valeur par défaut	Valeur par défaut à l'installation de shinken	Unités

SNMPCOMMUNITY	<p>Doit correspondre à la Communauté SNMP v1/v2 défini sur votre switch :</p> <ul style="list-style-type: none"> En SNMP v1/v2, la communauté est un équivalent du mot de passe pour se connecter sur les équipements. <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p>Remarque</p> <p>Par défaut, la donnée SNMPCOMMUNITY prend le contenu de la donnée globale \$SNMPCOMMUNITYREAD\$ dont la valeur est modifiable dans le fichier <code>/etc/shinken/resource.d/snmp.cfg</code>.</p> <ul style="list-style-type: none"> voir LES VARIABLES (Remplacement dynamique de contenu - Anciennement les MACROS), pour comprendre leur utilisation. <p>La modification des valeurs par défaut présentes dans le fichier du serveur (<code>/etc/shinken/resource.d/snmp.cfg</code>) nécessite un redémarrage :</p> <ul style="list-style-type: none"> du Synchronizer pour le voir dans l'interface et l'envoi d'une configuration ou d'un redémarrage de l'Arbiter pour que la modification soit prise en compte en supervision. <p>Donc, pour changer la valeur de SNMPCOMMUNITY, vous pouvez :</p> <ul style="list-style-type: none"> soit changer sa valeur, soit modifier le contenu de la donnée globale \$SNMPCOMMUNITYREAD\$ (en prenant en compte que tous les hôtes utilisant cette donnée seront impactés). </div>	\$SNMPCOMMUNITYREAD\$	public	--
SWITCH_TIMEOUT	<p>Cette variable permet au check de s'arrêter après un certain temps si une tâche ne s'est pas terminée.</p> <ul style="list-style-type: none"> Cela permet d'éviter que le programme ne s'exécute indéfiniment et de prévenir des problèmes de performances. L'unité est la seconde. 	60	60	Secondes

Par exemple, voici les données d'une connexion avec la communauté SNMPv1/2 :

? Unknown Attachment

Données spécifiques à certains checks

Les checks suivant *utilisent des données spécifiques qui sont décrite dans leur page respective.*

- "Cpu" (voir la page [FOR_MERGE - 002.9.9 - SEF-10330 - Cpu](#))
- "Memory" (voir la page [FOR_MERGE - 002.9.9 - SEF-10330 - Memory](#))

Donnée SWITCH_INTERFACES pour la duplication de certain checks (mécanisme de Duplicate Foreach)

La donnée **SWITCH_INTERFACES** est utilisé pour dupliquer les 3 checks suivant :

- "Nom d'interface Errors" (voir [FOR_MERGE - 002.9.9 - SEF-10331 - Nom d'interface : Errors \(utilise le mécanisme de duplicateForEach \)](#)),
- "Nom d'interface Status" (voir [FOR_MERGE - 002.9.9 - SEF-10331 - Nom d'interface : Status \(utilise le mécanisme de duplicateForEach \)](#))
- "Nom d'interface Usage" (voir [FOR_MERGE - 002.9.9 - SEF-10331 - Nom d'interface : Usage \(utilise le mécanisme de duplicateForEach \)](#))

Il vous faudra saisir le nom des interfaces qui vont être surveillées :

- Dans l'exemple ci-dessous, le modèle liste **par défaut** les interfaces appelées **port1** et **port2**.

? Unknown Attachment

- Changer cette liste avec les noms qui concernent votre équipement :
 - Par exemple Eth1, Eth2, Eth3, pour avoir les checks surveillantes ces interfaces.

RAJOUTER l'image identique à celle dessus, mais avec Eth1, Eth2, Eth3

Pour cette donnée,

- La **Virgule** sert de séparateur.
- Vous pouvez mettre n'importe quel chaîne de caractère.
 - Si vous voulez surveiller les interfaces réseau non continues, comme les Eth1, Eth2, Eth3 et Eth9, Eth10, il vous suffit d'en faire la liste :

```
Eth1,Eth2,Eth3,Eth9,Eth10
```

 **Astuce**

Si vous voulez générer une liste de 256 ports de la forme Eth0,Eth1, Eth2, ... Eth255, le faire à la main serait très fastidieux!

Nous avons donné la possibilité de générer **AUTOMATIQUEMENT** les nombres :

- La syntaxe **[nombre1-nombre2]** permet de générer plusieurs valeurs
- donc pour avoir notre liste, saisissez **Eth[0-255]**

Syntaxe	Résultats	Commentaire
Eth[5-9]	Eth5,Eth6,Eth7,Eth8,Eth9	Pour une liste d'interfaces continue
Eth[5-9], Eth [60-65]	Eth5,Eth6,Eth7,Eth8,Eth9,Eth60,Eth61,Eth62,Eth63,Eth64, Eth65	Pour faire des sauts entre plusieurs listes d'interfaces continues
Eth[2-3][0-5]	Eth20,Eth21,Eth22,Eth23,Eth24,Eth25,Eth30,Eth31,Eth32, Eth33,Eth34,Eth35	Pour faire un saut régulier entre des listes d'interfaces

La génération de la liste des valeurs souhaitées se fera à la sauvegarde de l'hôte et sera ensuite sous forme de la liste générée (*Eth20, Eth21 ...*).