Lab supervision Zabbix

BENNAT Walid

LAB 5 : Les modèles (template)

Définition :

Un **template (modèle)** dans **Zabbix** est un **ensemble préconfiguré d'éléments de supervision** — tels que des **items**, **déclencheurs (triggers)**, **graphes**, **règles de découverte (LLD)** et **macros** — qui peuvent être **appliqués à plusieurs hôtes** pour faciliter et uniformiser la surveillance.

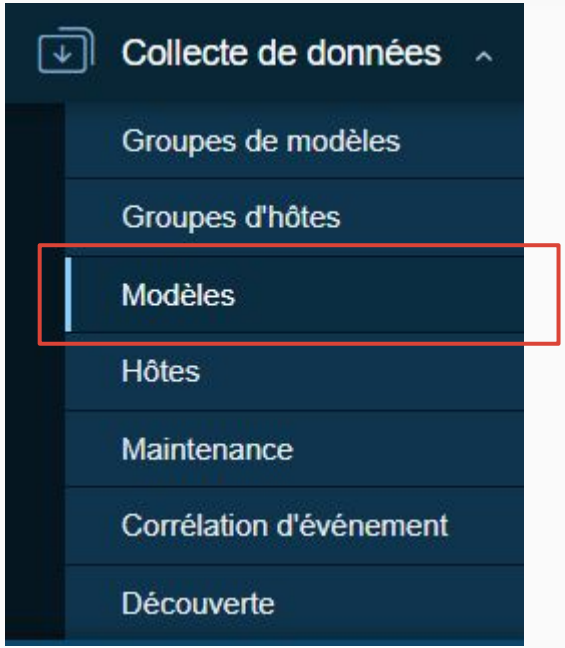
En d'autres termes :

Le template sert de **modèle réutilisable** : au lieu de configurer manuellement la supervision sur chaque serveur, routeur ou application, on crée un **template une seule fois**, puis on l'associe à autant d'hôtes que nécessaire.

Exemple :

- Le template **"Template OS Linux"** contient des éléments pour surveiller le CPU, la mémoire, le disque, etc.
- Si vous avez 50 serveurs Linux, il suffit de leur lier ce template pour qu'ils soient automatiquement supervisés avec les mêmes règles.

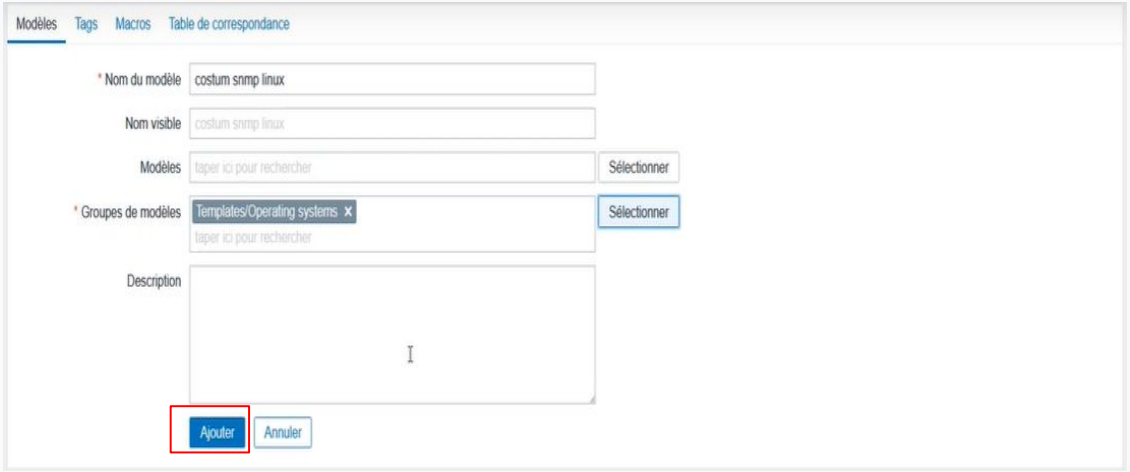
LAB 5 : Les modèles (template)

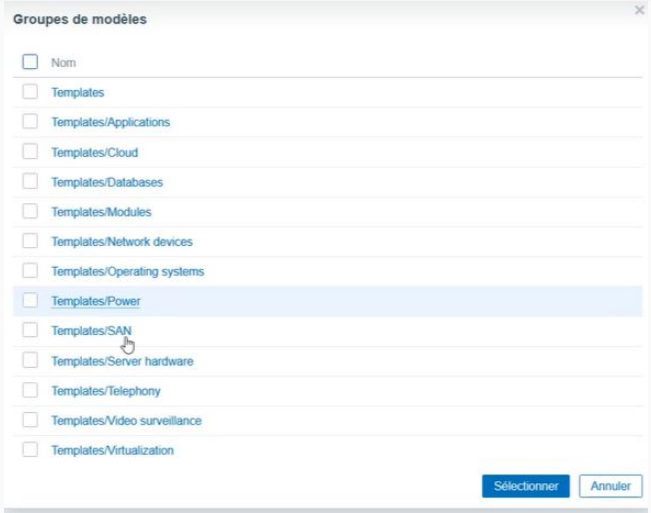


Nom ▲	Hôtes	Éléments	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web	Fabricant	Version	Modèles liés	Lié aux modèles
Azure VM Scale Set by HTTP	Hôtes	Éléments 54	Déclencheurs 6	Graphiques 12	Tableaux de bord 1	Découverte	Web	Zabbix	7.0-1		
Brocade FC by SNMP	Hôtes	Éléments 17	Déclencheurs 12	Graphiques 2	Tableaux de bord 1	Découverte 4	Web	Zabbix	7.0-2		
Brocade_Foundry Nonstackable by SNMP	Hôtes	Éléments 16	Déclencheurs 10	Graphiques 2	Tableaux de bord 1	Découverte 5	Web	Zabbix	7.0-2		
Brocade_Foundry Stackable by SNMP	Hôtes	Éléments 15	Déclencheurs 9	Graphiques 2	Tableaux de bord 1	Découverte 6	Web	Zabbix	7.0-2		
Ceph by Zabbix agent 2	Hôtes	Éléments 51	Déclencheurs 4	Graphiques 5	Tableaux de bord	Découverte 2	Web	Zabbix	7.0-2		
Chassis by IPMI	Hôtes	Éléments 1	Déclencheurs	Graphiques	Tableaux de bord	Découverte 2	Web	Zabbix	7.0-1		
Check Point Next Generation Firewall by SNMP	Hôtes	Éléments 42	Déclencheurs 10	Graphiques 7	Tableaux de bord 2	Découverte 10	Web	Zabbix	7.0-3		
Ciena 3906 by SNMP	Hôtes	Éléments 23	Déclencheurs 9	Graphiques 2	Tableaux de bord 1	Découverte 4	Web	Zabbix	7.0-0		
Cisco ASAv by SNMP	Hôtes	Éléments 6	Déclencheurs 2	Graphiques	Tableaux de bord 1	Découverte 5	Web	Zabbix	7.0-2		
Cisco Catalyst 3750V2-24FS by SNMP	Hôtes	Éléments 15	Déclencheurs 8	Graphiques	Tableaux de bord 1	Découverte 8	Web	Zabbix	7.0-2		

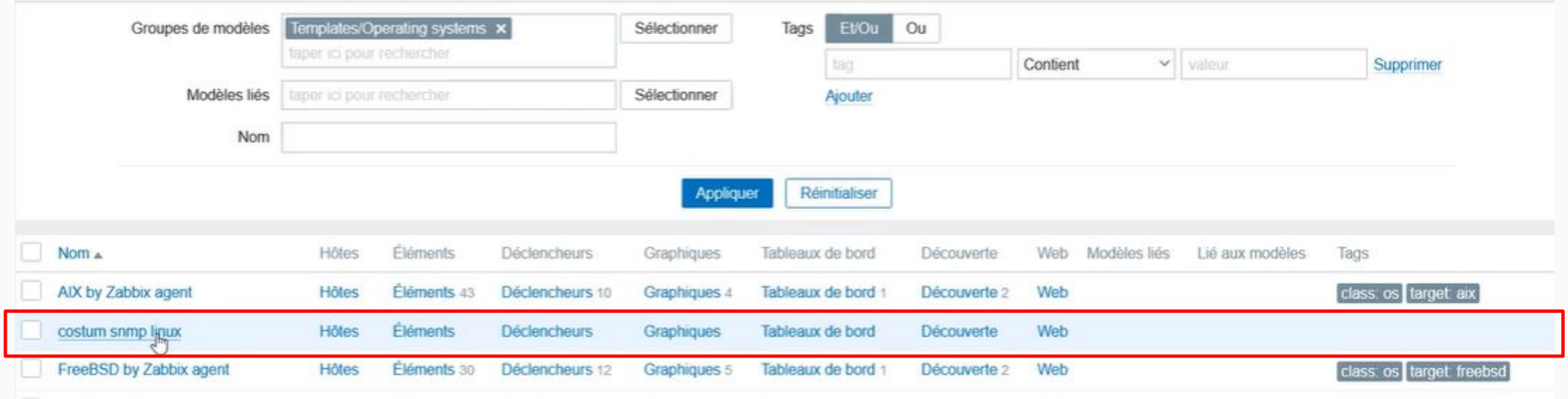
LAB 5 : Les modèles (template)

Il est nécessaire d'attribuer un nom au modèle ainsi qu'au groupe, ces deux champs étant obligatoires.





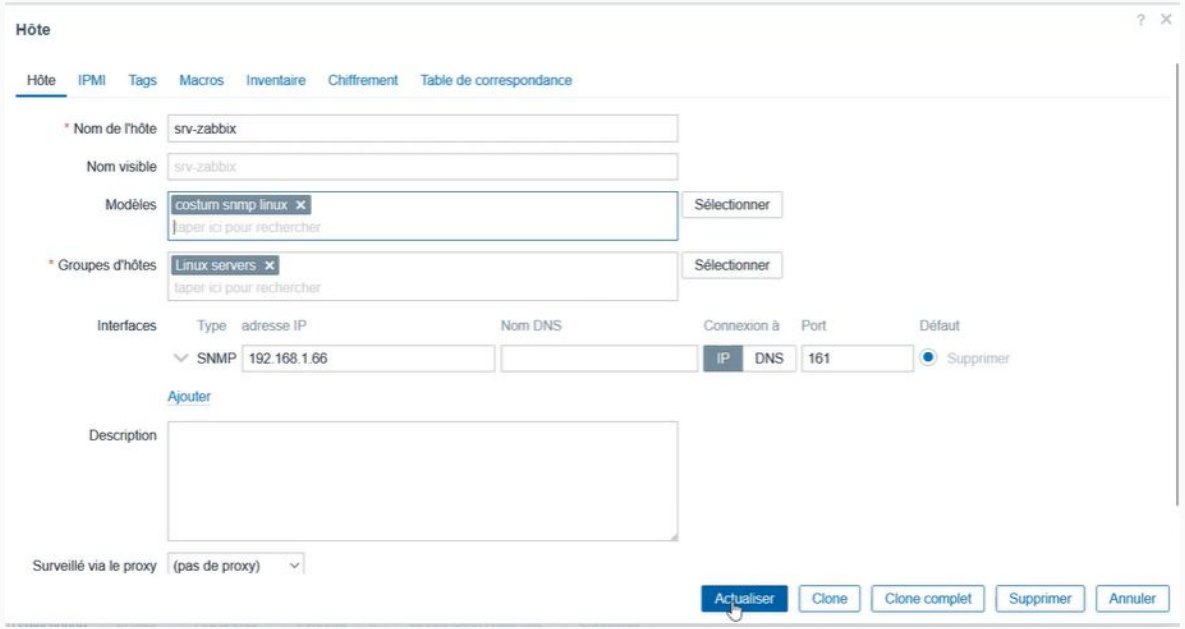
LAB 5 : Les modèles (template)



On constate que notre modèle a bien été créé avec succès.

LAB 5 : Les modèles (template)

1. On doit lier le modèle à notre hôte.



2. Pourquoi ?

Imagine que tu as **20 serveurs Linux**.
Tu ne vas pas configurer manuellement chaque indicateur (CPU, disque, etc.) sur les 20 hôtes
Tu appliques simplement le modèle **“Template OS Linux”** à chaque machine.
Résultat : tous les serveurs sont surveillés de la même façon, en quelques clics.

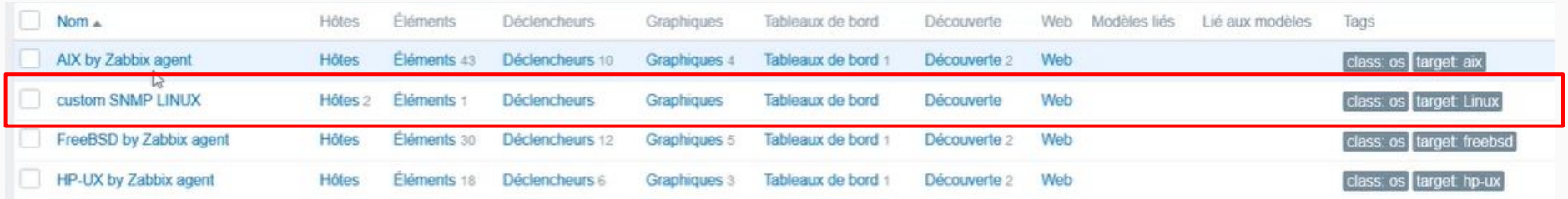
LAB 5 : Les modèles (template)

Nom du modèle (template)	Type de modèle	Ce qu'il surveille	Exemple d'utilisation
Template OS Linux by Zabbix agent	Système d'exploitation	CPU, mémoire, disque, processus, charge système, uptime.	Surveiller les serveurs Linux (Ubuntu, Debian, CentOS, etc.).
Template OS Windows by Zabbix agent	Système d'exploitation	Utilisation CPU, RAM, disque, services Windows, journaux d'événements.	Surveiller des serveurs ou postes Windows.
Template App MySQL by Zabbix agent	Application (Base de données)	État du service MySQL, connexions, requêtes par seconde, cache, latence.	Supervision d'un serveur MySQL ou MariaDB.
Template App Apache by Zabbix agent	Application (Serveur web)	Requêtes traitées, trafic HTTP, temps de réponse, erreurs.	Surveiller un serveur web Apache.
Template Net Cisco IOS by SNMP	Réseau	Interfaces réseau, trafic entrant/sortant, erreurs, disponibilité SNMP.	Surveiller un routeur ou un switch Cisco.
Template Custom (personnalisé)	Personnalisé	Élément(s) spécifique(s) à ton environnement : scripts, logs, métriques internes.	Supervision d'une application ou d'un service interne.

LAB 5 : Les modèles (template)

Configuration des modèles

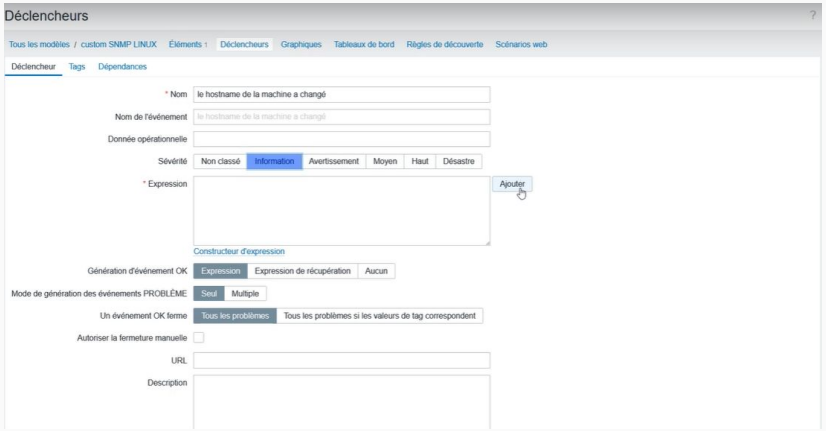
On a déjà créé notre modèle maintenant c'est les moment des configuration; du coup on va commencer par la création des declencheur.
On constate que notre modèle est vierge : aucune configuration n'a encore été effectuée.



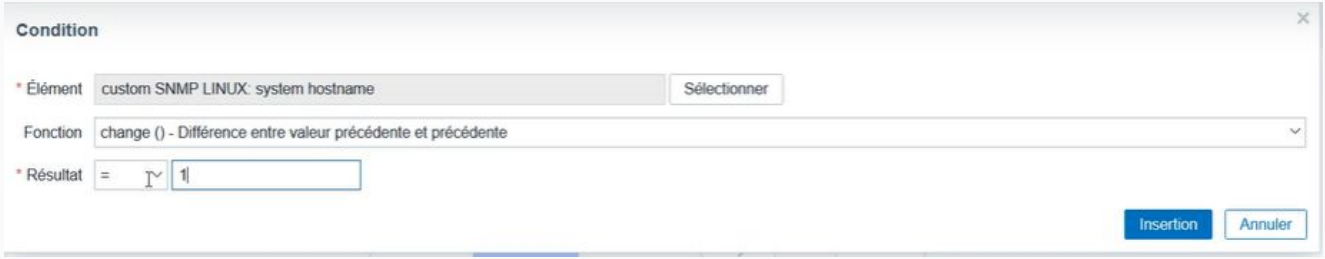
Nom ▲	Hôtes	Éléments	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web	Modèles liés	Lié aux modèles	Tags
AIX by Zabbix agent	Hôtes	Éléments 43	Déclencheurs 10	Graphiques 4	Tableaux de bord 1	Découverte 2	Web			class: os target: aix
custom SNMP LINUX	Hôtes 2	Éléments 1	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web			class: os target: Linux
FreeBSD by Zabbix agent	Hôtes	Éléments 30	Déclencheurs 12	Graphiques 5	Tableaux de bord 1	Découverte 2	Web			class: os target: freebsd
HP-UX by Zabbix agent	Hôtes	Éléments 18	Déclencheurs 6	Graphiques 3	Tableaux de bord 1	Découverte 2	Web			class: os target: hp-ux

LAB 5 : Les modèles (template)

Configuration des modèles

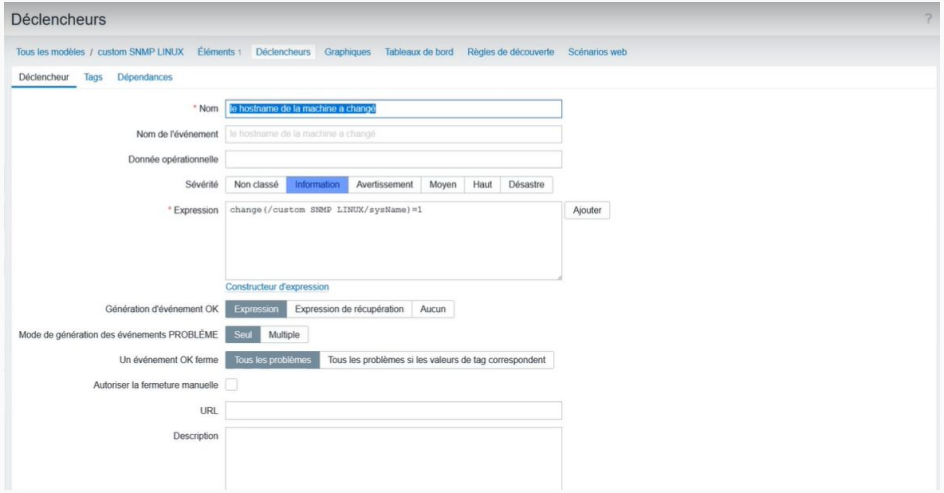


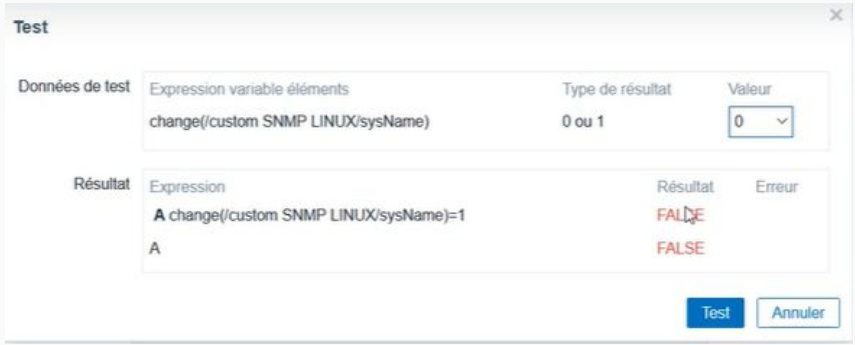
Utilise la fonction change( ) pour vérifier si la valeur de l'élément SNMP sysName est différente de la précédente. Si change( )=1, cela signifie que le hostname a changé et le déclencheur passe en état PROBLÈME.



LAB 5 : Les modèles (template)

Configuration des modèles





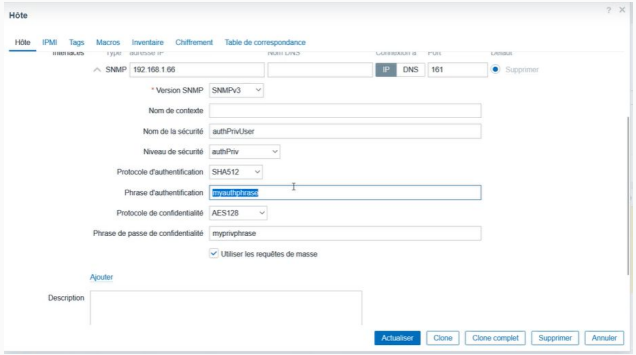
LAB 5 : Les modèles (template)

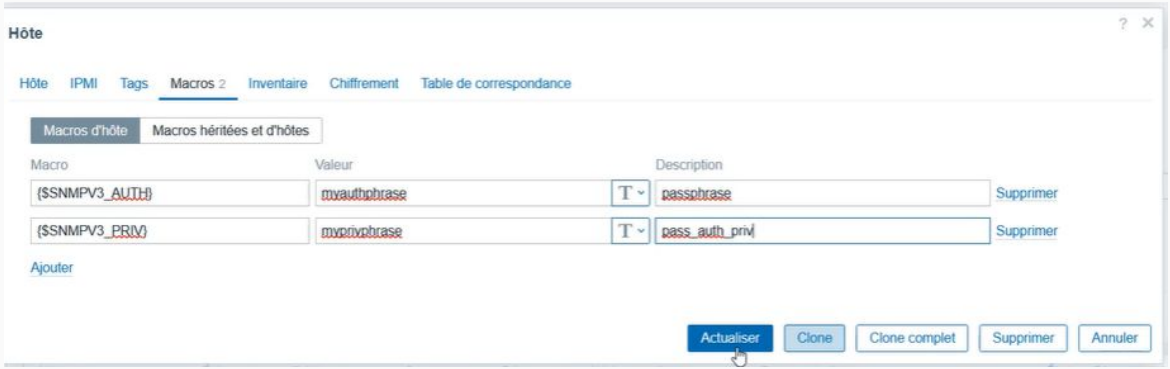
Les Macros

Les **macros dans Zabbix** sont des **variables dynamiques** qui te permettent d'éviter de répéter des valeurs fixes (comme des chemins, des adresses IP, des seuils, etc.).
Mais il existe **plusieurs types de macros**, et c'est là que la **différence** est importante

Elle commence toujours par un **$** ou **{}**

Exemple : pour le protocole SNMPv3, si plusieurs hôtes utilisent les mêmes identifiants, il est préférable de créer des macros afin d'éviter de ressaisir les mots de passe à chaque fois et de limiter les risques d'erreurs.

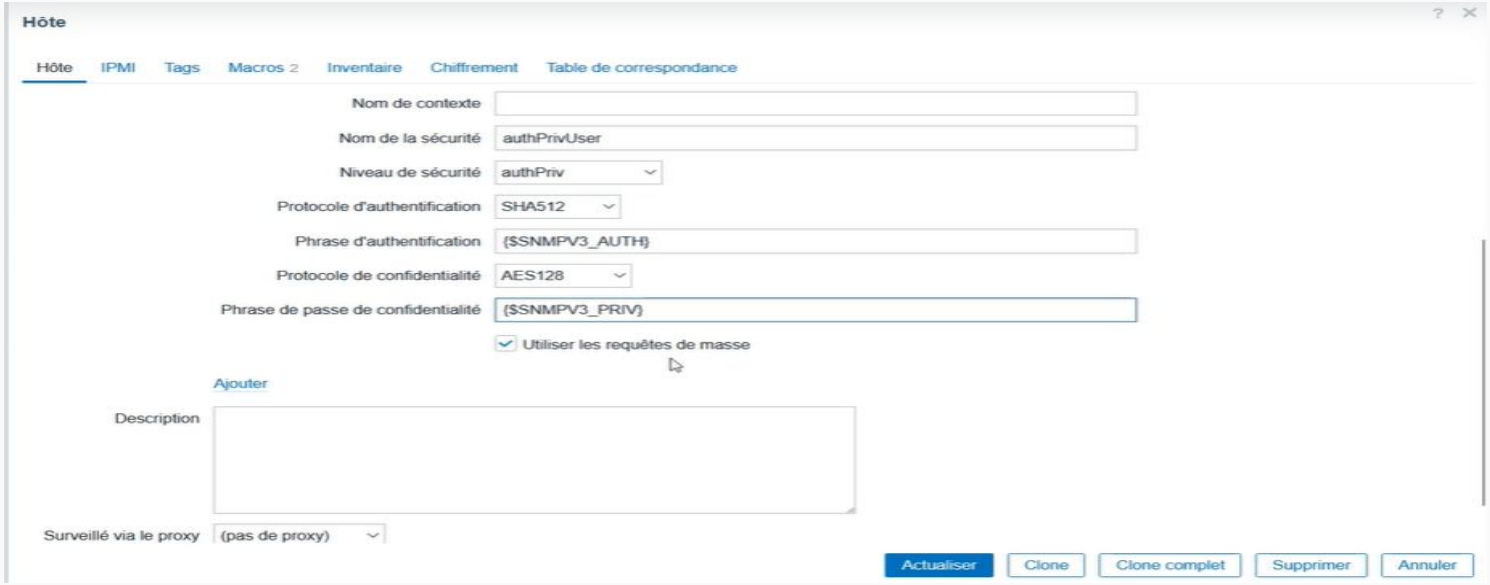




LAB 5 : Les modèles (template)

Les Macros

Résulat



LAB 5 : Les modèles (template)

Les Macros

Résulat

Nom de la macro	Valeur d'exemple	Utilisation	Objectif
{$CPU_WARN}	80	{Template_OS_Linux:system.cpu.util.last()}>={$CPU_WARN}	Déclenche une alerte si le CPU dépasse 80 %.
{$MEMORY_WARN}	90	{Template_OS_Linux:vm.memory.util.last()}>={$MEMORY_WARN}	Alerte si la mémoire utilisée dépasse 90 %.
{$DISK_WARN}	10	{Template_OS_Linux:vfs.fs.size[/,pfree].last()}<{$DISK_WARN}	Alerte si le disque a moins de 10 % d'espace libre.
{$PING_TIMEOUT}	5	{Template_ICMP_Ping:icmppingsec.last()}>={$PING_TIMEOUT}	Alerte si le ping met plus de 5 secondes à répondre.
{$TEMP_MAX}	70	{sensor.temp.last()}>={$TEMP_MAX}	Alerte si la température dépasse 70 °C.

LAB 5 : Les modèles (template)

Définition du LLD

LLD (Low-Level Discovery) signifie **“découverte automatique de bas niveau”**.

C’est un mécanisme qui permet à **Zabbix de détecter automatiquement** les éléments d’un système ou d’un équipement à surveiller,
et de **créer dynamiquement** les éléments de supervision correspondants (items, triggers, graphiques…).

Concrètement, cela veut dire que **Zabbix peut générer automatiquement** :

- des **items** (mesures),
- des **triggers** (alertes),
- des **graphiques**,
- des **applications** (groupes d’items),
- voire des **tableaux de bord** pour des éléments similaires.

LAB 5 : Les modèles (template)

Définition du LLD

Dans notre cas, nous allons prendre un exemple où nous surveillerons l'état des interfaces d'un équipement tel qu'un switch, un routeur ou un pare-feu.

our réaliser cette supervision, nous allons nous appuyer sur le protocole **SNMP** largement utilisé pour la gestion et la surveillance des équipements réseau.

Afin de **contrôler l'état opérationnel d'une interface**, nous utiliserons l'**OID `ifOperStatus`** de la **MIB-II**, dont le rôle est d'indiquer si l'interface est **active (up)**, **inactive (down)** ou dans un autre état (par exemple, **en test**).

Chaque interface est identifiée par un index (`ifIndex`), ce qui permet de connaître précisément l'état de chacune d'elles à l'aide de la requête `ifOperStatus.<index>`.

LAB 5 : Les modèles (template)

Définition du LLD

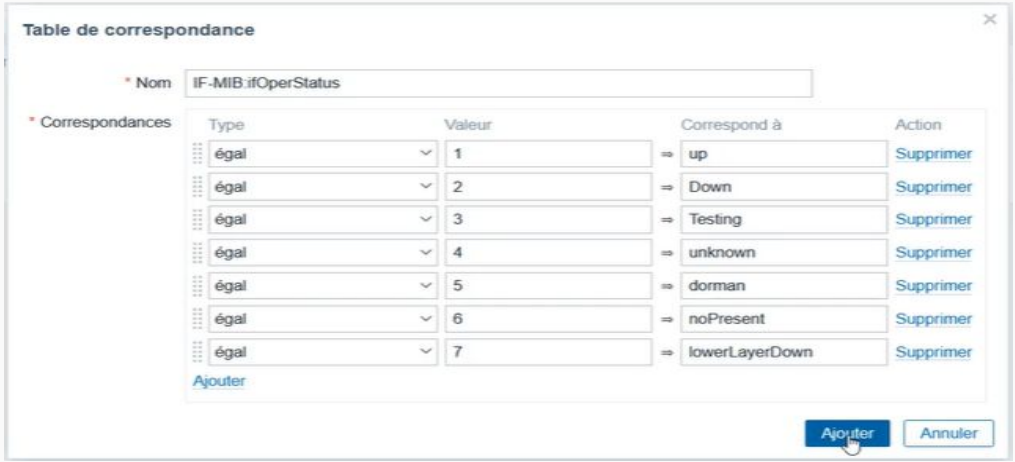
Valeur	Nom de l'état	Signification / Description
1	up(1)	L'interface est opérationnelle et fonctionne correctement.
2	down(2)	L'interface est hors service (désactivée ou en panne).
3	testing(3)	L'interface est actuellement en phase de test.
4	unknown(4)	L'état de l'interface est inconnu.
5	dormant(5)	L'interface est inactive, mais en attente d'un événement (ex. : connexion PPP).
6	notPresent(6)	Aucun matériel associé à cette interface n'a été détecté.
7	lowerLayerDown(7)	Une couche inférieure sur laquelle cette interface dépend est hors service.

LAB 5 : Les modèles (template)

Définition du LLD

Nous allons maintenant mettre en pratique le même exemple directement sur Zabbix.

Étape 1: création table de correspondance



LAB 5 : Les modèles (template)

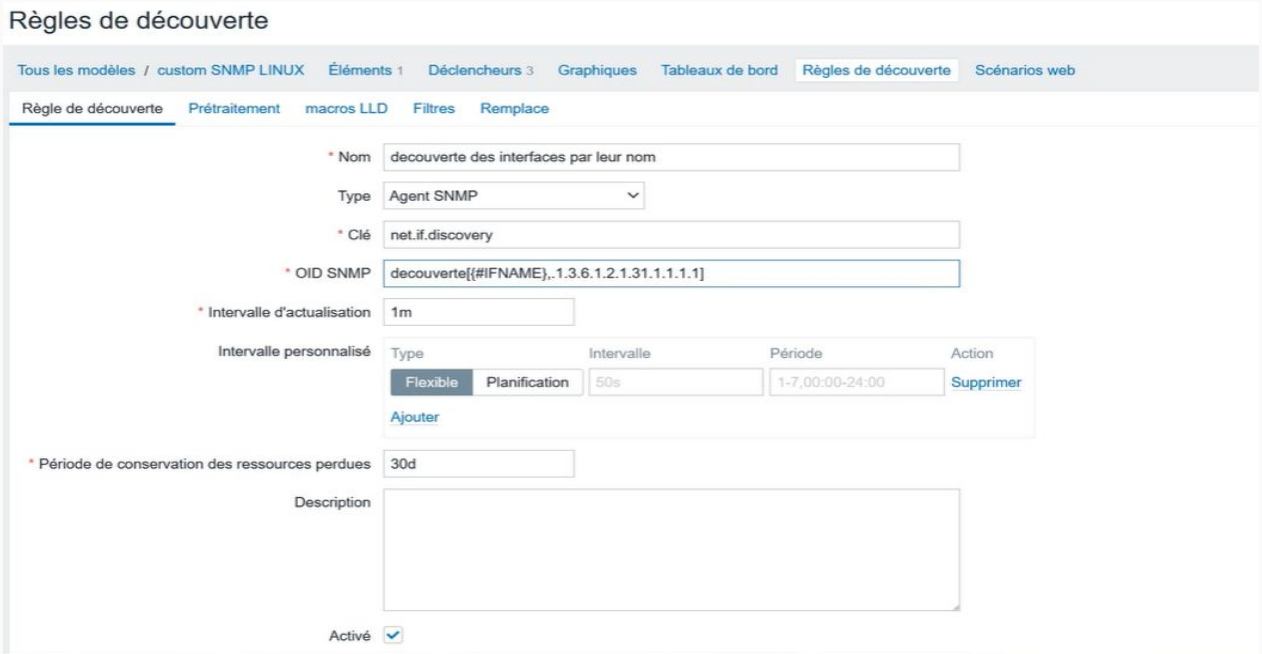
Définition du LLD

Étape 2 création règle de découvertes: Avant de pouvoir récupérer **l'état (ifOperStatus)** d'une interface, il faut d'abord **identifier le nom (ifname) ou l'index (ifIndex)** des interfaces réseau de l'équipement surveillé.
Remarque: faut installé les MIBs Pour traduire les OIDs en noms lisibles

sudo apt update
sudo apt install snmp snmp-mibs-downloader -y

LAB 5 : Les modèles (template)

Définition du LLD



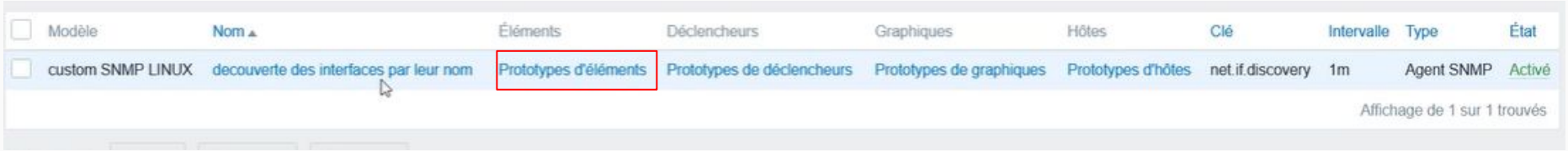
LAB 5 : Les modèles (template)

Définition du prototype

Définition : qu'est-ce qu'un prototype dans Zabbix ?

Dans **Zabbix**, un **prototype** est un **modèle d'élément**, de déclencheur, de graphique ou de découverte** qui sert à **générer automatiquement** plusieurs objets similaires **à partir d'une règle de découverte (LLD – Low Level Discovery)**.

En clair :
Les **prototypes** sont utilisés pour **créer dynamiquement** des éléments de supervision **sans les ajouter manuellement un par un**.



LAB 5 : Les modèles (template)

Définition du prototype