

Lab supervision Zabbix



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)



Collecte de données ^

Groupes de modèles

Groupes d'hôtes

Modèles

Hôtes

Maintenance

Corrélation d'événement

Découverte

<input type="checkbox"/> Nom ▲	Hôtes	Éléments	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web	Fabricant	Version	Modèles liés	Lié aux modèles
<input type="checkbox"/> Azure VM Scale Set by HTTP	Hôtes	Éléments 54	Déclencheurs 6	Graphiques 12	Tableaux de bord 1	Découverte	Web	Zabbix	7.0-1		
<input type="checkbox"/> Brocade FC by SNMP	Hôtes	Éléments 17	Déclencheurs 12	Graphiques 2	Tableaux de bord 1	Découverte 4	Web	Zabbix	7.0-2		
<input type="checkbox"/> Brocade_Foundry Nonstackable by SNMP	Hôtes	Éléments 16	Déclencheurs 10	Graphiques 2	Tableaux de bord 1	Découverte 5	Web	Zabbix	7.0-2		
<input type="checkbox"/> Brocade_Foundry Stackable by SNMP	Hôtes	Éléments 15	Déclencheurs 9	Graphiques 2	Tableaux de bord 1	Découverte 6	Web	Zabbix	7.0-2		
<input type="checkbox"/> Ceph by Zabbix agent 2	Hôtes	Éléments 51	Déclencheurs 4	Graphiques 5	Tableaux de bord	Découverte 2	Web	Zabbix	7.0-2		
<input type="checkbox"/> Chassis by IPMI	Hôtes	Éléments 1	Déclencheurs	Graphiques	Tableaux de bord	Découverte 2	Web	Zabbix	7.0-1		
<input type="checkbox"/> 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		
<input type="checkbox"/> Ciena 3906 by SNMP	Hôtes	Éléments 23	Déclencheurs 9	Graphiques 2	Tableaux de bord 1	Découverte 4	Web	Zabbix	7.0-0		
<input type="checkbox"/> Cisco ASA by SNMP	Hôtes	Éléments 6	Déclencheurs 2	Graphiques	Tableaux de bord 1	Découverte 5	Web	Zabbix	7.0-2		
<input type="checkbox"/> 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.

Modèles Tags Macros Table de correspondance

* Nom du modèle

Nom visible

Modèles Sélectionner

* Groupes de modèles Sélectionner

Description

Groupes de modèles

- ☐ Nom
- ☐ Templates
- ☐ Templates/Applications
- ☐ Templates/Cloud
- ☐ Templates/Databases
- ☐ Templates/Modules
- ☐ Templates/Network devices
- ☐ Templates/Operating systems
- ☒ Templates/Power
- ☐ Templates/SAN
- ☐ Templates/Server hardware
- ☐ Templates/Telephony
- ☐ Templates/Video surveillance
- ☐ Templates/Virtualization

LAB 5 : Les modèles (template)

Groupes de modèles Sélectionner

Modèles liés Sélectionner

Nom

Tags Ou Contient Supprimer

[Ajouter](#)

<input type="checkbox"/> Nom ▲	Hôtes	Éléments	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web	Modèles liés	Lié aux modèles	Tags
<input type="checkbox"/> AIX by Zabbix agent	Hôtes	Éléments 43	Déclencheurs 10	Graphiques 4	Tableaux de bord 1	Découverte 2	Web			class: os target: aix
<input type="checkbox"/> <u>costum snmp linux</u>	Hôtes	Éléments	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web			
<input type="checkbox"/> FreeBSD by Zabbix agent	Hôtes	Éléments 30	Déclencheurs 12	Graphiques 5	Tableaux de bord 1	Découverte 2	Web			class: os target: freebsd

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.

The screenshot shows the Zabbix 'Hôte' (Host) configuration page. The 'Hôte' tab is selected, with other tabs like 'IPMI', 'Tags', 'Macros', 'Inventaire', 'Chiffrement', and 'Table de correspondance' visible. The form contains the following fields and options:

- * Nom de l'hôte:** A text input field containing 'srv-zabbix'.
- Nom visible:** A text input field containing 'srv-zabbix'.
- Modèles:** A dropdown menu showing 'costum snmp linux' with a close button. Below it is a search bar with the placeholder 'taper ici pour rechercher'. A 'Sélectionner' button is to the right.
- * Groupes d'hôtes:** A dropdown menu showing 'Linux servers' with a close button. Below it is a search bar with the placeholder 'taper ici pour rechercher'. A 'Sélectionner' button is to the right.
- Interfaces:** A table with columns: Type, adresse IP, Nom DNS, Connexion à, Port, and Défaut.

Type	adresse IP	Nom DNS	Connexion à	Port	Défaut
SNMP	192.168.1.66		IP	DNS	161

A blue 'Ajouter' link is below the table.
- Description:** A large text area for additional information.
- Surveillé via le proxy:** A dropdown menu currently set to '(pas de proxy)'.

At the bottom right, there are five buttons: 'Actualiser' (highlighted with a mouse cursor), 'Clone', 'Clone complet', 'Supprimer', and 'Annuler'.

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 le moment de la configuration; du coup on va commencer par la création des déclencheurs.

On constate que notre modèle est vierge : aucune configuration n'a encore été effectuée.

<input type="checkbox"/> Nom ▲	Hôtes	Éléments	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web	Modèles liés	Lié aux modèles	Tags
<input type="checkbox"/> AIX by Zabbix agent	Hôtes	Éléments 43	Déclencheurs 10	Graphiques 4	Tableaux de bord 1	Découverte 2	Web			class: os target: aix
<input type="checkbox"/> custom SNMP LINUX	Hôtes 2	Éléments 1	Déclencheurs	Graphiques	Tableaux de bord	Découverte	Web			class: os target: Linux
<input type="checkbox"/> FreeBSD by Zabbix agent	Hôtes	Éléments 30	Déclencheurs 12	Graphiques 5	Tableaux de bord 1	Découverte 2	Web			class: os target: freebsd
<input type="checkbox"/> 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

The screenshot shows the 'Déclencheurs' configuration page. It includes a breadcrumb trail: 'Tous les modèles / custom SNMP LINUX / Éléments / Déclencheurs'. The page has tabs for 'Déclencheur', 'Tags', and 'Dépendances'. The 'Déclencheur' tab is active. The configuration form includes the following fields and options:

- * Nom:** le hostname de la machine a changé
- Nom de l'événement:** le hostname de la machine a changé
- Donnée opérationnelle:** (empty text box)
- Sévérité:** Non classé, Information (selected), Avertissement, Moyen, Haut, Désastre
- * Expression:** (empty text box with an 'Ajouter' button to its right)
- Constructeur d'expression:** (empty text box)
- Génération d'événement OK:** Expression (selected), Expression de récupération, Aucun
- Mode de génération des événements PROBLÈME:** Seul (selected), Multiple
- Un événement OK ferme:** tous les problèmes (selected), Tous les problèmes si les valeurs de tag correspondent
- Autoriser la fermeture manuelle:** ☐
- URL:** (empty text box)
- Description:** (empty text box)

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.

The 'Condition' dialog box is shown, used for defining the trigger condition. It contains the following fields:

- * Élément:** custom SNMP LINUX: system hostname (with a 'Sélectionner' button)
- Fonction:** change () - Différence entre valeur précédente et précédente (dropdown menu)
- * Résultat:** = 1 (with a dropdown menu showing '1' selected)

Buttons at the bottom right: 'Insertion' and 'Annuler'.

LAB 5 : Les modèles (template)

Configuration des modèles

Déclencheurs

Tous les modèles / custom SNMP LINUX / Éléments 1 / Déclencheurs / Graphiques / Tableaux de bord / Règles de découverte / Scénarios web

Déclencheur Tags Dépendances

* Nom

Nom de l'événement

Donnée opérationnelle

Sévérité ☐ Non classé ☒ Information ☐ Avertissement ☐ Moyen ☐ Haut ☐ Désastre

* Expression

Constructeur d'expression

Génération d'événement OK ☐ Expression ☐ Expression de récupération ☐ Aucun

Mode de génération des événements PROBLÈME ☐ Seul ☐ Multiple

Un événement OK forme ☐ Tous les problèmes ☐ Tous les problèmes si les valeurs de tag correspondent

Autoriser la fermeture manuelle ☐

URL

Description

Test

Données de test

Expression variable éléments	Type de résultat	Valeur
change(/custom SNMP LINUX/sysName)	0 ou 1	<input type="text" value="0"/>

Résultat

Expression	Résultat	Erreur
A change(/custom SNMP LINUX/sysName)=1	FALSE	
A	FALSE	

<input type="checkbox"/> Sévérité	Nom ▲	Donnée opérationnelle	Expression	État	Tags
<input type="checkbox"/> Information	le hostname de la machine a changé		change(/custom SNMP LINUX/sysName)=1	Activé	

Affichage de 1 sur 1 trouvés

0 sélectionné

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.

Hôte

Macros

SNMP 192.168.1.66

Version SNMP: SNMPv3

Nom de communauté:

Nom de la sécurité: authPrivUser

Niveau de sécurité: authPriv

Protocole d'authentification: SHA512

Phrase d'authentification:

Protocole de confidentialité: AES128

Phrase de passe de confidentialité:

☒ Utiliser les requêtes de masse

Appliquer

Description:

Actualiser Clone Clone complet Supprimer Annuler

Hôte

Macros 2

Macros d'hôte Macros héritées et d'hôtes

Macro	Valeur	Description	
(\$SNMPV3_AUTH)	myauthphrase	passphrase	Supprimer
(\$SNMPV3_PRIV)	myprivphrase	pass_auth_priv	Supprimer

Ajouter

Actualiser Clone Clone complet Supprimer Annuler

LAB 5 : Les modèles (template)

Les Macros

Résultat

Hôte

Hôte IPMI Tags Macros 2 Inventaire Chiffrement Table de correspondance

Nom de contexte

Nom de la sécurité authPrivUser

Niveau de sécurité authPriv

Protocole d'authentification SHA512

Phrase d'authentification (\$SNMPV3_AUTH)

Protocole de confidentialité AES128

Phrase de passe de confidentialité (\$SNMPV3_PRIV)

☒ Utiliser les requêtes de masse

Ajouter

Description

Surveillé via le proxy (pas de proxy)

Actualiser Clone Clone complet Supprimer Annuler

LAB 5 : Les modèles (template)

Les Macros

Résultat

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.

Pour 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

Table de correspondance

* Nom

* Correspondances

Type	Valeur	Correspond à	Action
égal	1	up	Supprimer
égal	2	Down	Supprimer
égal	3	Testing	Supprimer
égal	4	unknown	Supprimer
égal	5	dorman	Supprimer
égal	6	noPresent	Supprimer
égal	7	lowerLayerDown	Supprimer

[Ajouter](#)

[Ajouter](#) [Annuler](#)

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

Règles de découverte

Tous les modèles / custom SNMP LINUX Éléments 1 Déclencheurs 3 Graphiques Tableaux de bord Règles de découverte Scénarios web

Règle de découverte Prétraitement macros LLD Filtres Remplace

* Nom

Type

* Clé

* OID SNMP

* Intervalle d'actualisation

Intervalle personnalisé

Type	Intervalle	Période	Action
<input checked="" type="checkbox"/> Flexible	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>	Supprimer
<input type="checkbox"/> Planification			

[Ajouter](#)

* Période de conservation des ressources perdues

Description

Activé ☒

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**.

<input type="checkbox"/>	Modèle	Nom ▲	Éléments	Déclencheurs	Graphiques	Hôtes	Clé	Intervalle	Type	État
<input type="checkbox"/>	custom SNMP LINUX	decouverte des interfaces par leur nom	Prototypes d'éléments	Prototypes de déclencheurs	Prototypes de graphiques	Prototypes d'hôtes	net.if.discovery	1m	Agent SNMP	Activé

Affichage de 1 sur 1 trouvés

LAB 5 : Les modèles (template)

Définition du prototype

☺

* Nom

Type

* Clé

Type d'information

* OID SNMP

Unités

* Intervalle d'actualisation

Intervalle personnalisé

Type	Intervalle	Période	Action
<input checked="" type="radio"/> Flexible	<input type="radio"/> Planification	<input type="text" value="50s"/>	<input type="text" value="1-7,00 00-24.00"/>
			<input type="button" value="Supprimer"/>
Ajouter			

* Période de stockage de l'historique

* Période de stockage des tendances

Table de correspondance

Description