

# Supervision des hôtes

## Sommaire

### Introduction

- Quand sont lancés les checks des hôtes?
  - Etats possibles d'un hôte
  - Détermination de l'état de l'hôte
  - Changement d'état d'un hôte
  - Date d'expiration du statut
- Checks et dépendances  
Parallélisation des Checks

## Introduction

### Quand sont lancés les checks des hôtes?

Les hôtes sont vérifiés par les démons Shinken Enterprise :

- À intervalles réguliers, tels que définis dans la définition de l'hôte .
- À la demande quand il y a un changement d'état du check associé à l'hôte .
- À la demande selon la logique de dépendance de l'hôte.

## Partie actif

### Staging > Hôte

Hôte > Validé Customer 1

Général \*  
Données [ 0 ]  
Droits de l'utilisateur  
Supervision  
Checks [ 0 ]  
Notifications  
Expert

Propriété	Valeur	Venant des modèles
Période de maintenance planifiée	-- Par défaut [Aucun] --	

### Vérification du statut de l'élément (ACTIF et PASSIF peuvent être combiné)

**Actif** (Les commandes de vérifications sont ordonnancées et lancées par Shinken)

Actif activé	<input type="checkbox"/> Vrai <input type="checkbox"/> Faux <input checked="" type="checkbox"/> Par défaut [Vrai]
Vivant (Commande de vérification)	-- Par défaut [ check-host-alive (ping) ] --
Tag de Poller	-- Par défaut [ non tagué ] --
Période de vérification	-- Par défaut [Toujours] --
Nb maximum de tentatives de confirmation du statut de l'hôte	Par défaut [ 2 ]
Intervalle entre les vérifications (minutes)	Par défaut [ 1 ]
Intervalle de nouvelles tentatives de confirmations d'état (minutes)	Par défaut [ 1 ]
Temps maximum d'exécution d'un check (secondes)	Par défaut [ 60 → Défini dans shinken.cfg:check_running_timeout ]
Seuil d'alerte de l'utilisation CPU (secondes)	Par défaut [ 5 → Défini dans shinken.cfg:warning_threshold_cpu_usage ]

## Partie passif

### Staging > Hôte

Hôte > Validé Customer 1

Général

Propriété	Valeur	Venant des modèles
Temps maximum d'exécution d'un check (secondes)	Par défaut [ 60 → Défini dans shinken.cfg.check_running_timeout ]	
Seuil d'alerte de l'utilisation CPU (secondes)	Par défaut [ 5 → Défini dans shinken.cfg.warning_threshold_cpu_usage ]	

Passif (Shinken accepte les états reçus depuis des outils externes pour cet élément)

Passif activé  Vrai  Faux Par défaut [Vrai]

Vérification que l'état reçu des outils externes ne soit pas expiré  Vrai  Faux Par défaut [Faux]

Seuil d'expiration des états reçus des outils externes (secondes)

Données stockées

Métriologie

Lire et stocker les métriques  Vrai  Faux Par défaut [Vrai]

SLA

Seuil d'avertissement  99.000 % Par défaut [99.000%]

Seuil critique  97.000 % Par défaut [97.000%]

## Etats possibles d'un hôte

Les hôtes vérifiés peuvent être dans trois états différents :

- UP
- DOWN
- UNREACHABLE

## Détermination de l'état de l'hôte

Les vérifications d'hôtes sont faites par des [commandes](#), qui retournent un état soit **OK**, **WARNING**, **UNKNOWN**, ou **CRITICAL**. Shinken Enterprise traduit les codes retour des sondes par un état d'hôte qui est soit **UP**, **DOWN**, ou **UNREACHABLE**. La table ci-joint montre les correspondances entre les codes retours et l'état associé. Certains sous-process (décrits plus loin) peuvent modifier l'état final de l'hôte.

Résultats de Commande	Etat de l'hôte
OK	UP
WARNING	DOWN
UNKNOWN	DOWN
CRITICAL	DOWN

Si l'état principal de l'hôte est **DOWN**, Shinken Enterprise va tenter de déterminer si l'hôte est réellement **DOWN** ou s'il est juste **UNREACHABLE**. La différence entre **DOWN** et **UNREACHABLE** est importante, car elle permet de déterminer la réelle cause source du problème. Le tableau joint montre comment Shinken Enterprise définit le statut final en fonction du statut du parent ( tel que précisé dans la définition de l'hôte )

Etat de l'hôte précédent	Etat de l'hôte parent	Etat de l'hôte
DOWN	Au moins un parent est UP	DOWN
DOWN	Tous les parents sont, soit DOWN ou UNREACHABLE	UNREACHABLE

## Changement d'état d'un hôte

Comme vous le savez certainement, un hôte ne reste jamais dans le même état tout le temps. Quand Shinken Enterprise vérifie le statut d'un hôte, il est capable de détecter un changement d'état entre **UP**, **DOWN**, et **UNREACHABLE** et de prendre les actions appropriées .

Ces changements d'état résultent en différents types ( **HARD** or **SOFT** ), qui peuvent lancer des événements et des notifications. Détecter et gérer tous ces changements d'état est l'essence même de Shinken Enterprise .

Lorsque l'état d'un hôte change trop souvent, il est considéré comme étant en "**flapping**". Exemple un serveur qui se redémarre à chaque fois que l'OS charge.

Shinken Enterprise peut détecter quand un hôte entre en statut flapping, et peut alors bloquer l'envoi de notifications tant que l'état n'est pas stabilisé. Plus d'informations disponibles dans [Elements en Flapping](#).

## Date d'expiration du statut

Le statut d'un l'élément dépend aussi de la **durée d'expiration du statut** , si cette durée est expirée le statut d'élément devient **Données manquantes** .

La durée d'expiration du statut dépendant:

- Si la vérification est **passive** : de la valeur de la propriété "**Seuil d'expiration des états reçus des outils externes**" (clé d'import : *freshness\_threshold*). Par défaut si aucune valeur n'est attribuée le statut dure jusqu'au prochain changement d'état.
- Si la vérification est **active** et que l'état du statut est en mode "**HARD**" : de la valeur de la propriété "**Intervalle entre les vérifications**" (clé d'import : *check\_interval*)
- Si la vérification est **active** et que l'état du statut est en mode "**SOFT**" : de la valeur de la propriété "**Intervalle de nouvelles tentatives de confirmations d'état**" (clé d'import : *retry\_interval*)

## Checks et dépendances

Vous pouvez définir des parents sur un hôte afin de ne pas avoir à vérifier le statut de tous les hôtes dépendants. Plus d'informations disponibles dans la page [Logique des modèles](#).

## Parallélisation des Checks

Dans le Shinken Enterprise, tous les checks sont lancés en parallèle.