Mode actif et mode passif

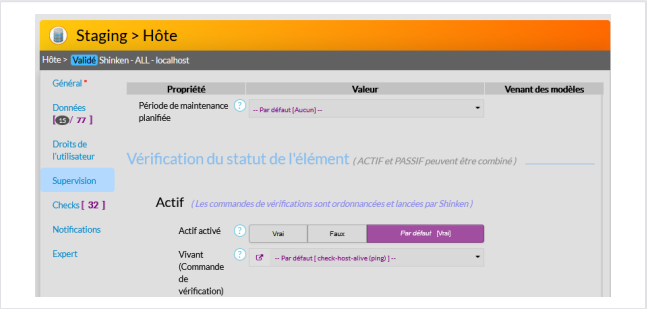
Sommaire

Contexte
Mode actif
Fonctionnement du mode actif
Activer le mode actif
Paramétrer la fréquence de vérification
Mode passif
Usages du mode passif
Fonctionnement du mode passif
Activer le mode passif
Soumettre les résultats de checks/hôtes passifs
Activer le mode passif sur les checks et hôtes
Gestion de l'expiration des statuts reçus
Si le mode actif et passif sont activés
Si seul le mode passif est activé
Exemple de gestion de l'expiration de l'état d'un check en mode passif à l'aide de la commande cmd-check__shinken-build-result

Contexte

Shinken propose deux modes de supervision des hôtes et des checks :

- **Actif** : Shinken planifie et exécute directement les commandes de vérification. L'exécution des vérifications se fait à intervalles réguliers.
- **Passif :** Shinken accepte et intègre les résultats depuis des outils externes, sans lancer lui-même les commandes de vérification. Ce mode est particulièrement adapté aux éléments dont le statut ne peut pas être déterminé efficacement à intervalles fixes — par exemple, des processus asynchrones comme des jobs batch ou des sauvegardes — ou encore des équipements situés derrière un pare-feu, empêchant une connexion directe depuis un Poller.



Ces deux modes peuvent être combinés. Il est ainsi possible de planifier des vérifications via Shinken (*mode actif*) tout en acceptant des résultats issus d'outils externes (*mode passif*).

Par défaut, les deux modes sont activés sur les hôtes et les checks.

Mode actif

Fonctionnement du mode actif

Les caractéristiques du mode actif sont :

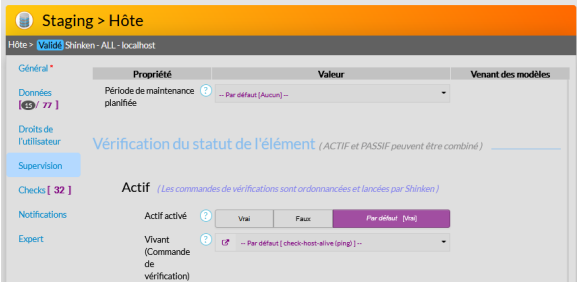
- La sonde est exécutée par le Poller.
- Elle est lancée à intervalles réguliers.

Fonctionnement détaillé du mode actif :

- **Ordonnancement par le Scheduler** : Le Scheduler est responsable de la planification des vérifications. Chaque check ou hôte dispose d' une fréquence de vérification configurable, définissant à quel intervalle il doit être supervisé.
- **Exécution de la sonde par le Poller** : Le Poller exécute la commande de vérification associée à l'élément. Cette commande :
  - collecte les données nécessaires depuis l'équipement supervisé,
  - analyse ces données pour déterminer :
    - le **statut** (*OK, ATTENTION, CRITIQUE, INCONNU*),
    - le **résultat court**,
    - le **résultat long**,
    - les **métriques.**
  - Le Poller transmet ensuite ces informations au Scheduler.
- **Traitement par le Scheduler** : Le Scheduler analyse les résultats reçus et, selon l'état et la configuration, déclenche les actions appropriées. Cela peut inclure l'envoi de notifications ou l'exécution de commandes via le gestionnaire d'événements, qui seront ensuite prises en charge et exécutées par un **Reactionner**.

Activer le mode actif

Le mode actif pour les checks et les hôtes est activé par défaut dans Shinken.



La configuration du mode actif se trouve dans l'onglet Supervision de l'élément.

La propriété **Actif activé** (*clé d'import active_checks_enabled*) permet d'activer ou de désactiver ce mode, soit sur un hôte (*voir la page* Éditer un Hôte), soit sur un check (*voir la page* Editer un check appliqué à un hôte).

Paramétrer la fréquence de vérification

La configuration de l'intervalle des vérifications dans Shinken est paramétrable pour chaque hôte ou chaque check.

Deux paramètres contrôlent la fréquence des contrôles :

- **Intervalle entre les vérifications** (*clé d'import : check_interval*) : Lorsque le statut d'un élément est **confirmé** (*voir la page* Statut confirmé (HARD) et non confirmé (SOFT)) , il est vérifié périodiquement selon cet intervalle.
- **Intervalle de nouvelles tentatives de confirmations d'état** (*clé d'import : retry_interval*) : Après le premier statut **NON-OK**, le statut devient non confirmé. Tant que le statut reste non confirmé, les vérifications sont réalisées périodiquement selon cet intervalle. Une fois l'état **confirmé** à nouveau, l'intervalle revient à celui défini par **l'Intervalle entre les vérifications.**

Sur l'**Interface de Visualisation**, il est possible de forcer la vérification du statut. Cette action n'impacte pas la planification des prochaines vérifications.

Mode passif

Usages du mode passif

Le **mode passif** est adapté dans les situations suivantes :

- **Checks asynchrones** : Certains processus sont naturellement asynchrones et ne peuvent pas être efficacement supervisés par des vérifications à intervalles réguliers. Cela concerne notamment des tâches comme les **jobs batch** ou les **sauvegardes**, où il n'est pas possible de prédire précisément le moment de leur achèvement ou d'un changement d'état.
- **Équipements protégés par un pare-feu** : Lorsque des équipements sont situés derrière un pare-feu qui empêche le Poller de lancer des sondes pour interroger leur état, le mode passif devient indispensable.

Exemples de checks asynchrones nécessitant d'être supervisés en mode passif :

- **Traps SNMP et alertes de sécurité** : Le nombre de traps SNMP ou d'alertes de sécurité générés sur une période donnée est imprévisible. Les superviser à intervalles fixes serait inefficace. Le mode passif permet de capturer ces événements immédiatement dès leur survenue.

- **Intervalle de vérification trop court** : Parfois, l'intervalle de vérification requis est inférieur à l'intervalle minimum d'ordonnancement possible dans Shinken (*une minute*). Dans ce cas, il est possible de superviser à une fréquence plus importante l'élément avec un script ou un agent et de transmettre les résultats à Shinken via un check avec le mode passif activé.

Fonctionnement du mode passif

Les caractéristiques du mode passif sont :

- La sonde n'est pas exécutée par le Poller mais par un outil externe/script.
- Les résultats de la sonde sont envoyés au Receiver.
- La fréquence des résultats dépend de la fréquence d'envoi des résultats par l'outil externe.

Fonctionnement détaillé du mode passif :

- **Vérification du statut** : Une application externe se charge de la vérification du statut et de récupérer les informations
- **Transmission des résultats au Receiver** : L'outil externe transmet les données au module de type "`ws-arbiter`" sur le Receiver (*voir la page* Module receiver-module-webservice) selon le format supporté par Shinken (*voir la page* Les Sondes).
- **Traitement par le Scheduler** : Le Scheduler analyse les résultats reçus et, selon l'état et la configuration, déclenche les actions appropriées. Cela peut inclure l'envoi de notifications ou l'exécution de commandes via le gestionnaire d'événements, qui seront ensuite prises en charge et exécutées par un **Reactionner.** Le traitement est similaire en mode passif et en mode actif.

Activer le mode passif

Soumettre les résultats de checks/hôtes passifs

Pour pouvoir soumettre un résultat passif dans l'architecture Shinken, il faut :

- Activer le module de type "`ws-arbiter`" sur le Receiver (*voir la page* Module receiver-module-webservice).
- Transmette les résultats à l'API exposée par le module du Receiver (*port 7760 par défaut*).
- S'assurer que le résultat des sondes respectent le format supporté par Shinken (*voir la page* Les Sondes).

Activer le mode passif sur les checks et hôtes

Le mode passif est activé par défaut pour les hôtes et les checks dans Shinken.

La configuration du mode passif se trouve dans l'onglet Supervision de l'élément.

La propriété **Passif activé** (*clé d'import passive_checks_enabled*) permet d'activer ou de désactiver ce mode sur un hôte (*voir la page* Éditer un Hôte) ou sur un check (*voir la page* Editer un check appliqué à un hôte *ainsi que ses pages sœurs*).

Il est important de noter que l'activation du mode passif pour un hôte et pour ses checks **n'est pas liée**. Par exemple, un hôte peut avoir le mode passif désactivé, tout en ayant un ou plusieurs checks avec le mode passif activé.

Gestion de l'expiration des statuts reçus

En mode passif, la mise à jour de l'état d'un élément est assurée par un outil externe.

L'absence de résultat peut être ambiguë : elle peut signifier qu'aucune erreur n'est survenue, ou au contraire qu'un problème empêche la transmission des données (*défaillance réseau, arrêt du script*).

Pour détecter ces situations, Shinken permet de définir un **seuil d'expiration des états reçus des outils externe** (*clé d'import freshness_threshold*). Dès que ce seuil est dépassé, une commande de vérification est automatiquement exécutée afin de confirmer l'état réel du service.

Lorsque Shinken ne reçoit aucun résultat pendant une durée supérieure à ce seuil :

- Il considère le statut comme **expiré**.
- Il exécute alors la commande définie par la propriété **Commande de vérification** (*clé d'import check_command*).
- Cette commande sera relancée **périodiquement**, à un intervalle correspondant au seuil d'expiration, **jusqu'à ce qu'un nouveau résultat soit reçu**.

La propriété **Vérification que l'état reçu des outils externes ne soit pas expiré** (*clé d'import check_freshness*) permet d'activer ou désactiver cette fonctionnalité pour chaque hôte ou check.

> La propriété **Commande de vérification** est commune aux modes passif et actif. Lorsque les deux modes sont activés, **la même commande** est utilisée à la fois pour les vérifications planifiées par Shinken et pour celles déclenchées automatiquement en cas **d'expiration** du résultat.

Si le mode actif et passif sont activés

La gestion de l'expiration des statuts ne sera active que si l'élément est **exclusivement** en mode passif.

Si un élément combine les modes actif et passif, le mécanisme d'expiration des statuts ne s'applique pas même s'il est configuré.

Si seul le mode passif est activé

Si la gestion de l'expiration des statuts n'est pas activée, l'absence de résultats peut masquer des problèmes critiques, tels qu'une coupure réseau ou une défaillance de l'outil externe.

Il est donc fortement recommandé d'activer le mécanisme d'expiration des statuts afin de confirmer l'état réel du service.

Deux approches sont possibles :

- **Développer une commande spécifique** : Cette commande a pour rôle de vérifier s'il existe effectivement un problème lorsque le seuil d' expiration est dépassé. Par exemple, elle peut contrôler l'état des équipements réseau pour déterminer si leur indisponibilité explique l' absence de retours.
- **Utiliser la commande livrée par Shinken cmd-check__shinken-build-result** : Lorsque le développement d'une commande dédiée n'est pas envisageable, Shinken propose cette commande générique. Elle permet de construire le retour de la sonde (*statut, résultat, résumé et métriques*) à partir d'arguments.
    - Couplée au mécanisme d'expiration des statuts, elle peut générer une erreur dès que le seuil est franchi, afin d'alerter rapidement sur un risque lié à l'absence de réponse.
    - Grâce à l'utilisation des **variables** (*voir la page* Les Variables (Remplacement dynamique de contenu - Anciennement les Macros)), il est également possible de réutiliser les résultats de la dernière vérification pour générer un retour cohérent et adapté à la situation.

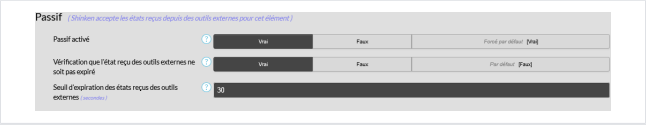
La commande **cmd-check__shinken-build-result** dispose des options suivantes :

Option	Valeur par défaut	Description
--status	**0**	Statut que l'on souhaite imposer à l'élément. Pour un check : • Valeurs acceptées : 0 (***OK***) , 1 (***WARNING***), 2 (***CRITICAL***), 3 (***UNKNOWN***) • Variable pour utiliser le précédent statut : $SERVICESTATE$ Pour un hôte : : • Valeurs acceptées : 0, (***OK***), 1, 2 ou 3 (***CRITICAL***) • Variable pour utiliser le précédent statut : $HOSTSTATE$
--output	**""**	Résultat court de la commande. • Variable pour utiliser le précédent résultat court : $SERVICEOUTPUT$ (*check*), $HOSTOUTPUT$ (*hôte*)
--long-output	**""**	Résultat long de la commande • Variable pour utiliser le précédent résultat long: $LONGSERVICEOUTPUT$ (*check*), $LONGHOSTOUTPUT$ (*hôte*)
--perfdata	**""**	Données de performances. • Variable pour utiliser les précédentes données de performances : $SERVICEPERFDATA$ (*check*) , $HOSTPERFDATA$ (*hôte*) Le format est le meme que celui des sondes (*voir la page* Les Sondes). Exemple : • metric_1=14;90;95 metric_2=44;90;95

Exemple de gestion de l'expiration de l'état d'un check en mode passif à l'aide de la commande cmd-check__shinken-build-result

On crée un check appliqué à l'hôte **"CHECK WITH PASSIVE MODE"** avec le mode passif d'activé et le mode actif de désactivé.

On active l'expiration du statut avec un seuil d'expiration de 30 secondes.



On utilise comme commande de vérification, la commande fournie par Shinken **cmd-check__shinken-build-result.**

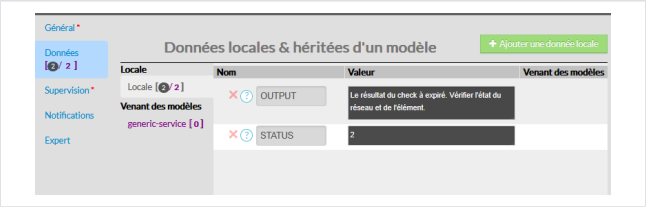
La commande utilise trois paramètres :

- La donnée du check STATUS.
- La donnée du check OUTPUT.

- La variable : $LONGSERVICEOUTPUT$.

On crée les données sur le check.

- STATUS : 2
- OUTPUT : Le résultat du check a expiré. Vérifier l'état du réseau et de l'élément.



Il faut ensuite attacher le check sur un hôte en le pousser en production.

Si Shinken ne reçoit aucune donnée pour ce check pendant plus de 30 secondes, la commande de vérification est automatiquement exécutée afin de signaler une suspicion sur l'état du service.
Dans l'Interface de Visualisation, le check apparaît alors en erreur, accompagné du résultat court défini, et du précédent résultat long.