

Grafana - v8.3.2

Contexte

Le Pack Shinken est inclus dans votre installation de Shinken Entreprise afin d'avoir une bonne visibilité de la santé de votre architecture de supervision.

Afin de pouvoir superviser les démons de l'architecture, la source [cfg-file-shinken](#) comprend un certain nombre de modèles d'hôte qu'il faudra importer et appliquer sur vos serveurs hébergeant vos démons Shinken.

Les différents modèles mis à votre disposition sont les suivants :

- shinken-arbiter
- shinken-broker
- shinken-broker-module-sla
- shinken-poller
- shinken-reactionner
- shinken-receiver
- shinken-scheduler
- shinken-synchronizer

Une fois les modèles appliqués, les checks apparaîtront dans l'onglet "Checks" de votre hôte.

Voici un exemple des différents checks du pack Shinken sur un hôte.

Sommaire

- [Introduction](#)
- [Installation](#)
- [Connexion avec Graphite](#)
- [Intégration dans Shinken](#)
- [Créer un tableau de bord \(dashboard \)](#)
- [Récupération de l'URL à intégrer dans Shinken](#)
- [Passage en HTTPS](#)
- [Lien vers le mapping nomuuid nécessaire pour Grafana, et suivi des requêtes](#)
- [Mise à jour d'une version supérieure à 5.4.0](#)
- [Authentification avec le widget Page Web](#)

? Unknown Attachment

Arbiter

Modèle d'hôte correspondant: **shinken-arbiter** (notez que ce modèle hérite du modèle **shinken**)

Afin de superviser le démon Arbiter, le modèle **shinken-arbiter** appliqué à votre hôte, attachera deux checks qui vérifieront la santé et la performance de ce démon.

Checks

Nom du Check	Description	Exemple de résultat
Arbiter - Alive	Vérifie que le démon Arbiter peut être correctement contacté sur le réseau.	? Unknown Attachment
Arbiter - Performance	Retourne le temps de connexion au démon Arbiter ainsi que la liste des connexions avec les autres démons de l'architecture.	? Unknown Attachment

Paramétrage des Checks

Les checks de l'Arbiter peuvent être configurés via des données fournies par le modèle.

Les données suivantes sont disponibles:

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec l'Arbiter	http	shinken
ARBITER_PORT	Port utilisé pour l'établissement de la connexion avec l'Arbiter	7770	Locale

Broker

Modèle d'hôte correspondant: **shinken-broker** (notez que ce modèle hérite du modèle **shinken** et **mongodb-no-replication**)

Afin de superviser le démon Broker, le modèle **shinken-broker** appliqué à votre hôte, attachera plusieurs checks qui vérifieront la santé et la performance de ce démon.

Checks

Nom du Check	Description	Exemple de résultat
Broker - Alive	Vérifie que le démon Broker peut être correctement contacté sur le réseau.	? Unknown Attachment
Broker - Performance API Connection	Vérifie les temps de connexion vers le Broker	? Unknown Attachment
Broker - Performance Modules Queues	Vérifie les performances des modules du Broker	? Unknown Attachment
Broker Daemon Module SLA	Vérifie le bon fonctionnement du module SLA	? Unknown Attachment

Paramétrage des Checks

Les checks du Broker peuvent être configurés via des données fournies par le modèle.

Les données suivantes sont disponibles:

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec le Broker	http	shinken
BROKER_PORT	Port utilisé pour l'établissement de la connexion avec Broker	7772	Locale
MONGO_CONNECTION_METHOD	Méthode de connexion au serveur MongoDB ("direct" ou "ssh")	ssh	mongodb-no-replication
MONGO_SSH_USER	Utilisateur distant avec lequel se connecter au serveur hébergeant MongoDB	shinken	mongodb-no-replication
MONGO_SSH_KEYFILE	Clé SSH privée permettant la connexion au serveur hébergeant MongoDB	~/.ssh/id_rsa	mongodb-no-replication

D'autres données sont également héritées du modèle mongodb-no-replication. Ces données sont documentées dans le pack Mongo. (prochainement)

Poller

Modèle d'hôte correspondant: **shinken-poller** (notez que ce modèle hérite du modèle **shinken**)

Afin de superviser le démon Poller, le modèle **shinken-poller** appliqué à votre hôte, attachera plusieurs checks qui vérifieront la santé et la performance de ce démon.

Checks

Nom du check	Description	Exemple de résultat
Poller - Running Well	Vérifie que le Poller est joignable sur le réseau, affiche ses tags et le statut de connexion avec les schedulers	? Unknown Attachment
Poller - Performance	Affiche les statistiques des performances de l'exécution des checks dans le Poller	? Unknown Attachment

Paramétrage des checks

Les checks du Poller peuvent être configurés via des données fournies par le modèle.

Les données suivantes sont disponibles pour le Poller:

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec le Poller	http	shinken
POLLER_PORT	Port utilisé pour établir la connexion avec le poller	7771	Locale

NB_CHECK_IN_TIMEOUT_TOLERANCE	Nombre de checks en timeout provoquant une sortie en erreur du check	0	Locale
ACTIVE_POLLER_LATENCY	Latence de connexion (en secondes) au-delà de laquelle le check sort en erreur	0.5	Locale

Reactionner

Modèle d'hôte correspondant: **shinken-reactionner** (notez que ce modèle hérite du modèle **shinken**)

Afin de superviser le démon Reactionner, le modèle **shinken-reactionner** appliqué à votre hôte, attachera plusieurs checks qui vérifieront la santé et la performance de ce démon.

Checks

Nom du check	Description	Exemple de résultat
Reactionner - Running Well	Vérifie que le Reactionner est joignable sur le réseau, affiche ses tags et le statut de connexion avec les schedulers	? Unknown Attachment
Reactionner - Performance	Affiche les statistiques des performances de l'exécution des checks dans le Reactionner	? Unknown Attachment

Paramètre du check

Les checks du Reactionner peuvent être configurés via des données fournies par le modèle.

Les données suivantes sont disponibles pour le Reactionner:

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec le Reactionner	http	shinken
REACTIONNER_PORT	Port utilisé pour établir la connexion avec le Reactionner	7769	Locale
NB_CHECK_IN_TIMEOUT_TOLERANCE	Nombre de checks en timeout provoquant une sortie en erreur du check	0	Locale
ACTIVE_REACTIONNER_LATENCY	Latence de connexion (en secondes) au-delà de laquelle le check sort en erreur	0.5	Locale

Receiver

Modèle d'hôte correspondant: **shinken-receiver** (notez que ce modèle hérite du modèle **shinken**)

Afin de superviser le démon Receiver, le modèle **shinken-receiver** appliqué à votre hôte, attachera plusieurs checks qui vérifieront la santé et la performance de ce démon.

Checks

Nom du check	Description	Exemple de résultat
Receiver - Alive	Vérifie que le démon Receiver peut être correctement contacté sur le réseau.	? Unknown Attachment
Receiver - Performance API Connection	Vérifie la latence de connexion au Receiver et ses performances	? Unknown Attachment

Paramètres du check

Les checks du Receiver peuvent être configurés via des données fournies par le modèle.

Les données suivantes sont disponibles pour le Receiver:

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
------------------	-------------	-------------------	-----------------------------------

SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec le Receiver	http	shinken
RECEIVER_PORT	Port utilisé pour établir la connexion avec le Receiver	7773	Locale

Scheduler

Modèle d'hôte correspondant: **shinken-scheduler** (notez que ce modèle hérite du modèle **shinken**)

Afin de superviser le démon Scheduler, le modèle **shinken-scheduler** appliqué à votre hôte, attachera plusieurs checks qui vérifieront la santé et la performance de ce démon.

Checks

Nom du check	Description	Exemple de sortie
Scheduler - Running Well	Vérifie que le démon Scheduler peut être correctement contacté sur le réseau, Affiche le nombre d'élément qu'il gère dans un tableau, en distinguant les hôtes, les Clusters et les Checks. Vous aurez également le Total des éléments.	? Unknown Attachment
Scheduler - Performance	Affiche l'utilisation CPU ainsi que les statistiques des Pollers et des Reactionners qu'il gère. Si des limites de CPU ou de RAM sont atteintes sur vos Pollers, alors ces limites seront affichées. Si certains de vos Pollers sont en spare, ils seront également affichés, avec le texte (SPARE) après leurs noms.	? Unknown Attachment ? Unknown Attachment

Paramètres du check

Les checks du Scheduler peuvent être configurés via des données fournies par le modèle.. Les paramètres SSH sont décrits sur la page [Rétention MongoDB](#)

Les données suivantes sont disponibles pour le Scheduler:

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec le Scheduler	http	shinken
SCHEDULER_PORT	Port utilisé pour établir la connexion avec le scheduler	7768	Locale
PASSIVE_POLLER_LATENCY	Latence de connexion (en secondes) au-delà de laquelle le check sort en erreur	0.5	Locale

Synchronizer

Modèle d'hôte correspondant: **shinken-synchronizer** (notez que ce modèle hérite du modèle **shinken** et **mongodb-no-replication**)

Afin de superviser le démon Synchronizer, le modèle **shinken-synchronizer** appliqué à votre hôte, attachera plusieurs checks qui vérifieront la santé et la performance de ce démon.

Checks

Nom du check	Description	Exemple de résultat
Synchronizer - Alive	Vérifie que le démon Synchronizer peut être correctement contacté sur le réseau.	? Unknown Attachment
Synchronizer - Performance API Connection	Vérifie la latence de connexion au Synchronizer et ses performances	? Unknown Attachment

Paramètre du check

Les checks du Synchronizer peuvent être configurés via des données fournies par le modèle.. Les paramètres SSH sont décrits sur la page [Rétention MongoDB](#)

Les données suivantes sont disponibles pour le Synchronizer:

Nom de la donnée	Description	Valeur par défaut	Hérité du modèle d'hôte ou locale
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec le Scheduler	http	shinken
SYNCHRONIZER_PORT	Port utilisé pour établir la connexion avec le Synchronizer	7765	Locale
MONGO_CONNECTION_METHOD	Méthode de connexion au serveur MongoDB ("direct" ou "ssh")	ssh	mongodb-no-replication
MONGO_SSH_USER	Utilisateur distant avec lequel se connecter au serveur hébergeant MongoDB	shinken	mongodb-no-replication
MONGO_SSH_KEYFILE	Clé SSH privée permettant la connexion au serveur hébergeant MongoDB	~/.ssh/id_rsa	mongodb-no-replication

D'autres données sont également héritées du modèle mongodb-no-replication. Ces données sont documentées dans le pack Mongo. (prochainement)

Cas particuliers

Détection de la haute disponibilité

Les modèles d'hôte du pack Shinken permettent de détecter les environnements distribués avec des démons configurés en Spare.

Dans le cas où un démon est spare, une pastille bleue "SPARE" est affichée à côté du statut de la vérification:

? Unknown Attachment

Détection d'erreurs de communications entre démons

Les checks du pack Shinken ont la capacité de détecter les problèmes de communications entre les démons.

En effet, si le démon supervisé lance un appel (API) et qu'il échoue, alors l'erreur sera écrite dans les logs de ce démon ([Les fichiers Logs](#)) et un Warning sera remonté :

? Unknown Attachment

La vérification se base sur les dernières 24h. Si toutes les communications aboutissent correctement durant les dernières 24h, le message disparaîtra et vous retrouverez un état OK.

i Les erreurs de communication peuvent ne pas être graves et ne pas avoir d'incidence sur votre architecture Shinken. Cependant, si vous recevez des erreurs de communications et que vous avez des doutes sur l'origine de ces problèmes de communications, par prévention, envoyez nous votre log pour analyse.