

Onglet Graphiques

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

- [Principe](#)
- [Sélection des métriques](#)
- [Sélection de la période et approximation](#)
- [Graphes](#)
- [Echelle de temps](#)

? Unknown Attachment

Arbiter

Modèle d'hôte correspondant: **shinken-arbiter**

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
ARBITER_PORT	Port utilisé pour l'établissement de la connexion avec l'Arbiter	7770
SHINKEN_PROTOCOL	Protocole utilisé pour établir la connexion avec l'Arbiter	http

Broker

Modèle d'hôte correspondant: **shinken-broker**

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
BROKER_PORT	Port utilisé pour l'établissement de la connexion avec Broker	7772

Poller

Modèle d'hôte correspondant: *shinken-poller*

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
POLLER_PORT	Port utilisé pour établir la connexion avec le poller	7771
NB_CHECK_IN_TIMEOUT_TOLERATE	Nombre de checks en timeout provoquant une sortie en erreur du check	0
ACTIVE_POLLER_LATENCY	Latence de connexion (en secondes) au-delà de laquelle le check sort en erreur	0.5

Reactionner

Modèle d'hôte correspondant: *shinken-reactionner*

Afin de superviser le démon Reactionner, 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
Reactionner - Running Well	Checks that the Reactionner daemon can be reached over the network, displays its tags and the connection status with the schedulers	? Unknown Attachment
Reactionner - Performance	Display performance statistics of notifications execution in the 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
REACTIONNER_PORT	Port utilisé pour établir la connexion avec le poller	7769
ACTIVE_REACTIONNER_LATENCY	Latence de connexion (en secondes) au-delà de laquelle le check sort en erreur	0.5

Receiver

Corresponding host template: *shinken-receiver*

Checks

Check name	Description	Example Output
Receiver - Alive	Checks that the Receiver can be reached over the network	? Unknown Attachment
Receiver - Performance API Connection	Checks connection time to the Receiver and its performances	? Unknown Attachment

Check settings

Receiver checks can be configured via datas provided by the template.

The following datas are available for the Receiver:

Data name	Description	Default Value
RECEIVER_PORT	Port used to establish the connection with the Receiver	7769

Scheduler

Corresponding host template: *shinken-scheduler*

Checks

Check name	Description	Example Output
Scheduler - Running Well	Checks that the Scheduler daemon can be reached over the network, displays how many elements it is managing	? Unknown Attachment
Scheduler - Performance	Displays CPU usage as well as poller and reactionners statistics.	? Unknown Attachment

Check settings

Scheduler checks can be configured via datas provided by the template. SSH parameters are fully described in [Rétention MongoDB](#)

The following datas are available for the Scheduler:

Data name	Description	Default Value
SCHEDULER_PORT	Port used to establish the connection with the Scheduler	7768

PASSIVE_POLLER_LATENCY	Connection latency above which the check will issue an error. (in seconds)	0.5
MONGO_CONNECTION_METHOD	Connection method to the MongoDB server ("direct" or "ssh")	ssh
MONGO_SSH_USER	User with whom to connect to the distant server	shinken
MONGO_SSH_KEYFILE	SSH private key file to use to connect to the distant server	~/.ssh/id_rsa

Synchronizer

Corresponding host template: *shinken-synchronizer*

Checks

Check name	Description	Example Output
Synchronizer - Alive	Checks that the Scheduler daemon can be reached over the network	? Unknown Attachment
Synchronizer - Performance API Connection	Checks connection time to the Synchronizer and its performances	? Unknown Attachment

Check settings

Synchronizer checks can be configured via datas provided by the template. SSH parameters are fully described in [Rétenion Mongodb](#)

The following datas are available for the Synchronizer:

Data name	Description	Default Value
SYNCHRONIZER_PORT	Port used to establish the connection with the Synchronizer	7765
MONGO_CONNECTION_METHOD	Connection method to the MongoDB server ("direct" or "ssh")	ssh
MONGO_SSH_USER	User with whom to connect to the distant server	shinken
MONGO_SSH_KEYFILE	SSH private key file to use to connect to the distant server	~/.ssh/id_rsa