

Pack Shinken

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

Sommaire

- Contexte
- Cas particuliers
 - Détection de la haute disponibilité
 - Détection d'erreurs de communication entre démons
 - Détection des différences de versions
 - Détection du mode Passif des démons Poller et Reactionner

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.

Nom	Duplicate For each	Groupe d'instance	Module d'hôte	Commande de check	Statut du check
Arbiter - Allue		arbiter	shinken-arbiter	check_shinken_arbiter.py	  
Arbiter - Performance		arbiter	shinken-arbiter	check_shinken_arbiter.py	  
Broker - Allue		broker	shinken-broker	check_shinken_broker.py	  
Broker - Performance API Connection		broker	shinken-broker	check_shinken_brokerapi_connection.py	  
Broker - Performance Module Queue		broker	shinken-broker	check_shinken_brokermodule_queue.py	  
Polier - Performance		poller	shinken-poller	check_shinken_poller.py	  
Polier - Running Watch		poller	shinken-poller	check_shinken_poller.py	  
Reactionner - Performance		reactionner	shinken-reactionner	check_shinken_reactionner.py	  
Reactionner - Running Watch		reactionner	shinken-reactionner	check_shinken_reactionner.py	  
Receiver - Allue		receiver	shinken-receiver	check_shinken_receiver.py	  
Scheduler - Performance		scheduler	shinken-scheduler	check_shinken_scheduler.py	  
Scheduler - Running Watch		scheduler	shinken-scheduler	check_shinken_scheduler.py	  
Synchronizer - Allue		synchronizer	shinken-synchronizer	check_shinken_synchronizer.py	  

Cas particuliers

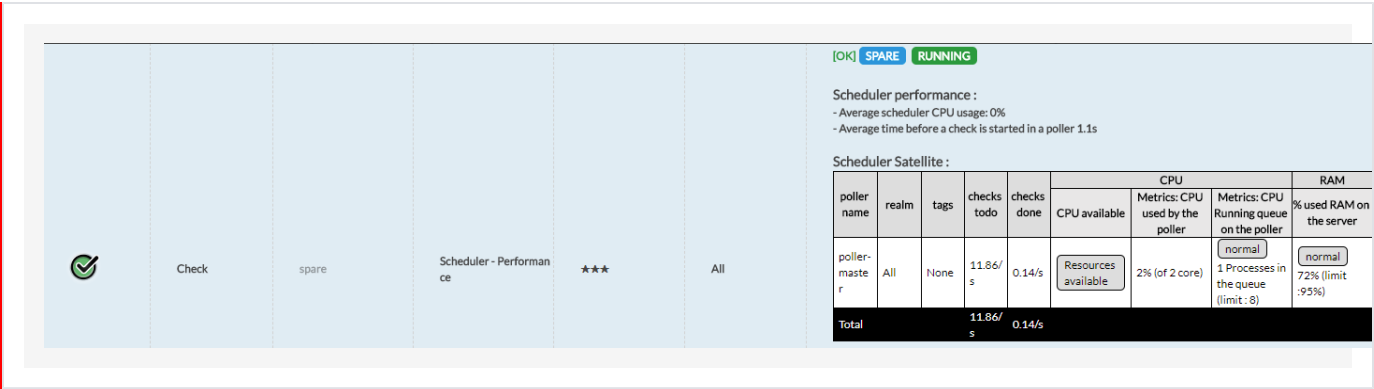
Détection de la haute disponibilité

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:

	Contexte	Statut	Type	Nom de l'élément	Nom de check	Adresse	Priorité	Royaume	Résultat	Résultat Long
	Pas de sélection	Pas de sélection	Pas de sélection	LAB VM5	Saisir un nom de check	Saisir une adresse	Pas de sélection	Pas de sélection	spare	Saisir du texte
			Hôte	LAB VM5		172.16.0.5	***	All	PING OK - Packet loss = 0%, RTA = 0.37 ms	
			Check	LAB VM5	Scheduler - Performance		***	All	 SPARE No elements to schedule	This scheduler is currently idle because it is configured as a Spare. It will take over another scheduler when a main scheduler stop working
			Check	LAB VM5	Scheduler - Running Watch		***	All	 SPARE Scheduler on realm All.	This scheduler is currently idle because it is configured as a Spare. It will take over another scheduler when a main scheduler stop working

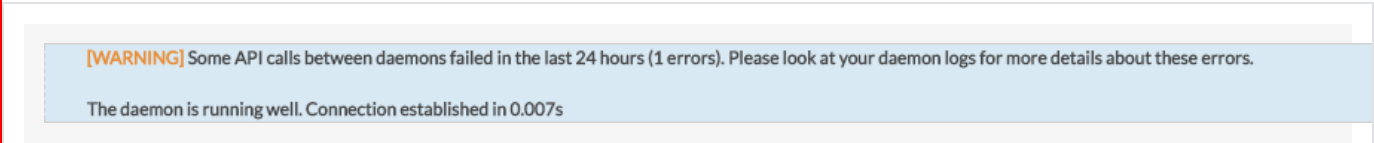
Lorsqu'un Spare prend le relais, une pastille "RUNNING" est rajoutée à la mention "SPARE" :



Détection d'erreurs de communication entre démons

Les checks du pack Shinken ont la capacité de détecter les problèmes de communication 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 :



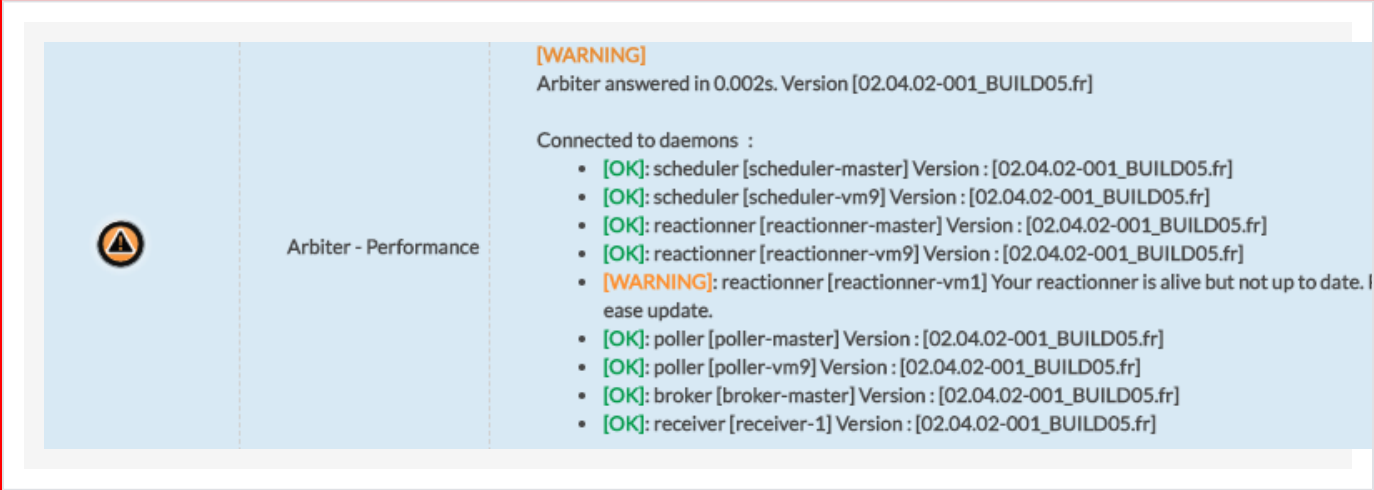
La vérification se base sur les dernières 24h. Si toutes les communications aboutissent correctement durant les dernières 24h, le message disparaîtra et vous retrouverez un état OK.

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 et que vous avez des doutes sur l'origine de ces problèmes de communication, par prévention, envoyez nous votre log pour analyse.

Détection des différences de versions

Les checks du pack Shinken sont également capables de détecter des différences de version entre les démons, et d'afficher l'information dans le retour du check.

Par exemple ici le retour du check "Arbiter - Performance" :



Par exemple ici le retour du check "Scheduler - Performance", indiquant une différence de version avec un Poller :

time before a check is started in a poller 0.83s

er Satellite :

realm	tags	checks todo	checks done	CPU			RAM
				CPU available	Metrics: CPU used by the poller	Metrics: CPU Running queue on the poller	% used RAM on the server
All	None	13.08/s	13.02/s	Resources available	13% (of 4 core)	normal 4 Processes in the queue (limit : 4)	normal 36% (limit :95%)
All	tag- vm9		WARNING : Your poller is alive but its version (02.04.02-001_BUILD03.fr) is not the same as its arbiter (02.04.02-001_BUILD04.fr). Please update.				
		13.08/s	13.02/s				

Détection du mode Passif des démons Poller et Reactionner

Les checks "Poller - Running Well" et "Reactionner - Running Well" du pack Shinken permettent de détecter et donc afficher si un démon est en mode passif.

Voici un exemple avec le check du Reactionner :

Check	sup2sup	Reactionner - Running Well	★★★	All	[OK] Your reactionner is running well. Version [02.04.01-005_BUILD11.fr] Reactionner [PASSIVE MODE] • tags : None
-------	---------	----------------------------	-----	-----	---