shinken-poller

Contexte

Le modèle shinken-poller vous permet de superviser un hôte hébergeant le démon Poller.

Sommaire

- Contexte
- Description du modèle
  - Checks
  - Paramétrage des checks
  - Données de performances
  - Détail des commandes
- Interprétation des données de l'état de santé du Poller
  - Statistiques de l'état de santé du Poller
  - Exemple d'un état de santé dégradé du Poller
- Interprétation des statistiques de performance du Poller
  - Statistiques générales sur l'exécution des checks
  - Charge du Poller
  - Utilisation du CPU
  - Utilisation de la mémoire (RAM)
  - Vérification de la charge de la machine

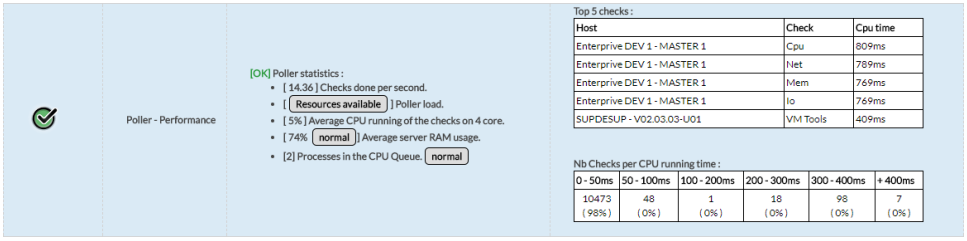
Description du modèle

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 avec son numéro de version, affiche ses tags et le statut de connexion avec les schedulers	
Poller - Performance	Affiche les statistiques des performances de l'exécution des checks dans le Poller	

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_TOLERATE	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

Données de performances

Les checks du modèle **shinken-poller** enregistrent des données de performance, qui peuvent ensuite être affichées dans l'interface de Visualisation sur l'Onglet Graphes ou bien le Widget Graphique.

Nom du check	Nom de la métrique	Explication

Poller - Running Well	nb_check_in_timeout	Nombre de checks qui sont entrés en timeout sur le Poller pendant les 20 dernières minutes. Cette durée est configurable dans le fichier de configuration du Poller, avec l'option *keep_timeout_time* (par défaut 1200s)
Poller - Performance	nb_action_done_per_sec	Nombre moyen de checks exécutés par le Poller (moyenne glissante calculée sur 1mn)
Poller - Performance	poller_load_state	Booléen (0 ou 100) qui indique si le Poller a atteint sa charge maximale: ▪ **0**: Le Poller n'a pas atteint sa charge maximale, il peut encore exécuter des checks supplémentaires ▪ **100**: Le Poller a atteint sa charge maximale, il ne peut pas exécuter plus de checks.
Poller - Performance	cpu_usage_percent	Pourcentage de temps processeur utilisé par le Poller
Poller - Performance	used_ram_percent	Pourcentage de mémoire utilisée sur le serveur

Détail des commandes

Nom du check	Commande du check	Ligne de commande
Poller - Performance	check_shinken_poller!**stats**	$PLUGINSDIR$/check_shinken -H "$HOSTADDRESS$" -p "$_HOSTPOLLER_PORT$" --shinkenversion "$SHINKENVERSION$" -t poller -m $ARG1$ --active_poller_latency "$_HOSTACTIVE_POLLER_LATENCY$" --check_tolerate "$_HOSTNB_CHECK_IN_TIMEOUT_TOLERATE$"
Poller - Running Well	check_shinken_poller!**alive**	$PLUGINSDIR$/check_shinken -H "$HOSTADDRESS$" -p "$_HOSTPOLLER_PORT$" --shinkenversion "$SHINKENVERSION$" -t poller -m $ARG1$ --active_poller_latency "$_HOSTACTIVE_POLLER_LATENCY$" --check_tolerate "$_HOSTNB_CHECK_IN_TIMEOUT_TOLERATE$"

Les modes dépréciés ("-m") :

- api_connection
- cpu_load
- overload_protection

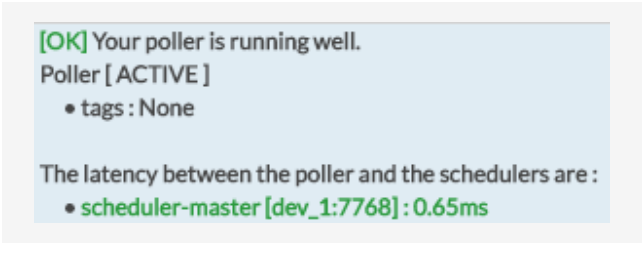
Interprétation des données de l'état de santé du Poller

Statistiques de l'état de santé du Poller

Le check "*Poller - Running Well*" vérifié l'état de santé du Poller supervisé.

Il remonte plusieurs informations sur le statut du Poller:

- Une mention qui indique si le Poller est actif ou passif. Dans l'exemple, on a affaire à un Poller actif.
- La liste des tags du Poller. Dans l'exemple, aucun tag n'est défini sur le Poller.
- La latence de connexion avec le ou les schedulers du royaume et des sous-royaumes.



Exemple d'un état de santé dégradé du Poller

Lorsque le Poller rencontre des problèmes qui peuvent signifier un fonctionnement dégradé, ces problèmes sont également remontés dans le résultat du check "*Poller - Running Well*".

Lorsque certains checks ont un temps d'exécution supérieur aux différents seuils définis, ils sont répertoriés dans le résultat du check "*Poller - Running Well*" pour permettre d'avertir des potentielles erreurs sur ces checks ou du Poller.

Dans l'exemple, le check nous signale que 10 checks ont pris plus que 10 secondes pour s'exécuter pendant les 20 dernières minutes, et ont leur exécution a donc été arrêtée de force.

[WARNING] You have 10 (> 0 tolerate) check(s) that timed out in the last 20m 0s.
The last actions that timed out :

Nb	Last	Host	Check	Timeout
1	16:49:59	LAB-VM23	sleep 7	10s
1	16:49:58	LAB-VM23	sleep 6	10s
1	16:49:58	LAB-VM23	sleep 2	10s
1	16:49:58	LAB-VM23	sleep 10	10s
1	16:49:58	LAB-VM23	sleep 5	10s

Poller [ACTIVE]
• tags : None

La durée de temps (en secondes) pendant laquelle les checks sont conservés dans l'ensemble des checks en timeout est configurable dans le fichier de configuration du Poller concerné.

Il est également possible de définir une seuil de tolérance afin d'afficher un avertissement seulement lorsqu'il y a plus qu'un certain nombre de checks en timeout. Pour cela, il faut modifier la donnée NB_CHECK_IN_TIMEOUT_TOLERATE dans l'hôte concerné dans la configuration.

/etc/shinken/pollers/mon_poller.cfg

```
define poller {
        ...
    keep_timeout_time    1200
    ...
}
```

Interprétation des statistiques de performance du Poller

La supervision d'un démon Poller présente un grand nombre de statistiques de performances qui permettent de visualiser le travail effectué par le Poller. Le Poller a pour rôle d'executer les checks.

Il est ainsi souvent un des premiers démons que l'on regarde lorsqu'on veut visualiser combien de checks sont exécutés. Aussi, pour dimensionner correctement une installation Shinken Entreprise, il est important de pouvoir visualiser combien de checks un Poller est capable d'exécuter pour pouvoir décider d'ajouter ou retirer des Pollers de la configuration.

Les checks du Poller fournis dans le pack Shinken fournissent donc un grand nombre de données sur les performances du Poller.

Statistiques générales sur l'exécution des checks

La première statistique remontée par le check est le nombre de checks effectués par seconde par le Poller supervisé.

Dans cet exemple, le Poller effectue 1.9 checks par seconde en moyenne, tous checks confondus.

[OK] Poller statistics :
- [1.9] Checks done per second.

Dans le Résultat long du check "*Poller - Performance*", deux tableaux présentent des statistiques sur les temps d'exécution des checks dans le Poller.

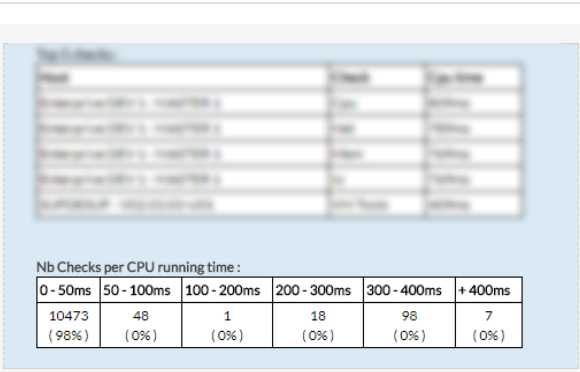
- Le premier tableau affiche les 5 checks consommant le plus de temps CPU parmi l'ensemble des checks exécutés sur le Poller. Dans ce tableau, pour chaque check est affiché le nom du check, l'hôte sur lequel il est accroché, et le temps d'exécution du check.

Top 5 checks :

Host	Check	Cpu time
Enterprive DEV 1 - MASTER 1	Cpu	809ms
Enterprive DEV 1 - MASTER 1	Net	789ms
Enterprive DEV 1 - MASTER 1	Mem	769ms
Enterprive DEV 1 - MASTER 1	Io	769ms
SUPDESUP - V02.03.03-U01	VM Tools	409ms

- Un deuxième tableau présente la répartition du temps d'exécution des checks. Dans le tableau en exemple, on voit que les checks

exécutés sur ce Poller s'exécutent majoritairement en moins de 50ms.



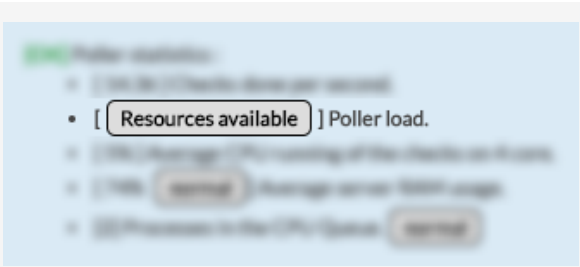
- Si le temps moyen d'exécution des checks est très long, ou très court, il est possible de modifier les réglages des périodes de temps du tableau dans le fichier de configuration du Poller concerné. Dans le fichier de configuration du Poller en question, l'option "*exec_stat_range*" doit être modifiée.

/etc/shinken/pollers/mon_poller.cfg

```
define poller {
...
# Ranges for the check : poller statistics
#exec_stat_range    50, 100, 200, 300, 400
...
}
```

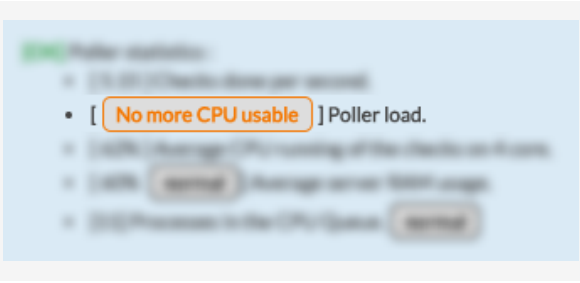
Charge du Poller

Le check du Poller fournit aussi la charge du Poller. Il s'agit d'un indicateur général indiquant si le Poller peut encore supporter des checks supplémentaires, ou si il est chargé au maximum. Cet indicateur n'est pas lié aux autres indicateurs de performances de la machine (File d'attente CPU, mémoire).



Une pastille orange précédant la mention "Poller load" signifie que le Poller ne peut plus prendre de checks supplémentaires. Cette pastille indique que le Poller fonctionne à la vitesse maximale qui lui est permise par le processeur.

C'est donc un signe indiquant qu'il faudrait ajouter un Poller supplémentaire dans l'architecture Shinken.

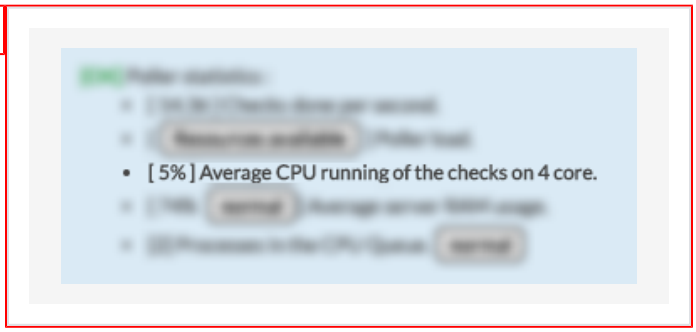


Utilisation du CPU

Le Poller se limite intelligemment selon différents critères pour éviter de surcharger inutilement la machine sur laquelle il est exécuté. Le premier critère de limitation est l'utilisation du CPU de la machine par le démon (et l'exécution des checks).

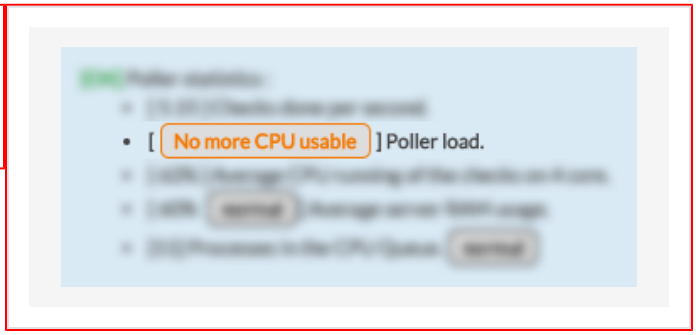
Si le Poller se rend compte que le CPU de la machine sur laquelle il est exécuté est surchargé, il se régule pour exécuter moins de checks et éviter de rendre la machine inutilisable.

Dans le check "*Poller - Performance*", la quantité de CPU utilisée par les checks est affichée dans le Résultat du check.



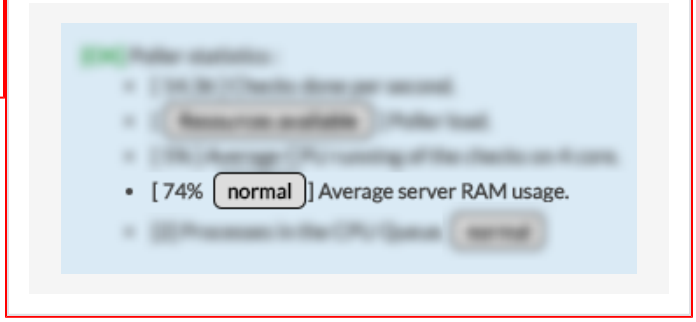
Si le CPU de la machine hébergeant le Poller est utilisé de manière intensive, le résultat du check "*Poller - Performance*" avertit de l'utilisation trop élevée du CPU. Dans ce cas, le Poller se limite et n'exécute plus de checks supplémentaires tant que l'utilisation du CPU est trop élevée.

La ligne indiquant la charge du Poller indique alors que le CPU est surchargé.

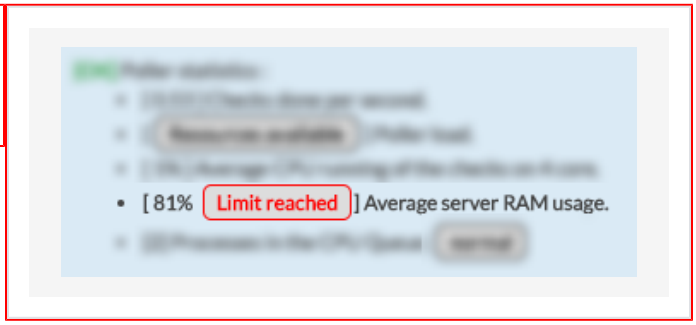


Utilisation de la mémoire (RAM)

Si le CPU n'est pas le facteur limitant pour le fonctionnement du Poller, l'état de la mémoire du serveur est alors vérifiée. Dans le résultat du check "*Poller - Performance*", l'utilisation de la mémoire sur le serveur est indiquée.



Si l'utilisation de la mémoire (RAM) sur le serveur dépasse le seuil définit dans la configuration du Poller, le résultat du check "*Poller - Performance*" affiche une avertissement indiquant l'utilisation excessive de la mémoire. Lorsque cet avertissement est affiché, le Poller n'exécute plus de checks supplémentaires tant que l'utilisation de la mémoire est supérieure au seuil défini.



Comme mentionné précedemment, le réglage du seuil est effectué dans le fichier de configuration du Poller concerné.

/etc/shinken/pollers/mon_poller.cfg

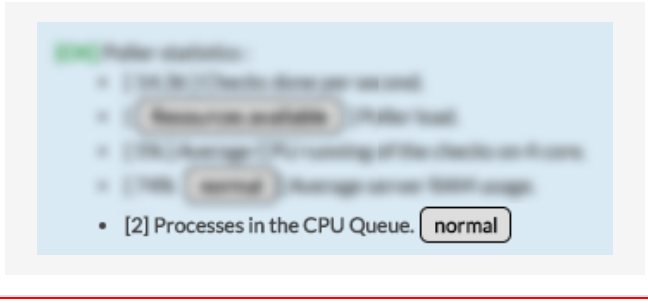
```
define poller {
...
# Percentage of used physical RAM beyond which the poller will not launch
any new check
max_ram_percent           95
...
}
```

Vérification de la charge de la machine

Le Poller possède une dernière mesure de performance qu'il observe et selon laquelle il limite son activité. Après la vérification de l'utilisation du CPU du Poller et de la mémoire du serveur, le Poller vérifie le nombre de processus dans la file d'attente du processeur.

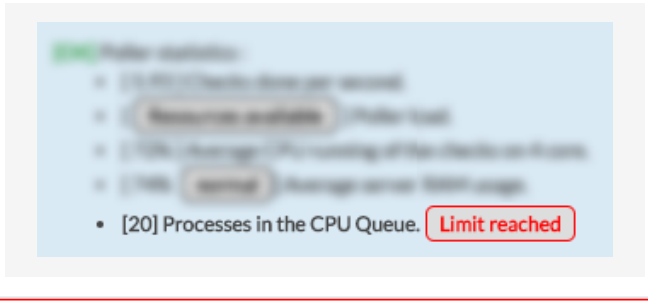
En fonction de la limite de processus dans la file d'attente du processeur, le Poller décide si il faut exécuter des checks supplémentaires.

Le nombre de processus dans la file d'attente du processeur est affiché dans le résultat du check "*Poller - Performance*".



Le nombre de processus dans la file d'attente du processeur est aussi accompagné d'une pastille indiquant le statut de cette mesure. Lorsque le nombre de processus est supérieur à la limite définie dans la configuration, cette pastille indique ce dépassement.

Dans ce cas, le Poller n'exécutera plus de checks supplémentaires tant que le nombre de processus dans la file d'attente du processeur sera supérieur au seuil choisi.



Le seuil mentionné précedemment se définit dans le fichier de configuration du Poller concerné. On note que ce seuil se définit par coeur du processeur, et pas globalement pour le processeur.

Ainsi, avec une limite définie à 3, et un processeur possédant 4 cœurs, le Poller ne se limitera que lorsqu'il y a plus de 12 processus dans la file d'attente du processeur.

/etc/shinken/pollers/mon_poller.cfg

```
define poller {
...
# Number of maximum runnings processes in the CPU Queue per CPU. Default
value is 4.
max_cpu_queue_per_cpu    4
...
}
```