

Les logs du Poller

Sommaire

- Logs spécifiques à ce démon au démarrage
 - Gestion de la configuration
 - Premier chargement de la configuration
 - Mise à jour de la configuration
 - Mise à jour des liens vers d'autres démons
 - Récupération des checks
 - Pour récupérer les checks à exécuter
 - Envoi des résultats de checks au Scheduler
 - Surcharge serveur en activité disque, ralentissant l'écriture des logs
 - Logs concernant les checks de vérifications Shinken
 - Suivi des lancements de sondes par les workers et de leurs performances
 - Les Workers arrivent à lancer toutes leurs sondes ou se limitent
 - Il manque des ressources CPU / RAM
 - Les ressources CPU / RAM sont suffisantes
 - PROBLEME: Un worker n'arrive pas à lancer ses checks
 - Le total des sondes et leurs performances au sein du poller
 - Logs de chargement des modules
 - Échanges entre le processus principal et les Workers
 - Suivi des actions en cours
 - Dans le démon (boucle principale)
 - Dans un Worker
 - Log de performance de la boucle du Poller
 - Performance des boucles des Workers
 - Un Worker est lent
 - Un Worker est en retard
 - Logs d'exécution des commandes de vérification
 - Gestion du logger
 - Activation du logger
 - Log d'activation du logger
 - Désactivation du logger
 - Log de désactivation du logger
 - État du logger
 - Logs d'exécution
 - Exécution normale
 - Exécution en erreur

Les fichiers de log du Poller sont situés dans le dossier `/var/log/shinken/`. Pour plus d'informations, consultez la page [Fichiers Logs](#).

Logs spécifiques à ce démon au démarrage

Au démarrage et tous les jours à minuit, en plus des logs communs à tous les démons, le Poller indique l'état du logger optionnel d'exécution :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ START-DAEMON ] [ LOGGERS CONFIGURATION ]
-----
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ START-DAEMON ] [ LOGGERS CONFIGURATION ] Optional loggers
activation :
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ START-DAEMON ] [ LOGGERS CONFIGURATION ] - DISABLED : [
CHECK EXECUTION ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ START-DAEMON ] [ LOGGERS CONFIGURATION ]
-----
```

Ce logger peut être activé pour suivre l'exécution des commandes de vérification (voir la section [Logs d'exécution des commandes de vérification](#)).

Au démarrage du démon, les lignes suivantes indiquent les limites systèmes qui sont appliquées :

Démarrage du daemon

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ SYSTEM ] System resource number of open files is
set to (soft:1024 / hard:1024 ) (from parameter max_file_descriptor_limit)
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ SYSTEM ] System resource number of processes
/threads is set to (soft:unlimited / hard:unlimited ) (set at system max values)
```

Avec comme informations principales:

- le nombre de fichiers/socket ouvrables,
- le nombre max de processus/threads.

Gestion de la configuration

Premier chargement de la configuration


Lorsque le Poller reçoit sa configuration pour la première fois, deux logs **INFO** sont affichés.

- Le premier indiquant que le Poller rentre dans la phase de chargement d'une nouvelle configuration.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] ----- Loading the new configuration from the arbiter
```

- Le deuxième indiquant que le Poller a reçu la configuration de l'Arbiter.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter send us a new configuration: [configuration_uuid=configuration-uuid, arbiter=arbiter-name, architecture=architecture-name, date=YYYY-MM-DD HH:MM:SS]
```

-  Dans le cas où le Poller n'est pas de la même version que l'Arbiter et que l'option `*mismatch_version_error*` est activée sur l'Arbiter:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ poller-name ] Incompatible daemon version : Your Arbiter daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY] while this daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY]. Refusing this configuration.
```

-  Dans le cas où le Poller n'est pas de la même version que l'Arbiter et que l'option `*mismatch_version_error*` est désactivée sur l'Arbiter:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-name ] Incompatible daemon version : Your Arbiter daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY] while this daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY].
```

Mise à jour de la configuration

Lorsqu'il y a une mise à jour de la configuration, deux logs en **INFO** sont affichés.

- Le premier indiquant que le Poller rentre dans la phase de chargement d'une nouvelle configuration.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] [ UPDATE ] ----- Loading a configuration update from the arbiter
```

- Le deuxième indiquant que le Poller a reçu la configuration de l'Arbiter.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] [ UPDATE ] The arbiter send us a new configuration: [configuration_uuid=configuration-uuid, arbiter=arbiter-name, architecture=architecture-name, date=YYYY-MM-DD HH:MM:SS]
```

-  Dans le cas où le Poller n'est pas de la même version que l'Arbiter et que l'option `*mismatch_version_error*` est activée sur l'Arbiter:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ poller-name ] Incompatible daemon version : Your Arbiter daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY] while this daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY]. Refusing this configuration.
```

-  Dans le cas où le Poller n'est pas de la même version que l'Arbiter et que l'option `*mismatch_version_error*` est désactivée sur l'Arbiter:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-name ] Incompatible daemon version : Your Arbiter daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY] while this daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY].
```

Mise à jour des liens vers d'autres démons

Lorsque l'Arbiter détecte un changement de lien entre les démons, quatre logs en **INFO** seront affichés.

- Ces deux logs affichent le(s) lien(s) du(des) démon(s) supprimé(s).

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter asked us to remove daemons:  
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] - REMOVED scheduler : [name=scheduler1-name]  
[shard_id= XXX] [uri=http://scheduler_address:port/]
```

- Ces deux logs affichent le(s) lien(s) du(des) démon(s) ajouté(s).

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter send us new daemons:  
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] + ADDED scheduler : [name=scheduler2-name]  
[shard_id= XXX] [uri=http://scheduler_address:port/]
```

Récupération des checks

Pour récupérer les checks à exécuter

- **Poller actif**
 - Le Poller va demander au Scheduler les checks qu'il doit exécuter.
 - Le Poller indique un temps de travail dont il dispose (*en temps cpu*).
 - Le Scheduler lui donne des checks pour le temps demandé (*suivant le temps d'exécution moyen constatés pour les checks sur ce Poller*). Il lui donne pour un temps inférieur ou égal au temps disponible.
 - Un log permet d'avoir le nombre de checks récupérés. Ce log s'affiche même si aucun check n'a été récupéré :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS ] [ scheduler-master ] [ GET ] Requesting checks  
todo from this scheduler for 2.000s cpu time [received=3 check(s) for 0.039s cpu time]
```

- **Poller passif**
 - Le Scheduler demande au Poller le temps CPU disponible.
 - Le Scheduler lui envoie des checks à traiter pour le temps disponible (suivant le temps d'exécution moyen pour les checks constatés sur ce Poller). Il lui donne pour un temps inférieur ou égal au temps disponible.
 - Si des checks sont reçus, un log permet d'avoir le nombre de checks récupérés selon le temps de travail disponible sur le Poller :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS ] [ scheduler-master ] [ RECEIVED ] We received  
checks todo from this scheduler for 2.000s cpu time [received=1 check(s) for 0.317s cpu time]
```

Envoi des résultats de checks au Scheduler

- **Poller actif**
 - Une fois les commandes exécutées, le Poller envoie les résultats au Scheduler.
 - Un log permet d'avoir le nombre de résultats de checks envoyés au Scheduler et le temps mis pour être envoyé.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS RESULTS ] [ scheduler-master ] [ PUSHED ] 1 check's  
result(s) sends to this scheduler in [0.043]s
```

- **Poller passif**
 - Une fois les checks exécutés, le Poller stocke les résultats en attendant que le Scheduler vienne les récupérer.
 - A chaque tour de boucle du Scheduler, ce dernier demande au Poller s'il a des résultats de checks disponibles.
 - Si des résultats sont disponibles, un log permet d'avoir le nombre de résultats checks renvoyés au Scheduler.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS RESULTS ] [scheduler-master] [ GIVEN ] 1 check's result(s) given to answer scheduler request
```

Surcharge serveur en activité disque, ralentissant l'écriture des logs

Si le serveur hébergeant le démon est surchargé en termes d'IO disques sur le volume qui héberge le fichier de log, alors ce dernier va mettre du temps à s'écrire et va ralentir tout le démon. Il faut alors si c'est faisable isoler le volume des disques sur un disque moins chargé pour ne pas ralentir le démon.

En cas de soucis, il sera écrit les lignes suivantes :

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
-----
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ] [ WRITING ] The log write times is very high (1.87s). Please look at your log disk performance.
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
-----
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
```

Logs concernant les checks de vérifications Shinken

Quand un check de supervision du démon est fait, on va avoir plusieurs entrées dans les logs qui concernent des données que le démon garde sur diverses statistiques.

Un log permet d'avoir le temps pris sur le calcul des dernières commandes en timeout:

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] Compute "Checks in timeouts" stats : 0.000s in a total of 2048 commands in timeouts
```

Un log permet d'avoir le temps de calcul concernant les ranges d'exécution des checks/notifications en fonction du temps (<100ms, <400ms, etc):

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] Compute "Checks per CPU running time" : 0.000s (on a total of 2048 checks)
```

Un log permet d'avoir le temps de calcul pour avoir les 5 commandes les plus longues en temps CPU:

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] top5 execution time 0.003s (loop over 1 ranges and 343 elements)
```

Un dernier log permet d'avoir le temps complet du calcul des statistiques du démon:

- Sur une partie commune à tous les démons (version, chemins, etc.)
- Sur la partie qui concerne uniquement ce qui concerne ce démon, donc sont inclus ici les temps précédents
- Il est affiché en :
 - **DEBUG**: si le temps de calcul est inférieur à la valeur du paramètre `display_statistics_compute_time_if_higher` du démon.
 - **INFO**: si le temps de calcul est supérieur ou égal au paramètre `display_statistics_compute_time_if_higher` du démon.

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] Daemon stats were computed in 0.020s (0.001 for daemon common part, 0.020 for poller part)
```

En cas d'affichage INFO on met un petit morceau en plus sur comment gérer le niveau de log:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ STATS          ] Daemon stats were computed in 0.004s (0.000 for daemon common part, 0.004 for poller part) (NOTE: log is displayed in INFO because 0.004 is higher than display_statistics_compute_time_if_higher=1ms in the daemon cfg)
```

Le démon nettoie ses structures de statistiques toutes les 5minutes, ce qui sera vu par la ligne de log suivante:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ STATS          ] Clean checks in timeouts structure in 0.000s (before clean: 0 commands in timeouts, after clean: 0)
```

Suivi des lancements de sondes par les workers et de leurs performances

Les Workers arrivent à lancer toutes leurs sondes ou se limitent

On peut savoir si les workers manquent de disponibilité CPU/RAM en suivant les logs.

Il manque des ressources CPU / RAM

Si des ressources (*CPU, RAM*) ont manqué sur ce tour de boucle, alors on aura les logs suivants

- des lignes en WARNING mettent en évidence les problèmes,
- les lignes avec MISSING RESOURCE indiquent quelles ressources limitent le fonctionnement des workers.

```
[YYYY-MM-DD HH:MM:SS] INFO   : [ poller-name ] [ POLLER STATISTICS ] 6 checks to run
[YYYY-MM-DD HH:MM:SS] INFO   : [ poller-name ] [ POLLER STATISTICS ] => 5 are waiting in workers [ WORKER
1: 1 ] [ WORKER 2: 4 ] ( Estimated CPU time : 0.473998s )
[YYYY-MM-DD HH:MM:SS] WARNING: [ poller-name ] [ POLLER STATISTICS ]      --> Launched 13/18 checks in this
loop [ WORKER 1: 6/7 ] [ WORKER 2: 7/11 ]
[YYYY-MM-DD HH:MM:SS] WARNING: [ poller-name ] [ POLLER STATISTICS ]      --> Wait time for system resources
availability [ WORKER 1: 0.414s/1.057s ] [ WORKER 2: 0.835s/1.032s ]
[YYYY-MM-DD HH:MM:SS] INFO   : [ poller-name ] [ POLLER STATISTICS ] => 1 still to be dispatched to workers
( Estimated CPU time: 0.137000s )
[YYYY-MM-DD HH:MM:SS] INFO   : [ poller-name ] [ POLLER STATISTICS ]      --> Sent this turn [ WORKER 1: 7 ]
[ WORKER 2: 8 ]
[YYYY-MM-DD HH:MM:SS] INFO   : [ poller-name ] [ SYSTEM LIMITS ]          CPU limit OK [ CPU %
use: 76.2% 71.4% 96.9% 81.3% ( limit: 80% ) ]
[YYYY-MM-DD HH:MM:SS] WARNING: [ poller-name ] [ SYSTEM LIMITS ] [ MISSING RESOURCE ] CPU running queue
limit REACHED 36 / 16
[YYYY-MM-DD HH:MM:SS] INFO   : [ poller-name ] [ SYSTEM LIMITS ]          RAM limit OK 94.0% /
95.0%
[YYYY-MM-DD HH:MM:SS] INFO   : [ poller-name ] [ WORKER LIMITS ]         WORKER running process
nb limit OK [ WORKER 1: 10/256 ] [ WORKER 2: 6/256 ]
```

Les ressources CPU / RAM sont suffisantes

Si tous les checks ont pu être lancés, on aura les logs suivants (*en INFO*) :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] 8 checks to run
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] => 0 are waiting in workers [ WORKER
1: 0 ] [ WORKER 2: 0 ] ( Estimated CPU time : 0.000000s )
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] --> Launched 7/7 checks in this
loop [ WORKER 1: 3/3 ] [ WORKER 2: 4/4 ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] --> Wait time for system resources
availability [ WORKER 1: 0.000s/1.046s ] [ WORKER 2: 0.000s/1.001s ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] => 8 still to be dispatched to workers
( Estimated CPU time: 1.315000s )
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] --> Sent this turn [ WORKER 1: 3 ]
[ WORKER 2: 4 ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ SYSTEM LIMITS ] CPU limit OK [ CPU %
use: 61.6% 50.6% 63.9% 54.2% ( limit: 80% ) ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ SYSTEM LIMITS ] CPU running queue
limit OK 3 / 16
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ SYSTEM LIMITS ] RAM limit OK 82.0% /
95.0%
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ WORKER LIMITS ] WORKER running process
nb limit OK [ WORKER 1: 4/256 ] [ WORKER 2: 4/256 ]
```

PROBLEME: Un worker n'arrive pas à lancer ses checks

Si le worker n'a pas réussi à lancer de check pendant plus de 5 secondes, le log suivant sera présent :

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ poller-name ] [ WORKER WORKER_ID ] is full since [XX.XXXs] because it has
more than 1.5s of check to manage ( => YY checks pending with estimated Z.ZZZZZs cpu time )
```

où :

- WORKER_ID : est l'id du worker.
- XX.XXX : est la durée depuis laquelle le Poller n'envoie plus de check à traiter au worker.
- YY : est le nombre de checks à lancer en attente sur le worker.

Le total des sondes et leurs performances au sein du poller

On peut suivre les performances du Poller via le log suivant, pour les checks provenant des Schedulers & Synchronizers :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] [ PERFS ] [ CURRENT LOOP ] Waiting to
be pushed to workers 10 checks, Estimated CPU Time: 1.50s
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] [ PERFS ] [ CURRENT LOOP ] Waiting to
be returned to Schedulers/Synchronizer 7 checks, Consumed CPU Time: 1.25s
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] [ PERFS ] [ CURRENT LOOP ] [ WORKERS ]
In workers [ WORKER 1: 10 ( Estimated CPU time: 1.51s ) + in queues ( from daemon:0, to daemon:7 ) ] [
WORKER 2: 11 ( Estimated CPU time: 1.74s ) + in queues ( from daemon:0, to daemon:6 ) ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] [ PERFS ] [ CURRENT LOOP ] [ WORKERS ]
[ Launched = 10 checks, Estimated CPU Time: 1.27s ] [ Executing = 21 checks, Estimated CPU Time: 3.25s ] [
Finished = 6 checks, Consumed CPU Time: 2.07s, -0.01s from Estimated CPU Time ]
```

Avec :

- Estimated CPU Time= temps CPU pur (sans les appels réseaux et autres sleep) mesuré en moyenne sur ce check,
- Consumed CPU Time= temps CPU mesuré sur CE dernier lancement,
- +0.03s from Estimated CPU Time = différence de temps CPU (en + ou en -) qu'on a eu par rapport à leurs exécutions moyennes passées,
- Launched = dans le worker, démarré sur ce tour de boucle,
- Executing = dans le worker, en cours d'exécution (lancé ce tour, ou un tour précédent),
- Finished = terminé et retourné au Poller,
- Waiting to be returned = dans le processus principal, en attente d'être renvoyés vers les Schedulers/Synchronizers.

Et les mêmes informations, mais sur une moyenne d'une minute glissante :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] [ PERFS ] [ lmin AVG ] Waiting to be
pushed to workers 10.87 checks/s, Estimated CPU Time: 1.77s
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] [ PERFS ] [ lmin AVG ] Waiting to be
returned ] 6.31 checks/s, Consumed CPU Time: 1.09s
[2026-01-19 17:59:40] INFO : [ poller-name ] [ POLLER STATISTICS ] [ PERFS ] [ lmin AVG ] [ WORKERS ] [
Launched = 6.37 checks/s, Estimated CPU Time: 1.08s ] [ Executing = 5.45 checks/s, Estimated CPU Time: 1.022
s ] [ Finished = 6.25 checks/s, Consumed CPU Time (total): 1.49s, -0.03s from Estimated CPU Time ]
```

Note: en cas de charge, on doit observer des valeurs sensiblement équivalentes pour les données suivantes:

- Launched,
- Finished,
- Waiting to be returned.

Cela représente le débit du Poller.

Logs de chargement des modules

Les démons ont une phase de chargement des modules qui est décrite dans la page [GLOBAL - Les logs de gestion des modules - chapitre \[MODULES-MANAGER \]](#)

Échanges entre le processus principal et les Workers

Suivi des actions en cours

Dans le démon (boucle principale)

Le log suivant indique

- le nombre d'actions en attente d'être envoyées à un worker, dans le démon principal

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER STATISTICS ] => 1 still to be dispatched to workers
( Estimated CPU time: 0.137000s )
```

Dans un Worker

Le log suivant détaille l'état du Worker :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ WORKER 1 STATISTICS ] last loop activity:0.156s ago, last
action launched:0.287s ago, checks received from main daemon:6 ( 0.882999s CPU time ), checks returned to
main daemon:4 ( 0.454400s CPU time ), in queue [ received from Poller:0 (0b), sent to Poller:1 (2991b) ], in
worker checks [ todo:1 running:10 / total:11 ] = ( 1.988996s estimated CPU time )
```

Log de performance de la boucle du Poller

Les logs suivants permettent de suivre le temps d'exécution de la boucle principale du Poller

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ === Loop start === ] [ Loop number=XXX ] =====
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ === Loop stop === ] [ Loop number=XXX ] [PERF] [ X.XXX ]s
```

Performance des boucles des Workers

Un Worker est lent

⚠ Quand le tour de boucle d'un Worker n'a pas fini avant un certain délai, le log suivant signale la lenteur observée.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-name ] [ WORKER X ] is slow, last tick was X.XXXs ago, over limit
of X.XXXs
```

Un Worker est en retard

✘ Quand le tour de boucle d'un Worker prend beaucoup trop de temps, le log suivant signale le retard observé.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ poller-name ] [ WORKER X ] is late, last tick was X.XXXs ago, over limit of X.XXXs
```

Logs d'exécution des commandes de vérification

Gestion du logger

L'activation ou la désactivation du logger optionnel se fait via l'utilisation d'une commande **curl** depuis le shell.

Les paramètres requis sont :

- **POLLER_IP** : l'adresse IP ou le nom du serveur où tourne le Poller (ou **localhost** si le shell est lancé sur le serveur du Poller directement),
- **POLLER_PORT** : le port du Poller,
- **POLLER_PROTOCOLE** :
 - **https** si le protocole HTTPS a été activé sur le démon (voir le paramètre *use_ssl* dans le fichier */etc/shinken/daemons/pollerd.ini* du démon),
 - **http** sinon,
- **LOGGER_ID** : l'identifiant du logger, soit :
 - **CHECK_EXECUTION** pour les commandes de vérification.

Le résultat de cette commande renvoie un document JSON.

Activation du logger



Suivant l'activité du Poller, l'activation de ce logger peut générer un gros volume de données.

Sur un site de production, il est conseillé de limiter l'activation de ce logger sur de courtes périodes (*par exemple une heure au plus*), afin de limiter le surplus d'occupation disque engendré.

Commande

```
curl -s -k "POLLER_PROTOCOLE://POLLER_IP:POLLER_PORT/set_log_enable?logger_id=LOGGER_ID&enable=1"
```

Exemple

```
curl -s -k "http://localhost:7771/set_log_enable?logger_id=CHECK_EXECUTION&enable=1"
```

La commande doit renvoyer la sortie :

Exemple

```
{"message": "logger [Shinken] [CHECK EXECUTION CONTROL] is enabled"}
```

Log d'activation du logger

Quand le logger est activé, le log suivant est généré :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ INFO-DAEMON ] [ LOGGERS CONFIGURATION ] - ENABLING : [ XXXX EXECUTION ]
```

Exemple

```
[2025-11-06 16:45:36] INFO : [ poller-master ] [ INFO-DAEMON ] [ LOGGERS CONFIGURATION ] - ENABLING : [ CHECK EXECUTION ]
```

Désactivation du logger

Commande

```
curl -s -k "POLLER_PROTOCOLE://POLLER_IP:POLLER_PORT/set_log_enable?logger_id=LOGGER_ID&enable=0"
```

Exemple

```
curl -s -k "http://localhost:7771/set_log_enable?logger_id=CHECK_EXECUTION&enable=0"
```

La commande doit renvoyer la sortie :

Exemple

```
{"message": "logger [Shinken] [CHECK EXECUTION CONTROL] is disabled"}
```

Log de désactivation du logger

Quand le logger est désactivé, le log suivant est généré :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ INFO-DAEMON ] [ LOGGERS CONFIGURATION ] - DISABLING : [ XXXX EXECUTION ]
```

Exemple

```
[2025-11-06 16:58:23] INFO : [ poller-master ] [ INFO-DAEMON ] [ LOGGERS CONFIGURATION ] - DISABLING : [ CHECK EXECUTION ]
```

État du logger

Commande

```
curl -s -k "POLLER_PROTOCOLE://POLLER_IP:POLLER_PORT/get_log_info?logger_id=LOGGER_ID"
```

Exemple pour obtenir l'état du logger d'exécution des commandes de notification

```
curl -s -k "http://localhost:7771/get_log_info?logger_id=CHECK_EXECUTION"
```

La commande doit renvoyer une sortie du style :

Exemple pour obtenir l'état du logger d'exécution des commandes de notification

```
[{"id": "CHECK_EXECUTION", "name": "[Shinken] [CHECK EXECUTION CONTROL]", "enable": true}]
```

Logs d'exécution

Exécution normale

Lorsqu'une commande s'exécute, et que le logger d'exécution a été activé, les logs suivants sont générés.

Si la commande termine avec un code de terminaison compris entre 1 et 3 (*inclus*), Shinken considère que la commande a fonctionné correctement (voir la page [Les Sondes](#), section "Code retour").

```
[YYYY-MM-DD HH:MM:SS] INFO : [ WORKER X ] [ XXXXX EXECUTION ] [ DONE ] [ UUID ] [ COMMAND NAME ] :
NOM_DE_LA_COMMANDE
[YYYY-MM-DD HH:MM:SS] INFO : [ WORKER X ] [ XXXXX EXECUTION ] [ DONE ] [ UUID ] [ COMMAND ] :
CHEMIN_DE_LA_COMMANDE ...
[YYYY-MM-DD HH:MM:SS] INFO : [ WORKER X ] [ XXXXX EXECUTION ] [ DONE ] [ UUID ] [ STATUS ] : RESULTAT
D'EXÉCUTION
[YYYY-MM-DD HH:MM:SS] INFO : [ WORKER X ] [ XXXXX EXECUTION ] [ DONE ] [ UUID ] [ RESULT ] : SORTIE
GÉNÉRÉE LIGNE 1
[YYYY-MM-DD HH:MM:SS] INFO : [ WORKER X ] [ XXXXX EXECUTION ] [ DONE ] [ UUID ] [ RESULT ] : SORTIE
GÉNÉRÉE LIGNE 2
```

Exemple

```
[2025-11-06 18:15:29] INFO : [ WORKER 4 ] [ CHECK EXECUTION ] [ DONE ] [
5e38b68a4e8845fd8dfbf47090b01424 ] [ COMMAND NAME ] : nautilus Host-01-//-nautilus Check-04-//-USS-NAUTILUS
command
[2025-11-06 18:15:29] INFO : [ WORKER 4 ] [ CHECK EXECUTION ] [ DONE ] [
5e38b68a4e8845fd8dfbf47090b01424 ] [ COMMAND ] : /var/lib/shinken-user/libexec/uss-nautilus
/nautilus_commande ...
[2025-11-06 18:15:29] INFO : [ WORKER 4 ] [ CHECK EXECUTION ] [ DONE ] [
5e38b68a4e8845fd8dfbf47090b01424 ] [ STATUS ] : 1 (WARNING)
[2025-11-06 18:15:29] INFO : [ WORKER 4 ] [ CHECK EXECUTION ] [ DONE ] [
5e38b68a4e8845fd8dfbf47090b01424 ] [ RESULT ] : output by uss-nautilus
[2025-11-06 18:15:29] INFO : [ WORKER 4 ] [ CHECK EXECUTION ] [ DONE ] [
5e38b68a4e8845fd8dfbf47090b01424 ] [ RESULT ] : |
```

Exécution en erreur

Les commandes de vérification dont le code de terminaison n'est pas compris entre 0 et 3 (*inclus*) sont considérées comme non fonctionnelles (voir la page [Les Sondes](#), section "Code retour").

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ WORKER X ] [ XXXXX EXECUTION ] [ ERROR ] [ UUID ] [ COMMAND NAME ] :
NOM_DE_LA_COMMANDE
[YYYY-MM-DD HH:MM:SS] ERROR : [ WORKER X ] [ XXXXX EXECUTION ] [ ERROR ] [ UUID ] [ COMMAND ] :
CHEMIN_DE_LA_COMMANDE ...
[YYYY-MM-DD HH:MM:SS] ERROR : [ WORKER X ] [ XXXXX EXECUTION ] [ ERROR ] [ UUID ] [ STATUS ] : RESULTAT
D'EXÉCUTION
[YYYY-MM-DD HH:MM:SS] ERROR : [ WORKER X ] [ XXXXX EXECUTION ] [ ERROR ] [ UUID ] [ RESULT ] : SORTIE
GÉNÉRÉE LIGNE 1
[YYYY-MM-DD HH:MM:SS] ERROR : [ WORKER X ] [ XXXXX EXECUTION ] [ ERROR ] [ UUID ] [ RESULT ] : SORTIE
GÉNÉRÉE LIGNE 2
```

Exemple

```
[2025-11-06 18:12:03] ERROR : [ WORKER 2 ] [ CHECK EXECUTION ] [ ERROR ] [
16049b9c632040b7822b80cb934c36ff ] [ COMMAND NAME ] : sonde_288
[2025-11-06 18:12:03] ERROR : [ WORKER 2 ] [ CHECK EXECUTION ] [ ERROR ] [
16049b9c632040b7822b80cb934c36ff ] [ COMMAND ] : /usr/bin/bash ...
[2025-11-06 18:12:03] ERROR : [ WORKER 2 ] [ CHECK EXECUTION ] [ ERROR ] [
16049b9c632040b7822b80cb934c36ff ] [ STATUS ] : 8 (ERROR)
[2025-11-06 18:12:03] ERROR : [ WORKER 2 ] [ CHECK EXECUTION ] [ ERROR ] [
16049b9c632040b7822b80cb934c36ff ] [ RESULT ] : failed to
[2025-11-06 18:12:03] ERROR : [ WORKER 2 ] [ CHECK EXECUTION ] [ ERROR ] [
16049b9c632040b7822b80cb934c36ff ] [ RESULT ] : execute on this system
```