

# Les logs du Reactionner

## 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 notifications et des événements
  - Pour récupérer les notifications et les événements à exécuter
  - Envoi des résultats de notifications et d'événements au Scheduler
- Surcharge serveur en activité disque, ralentissant l'écriture des logs
- Logs concernant les checks de vérifications Shinken
- Suivi des lancements des commandes par les workers et de leur performances
  - Les Workers arrivent à lancer toutes les commandes ou se limitent
  - Il manque des ressources CPU / RAM
  - Les ressources CPU / RAM sont suffisantes
  - PROBLEME: Un worker n'arrive pas à lancer ses commandes
- Performances du Reactionner
- 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 Reactionner
  - Performance des boucles des Workers
  - Un Worker est lent
  - Un Worker est en retard
- Logs d'exécution des commandes de notification ou des gestionnaires d'événements
  - Gestion d'un logger
    - Activation d'un logger
      - Log d'activation d'un logger
    - Désactivation d'un logger
      - Log de désactivation d'un logger
    - État d'un logger
  - Logs d'exécution
    - Exécution normale
    - Exécution en erreur

Les fichiers de log du Reactionner 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 Reactionner indique l'état des loggers optionnels d'exécution :

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

Ces loggers peuvent être activés pour suivre l'exécution des commandes de notification et des gestionnaires d'événement ( voir la section [Logs d'exécution des commandes de notification ou des gestionnaires d'événements](#) ).



Si les commandes de notification ou les gestionnaires d'événement ne peuvent pas être exécutées normalement, ces loggers génèrent systématiquement des logs en erreur.

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

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-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 : [ reactionner-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 Reactionner reçoit sa configuration pour la première fois deux logs **INFO** sont affichés.

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


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

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

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-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 reactionner n'est pas de la même version que l'arbiter et que l'option `*mismatch_version_error*` est activé sur l'arbiter:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ reactionner-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 reactionner n'est pas de la même version que l'arbiter et que l'option `*mismatch_version_error*` est désactivé sur l'arbiter:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ reactionner-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


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

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

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

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

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-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 reactionner n'est pas de la même version que l'arbiter et que l'option `*mismatch_version_error*` est activé sur l'arbiter:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ reactionner-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 reactionner n'est pas de la même version que l'arbiter et que l'option \*mismatch\_version\_error\* est désactivé sur l'arbiter:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ reactionner-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 que l'Arbiter détecte un changement de lien entre les démons quatre logs en **INFO** seront affichés.

- Les deux premiers logs affichent le(les) lien(s) du(des) démon(s) supprimé(s).

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

- Les deux premiers logs affichent le(les) lien(s) du(des) démon(s) ajouté(s).

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

## Récupération des notifications et des événements

### Pour récupérer les notifications et les événements à exécuter

- **Reactionner actif**
  - Le Reactionner va demander au Scheduler.
  - Le Reactionner indique un temps de travail disponible ( en temps cpu ).
  - Le Scheduler lui donne des notifications et événements ( suivant le temps d'exécution moyen de ces actions constatées sur ce Reactionner ). Il lui donne pour un temps inférieur ou égal au temps de travail demandé.
  - Un log permet d'avoir ce nombre de notifications et le nombre d'événements récupérés. Ce log s'affiche même si aucun événement ou notification n'a été récupéré :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-name ] [ ACTIONS ] [ scheduler-master ] [ GET ] Requesting
actions todo from this scheduler for 2.000s cpu time [received=11 notification(s) / 0 event(s) for 1.087s
cpu time]
```

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

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-name ] [ ACTIONS ] [ scheduler-master ] [ RECEIVED ] We
received actions todo from this scheduler for 1.000s cpu time [received=2 notification(s) / 0 event(s) for
0.160s cpu time]
```

### Envoi des résultats de notifications et d'événements au Scheduler

- **Reactionner actif**

- Une fois les notifications et événements exécutés, le Reactionner envoie les résultats au Scheduler.
- Un log permet d'avoir le nombre de résultats de notifications et d'événements envoyés au Scheduler et le temps mis pour être envoyé.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-name ] [ ACTIONS RESULTS ] [scheduler-master] [ PUSHED ] 1
action's result(s) sends to this scheduler in [0.043]s [1 notifications / 0 events]
```

- **Reactionner passif**

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

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-name ] [ ACTIONS RESULTS ] [scheduler-master] [ GIVEN ] 1
action's result(s) given to answer scheduler request [1 notifications / 0 events]
```

## 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 dans les logs :

```
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 writes time 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 : [ reactionner-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 : [ reactionner-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 : [ reactionner-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 : [ reactionner-name ] [ STATS          ] Daemon stats were computed in 0.020s
(0.001 for daemon common part, 0.020 for reactionner
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 : [ reactionner-name ] [ STATS          ] Daemon stats were computed in 0.004s
(0.000 for daemon common part, 0.004 for reactionner 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 : [ reactionner-name ] [ STATS          ] Clean checks in timeouts structure in
0.000s (before clean: 0 commands in timeouts, after clean: 0)
```

## Suivi des lancements des commandes par les workers et de leur performances

### Les Workers arrivent à lancer toutes les commandes 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   : [ reactionner-name ] [ REACTIONNER STATISTICS ] 3 notifications/event
handlers to run
[YYYY-MM-DD HH:MM:SS] INFO   : [ reactionner-name ] [ REACTIONNER STATISTICS ] => 3 are waiting in workers
[ WORKER 1: 2 ] [ WORKER 2: 1 ] ( Estimated CPU time : 0.117000s )
[YYYY-MM-DD HH:MM:SS] WARNING: [ reactionner-name ] [ REACTIONNER STATISTICS ]      --> Launched 2/5
notifications/event handlers in this loop [ WORKER 1: 1/3 ] [ WORKER 2: 1/2 ]
[YYYY-MM-DD HH:MM:SS] WARNING: [ reactionner-name ] [ REACTIONNER STATISTICS ]      --> Wait time for system
resources availability [ WORKER 1: 0.000s/1.358s ] [ WORKER 2: 0.000s/1.371s ]
[YYYY-MM-DD HH:MM:SS] INFO   : [ reactionner-name ] [ REACTIONNER STATISTICS ] => 0 still to be dispatched
to workers ( Estimated CPU time: 0.000000s )
[YYYY-MM-DD HH:MM:SS] INFO   : [ reactionner-name ] [ REACTIONNER STATISTICS ]      --> Sent this turn [
WORKER 1: 0 ] [ WORKER 2: 0 ]
[YYYY-MM-DD HH:MM:SS] WARNING: [ reactionner-name ] [ SYSTEM LIMITS ] [ MISSING RESOURCE ] CPU limit REACHED
[ CPU %use: 93.9% 90.6% 97.4% 83.3% ( limit: 80% ) ]
[YYYY-MM-DD HH:MM:SS] WARNING: [ reactionner-name ] [ SYSTEM LIMITS ] [ MISSING RESOURCE ] CPU running queue
limit REACHED 32 / 16
[YYYY-MM-DD HH:MM:SS] INFO   : [ reactionner-name ] [ SYSTEM LIMITS ]                RAM limit OK
77.0% / 95.0%
[YYYY-MM-DD HH:MM:SS] INFO   : [ reactionner-name ] [ WORKER LIMITS ]                WORKER running
process nb limit OK [ WORKER 1: 1/256 ] [ WORKER 2: 1/256 ]
```

### Les ressources CPU / RAM sont suffisantes

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

```

[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ REACTIONNER STATISTICS ] 5 notifications/event
handlers to run
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ REACTIONNER STATISTICS ] => 0 are waiting in workers
[ WORKER 1: 0 ] [ WORKER 2: 0 ] ( Estimated CPU time : 0.000000s )
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ REACTIONNER STATISTICS ] --> Launched 0/0
notifications/event handlers in this loop [ WORKER 1: 0/0 ] [ WORKER 2: 0/0 ]
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ REACTIONNER STATISTICS ] --> Wait time for system
resources availability [ WORKER 1: 0.000s/1.002s ] [ WORKER 2: 0.000s/1.006s ]
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ REACTIONNER STATISTICS ] => 5 still to be dispatched
to workers ( Estimated CPU time: 0.195000s )
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ REACTIONNER STATISTICS ] --> Sent this turn [
WORKER 1: 0 ] [ WORKER 2: 0 ]
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ SYSTEM LIMITS ] CPU limit OK [
CPU %use: 83.1% 80.8% 97.5% 78.1% ( limit: 80% ) ]
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ SYSTEM LIMITS ] CPU running queue
limit OK 15 / 16
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ SYSTEM LIMITS ] RAM limit OK
74.0% / 95.0%
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-name ] [ WORKER LIMITS ] WORKER running
process nb limit OK [ WORKER 1: 0/256 ] [ WORKER 2: 0/256 ]

```

## PROBLEME: Un worker n'arrive pas à lancer ses commandes

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: [ reactionner-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 Reactionner n'envoie plus de commande à traiter au worker,
- YY : est le nombre de commandes à lancer en attente sur le worker.

## Performances du Reactionner

On peut suivre le fonctionnement du Reactionner via le log suivant :

```

[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-master ] [ REACTIONNER STATISTICS ] [ PERFS ] [ CURRENT LOOP ]
Waiting to be pushed to workers 0 notifications/event handlers, Estimated CPU Time: 0.00s
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-master ] [ REACTIONNER STATISTICS ] [ PERFS ] [ CURRENT LOOP ]
Waiting to be returned to Schedulers/Synchronizer 13 notifications/event handlers, Consumed CPU Time: 0.29s
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-master ] [ REACTIONNER STATISTICS ] [ PERFS ] [ CURRENT LOOP ]
[ WORKERS ] In workers [ WORKER 1: 1 ( Estimated CPU time: 0.04s ) + in queues ( from daemon:0, to daemon:5
) ] [ WORKER 2: 13 ( Estimated CPU time: 0.49s ) + in queues ( from daemon:0, to daemon:4 ) ]
[YYYY-MM-DD HH:MM:SS] INFO      : [ reactionner-master ] [ REACTIONNER STATISTICS ] [ PERFS ] [ CURRENT LOOP ]
[ WORKERS ] [ Launched = 16 notifications/event handlers, Estimated CPU Time: 0.61s ] [ Executing = 5
notifications/event handlers, Estimated CPU Time: 0.19s ] [ Finished = 14 notifications/event handlers,
Consumed CPU Time: 0.65s, +0.26s 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 Reactionner,,
- 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 : [ reactionner-master ] [ REACTIONNER STATISTICS ] [ PERFS ] [ lmin AVG ]
Waiting to be pushed to workers 0.80 notifications/event handlers/s, Estimated CPU Time: 0.03s
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-master ] [ REACTIONNER STATISTICS ] [ PERFS ] [ lmin AVG ]
Waiting to be returned ] 0.56 notifications/event handlers/s, Consumed CPU Time: 0.01s
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-master ] [ REACTIONNER STATISTICS ] [ PERFS ] [ lmin AVG ] [
WORKERS ] [ Launched = 0.83 notifications/event handlers/s, Estimated CPU Time: 0.03s ] [ Executing = 0.36
notifications/event handlers/s, Estimated CPU Time: 0.013s ] [ Finished = 0.68 notifications/event handlers
/s, Consumed CPU Time (total): 0.03s, +0.01s 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 Reactionner.

## Logs de chargement des modules

Les démons ont une phase de chargement des modules qui est décrite dans la page [Reactionner - GLOBAL - 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 : [ reactionner-name ] [ REACTIONNER 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 : [ reactionner-name ] [WORKER XX][COMMAND TO PROCESS] Worker load_todo_actions :
X.XXX nb_action : XX
```

## Log de performance de la boucle du Reactionner

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

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

## Performance des boucles des Workers

Le log suivant permet de suivre l'activité de chaque Worker et de s'assurer que chacun continue de tourner

```
[YYYY-MM-DD HH:MM:SS] INFO : [ reactionner-name ] [ WORKERS ] Last activity [ WORKER X: X.XXXs ago, WORKER
Y: Y.YYYs ago]
```

### 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 : [ reactionner-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 : [ reactionner-name ] [ WORKER X ] is late, last tick was X.XXXs ago, over limit of X.XXXs
```

## Logs d'exécution des commandes de notification ou des gestionnaires d'événements

### Gestion d'un logger

L'activation ou la désactivation des loggers optionnels se fait via l'utilisation d'une commande **curl** depuis le shell.

Les paramètres requis sont :

- REACTIONNER\_IP : l'adresse IP ou le nom du serveur où tourne le Reactionner ( *ou localhost si le shell est lancé sur le serveur du Reactionner directement* ),
- REACTIONNER\_PORT : le port du Reactionner,
- REACTIONNER\_PROTOCOLE :
  - **https** si le protocole HTTPS a été activé sur le démon ( *voir le paramètre use\_ssl dans le fichier /etc/shinken/daemons/reactionnerd.ini du démon* ),
  - **http** sinon,
- LOGGER\_ID : l'identifiant du logger, soit :
  - **NOTIFICATION\_EXECUTION** pour les commandes de notification,
  - **EVENTHANDLER\_EXECUTION** pour les gestionnaires d'événement.

Le résultat de cette commande renvoie un document JSON.

### Activation d'un logger



Suivant l'activité du Reactionner, l'activation de ces loggers peut générer un gros volume de données.

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

#### Commande

```
curl -s -k "REACTIONNER_PROTOCOLE://REACTIONNER_IP:REACTIONNER_PORT/set_log_enable?logger_id=LOGGER_ID&enable=1"
```

#### Exemple pour activer le logger d'exécution des commandes de notification

```
curl -s -k "http://localhost:7769/set_log_enable?logger_id=NOTIFICATION_EXECUTION&enable=1"
```

La commande doit renvoyer une sortie du style :

#### Résultat

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

#### Exemple pour activer le logger d'exécution des commandes de notification

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

## Log d'activation d'un logger

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

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

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

## Désactivation d'un logger

### Commande

```
curl -s -k "REACTIONNER_PROTOCOLE://REACTIONNER_IP:REACTIONNER_PORT/set_log_enable?  
logger_id=LOGGER_ID&enable=0"
```

### Exemple pour désactiver le logger d'exécution des commandes de notification

```
curl -s -k "http://localhost:7769/set_log_enable?logger_id=NOTIFICATION_EXECUTION&enable=0"
```

La commande doit renvoyer une sortie du style :

### Résultat

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

### Exemple pour désactiver le logger d'exécution des commandes de notification

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

## Log de désactivation d'un logger

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

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

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

## État d'un logger

### Commande

```
curl -s -k "REACTIONNER_PROTOCOLE://REACTIONNER_IP:REACTIONNER_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:7769/get_log_info?logger_id=NOTIFICATION_EXECUTION"
```

La commande doit renvoyer une sortie du style :

#### Résultat

```
[[{"id": "NOTIFICATION_EXECUTION", "name": "[Shinken] [XXXXX EXECUTION CONTROL]", "enable": BOOLEEN}]
```

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

```
[[{"id": "NOTIFICATION_EXECUTION", "name": "[Shinken] [NOTIFICATION EXECUTION CONTROL]", "enable": false}]
```

## Logs d'exécution

### Exécution normale

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

```
[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 16:52:09] INFO : [ WORKER 3 ] [ EVENTHANDLER EXECUTION ] [ DONE ] [  
b526a2d704f84581a85967a02b37d9c0 ] [ COMMAND NAME ] : notify-host-by-email  
[2025-11-06 16:52:09] INFO : [ WORKER 3 ] [ EVENTHANDLER EXECUTION ] [ DONE ] [  
b526a2d704f84581a85967a02b37d9c0 ] [ COMMAND ] : /var/lib/shinken/libexec/notifications/notify_by_email.py  
...  
[2025-11-06 16:52:09] INFO : [ WORKER 3 ] [ EVENTHANDLER EXECUTION ] [ DONE ] [  
b526a2d704f84581a85967a02b37d9c0 ] [ STATUS ] : 0 (OK)  
[2025-11-06 16:52:09] INFO : [ WORKER 3 ] [ EVENTHANDLER EXECUTION ] [ DONE ] [  
b526a2d704f84581a85967a02b37d9c0 ] [ RESULT ] : 2025-11-06 16:52:08,858:INFO: Mail sent successfully
```

### Exécution en erreur



Ces logs sont générés dans le fichier de log du Reactionner **indépendamment de l'activation** ou non du logger.

```
[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 16:14:55] ERROR : [ WORKER 1 ] [ NOTIFICATION EXECUTION ] [ ERROR ] [
af986862b07048af87a9513241182613 ] [ COMMAND NAME ] : USS-NAUTILUS jNotifier command check
[2025-11-06 16:14:55] ERROR : [ WORKER 1 ] [ NOTIFICATION EXECUTION ] [ ERROR ] [
af986862b07048af87a9513241182613 ] [ COMMAND ] : /usr/bin/curl ...
[2025-11-06 16:14:55] ERROR : [ WORKER 1 ] [ NOTIFICATION EXECUTION ] [ ERROR ] [
af986862b07048af87a9513241182613 ] [ STATUS ] : 6 (ERROR)
[2025-11-06 16:14:55] ERROR : [ WORKER 1 ] [ NOTIFICATION EXECUTION ] [ ERROR ] [
af986862b07048af87a9513241182613 ] [ RESULT ] : % Total % Received % Xferd Average Speed Time
Time Time Current
[2025-11-06 16:14:55] ERROR : [ WORKER 1 ] [ NOTIFICATION EXECUTION ] [ ERROR ] [
af986862b07048af87a9513241182613 ] [ RESULT ] : Dload Upload Total
Spent Left Speed
[2025-11-06 16:14:55] ERROR : [ WORKER 1 ] [ NOTIFICATION EXECUTION ] [ ERROR ] [
af986862b07048af87a9513241182613 ] [ RESULT ] :
[2025-11-06 16:14:55] ERROR : [ WORKER 1 ] [ NOTIFICATION EXECUTION ] [ ERROR ] [
af986862b07048af87a9513241182613 ] [ RESULT ] : 0 0 0 0 0 0 0 0 0 ---:---:---:
-- ---:---: 0curl: (6) Could not resolve host: UNSET
```