

Options de configuration

En créant ou éditant votre fichier de configuration ,gardez à l'esprit que :

- les lignes commençant par le caractère "#" sont considérés comme des commentaires
- les noms de variable sont sensibles aux majuscules/minuscules

Le fichier principal de configuration est "shinken.cfg". Il est situé dans le répertoire **"/etc/shinken/"** .

Ci dessous, la liste des paramètres utilisés. Pour les autres paramètres (non réglés par défaut) voir [Configuration avancée](#).

Options utilisées par défaut

Cfg dir et Cfg files

Format :

```
cfg_dir=<directory_name>  
cfg_file=<file_name>
```

Ceux-ci sont des assertions, et non des paramètres . L'arbitre les considère comme des ordres pour ouvrir d'autre(s) fichier(s) de configuration. Pour le `cfg_dir` one, l'arbitre lit uniquement les fichiers se terminant par ".cfg". L'arbitre lit les directives de manière récursive pour les fichiers, mais ne considère plus les lignes à l'intérieur comme assertions.

Cela signifie que les fichiers `cfg_dir` ou `cfg_file` sont considérés comme paramètres en dehors de `Shinken.cfg`. L'arbitre gère le fichier principal de configuration différemment des autres fichiers.

Automatic State Retention Update Interval

Format:

```
retention_update_interval=<minutes>
```

Défaut:

```
retention_update_interval=60
```

Ce paramètre détermine l'intervalle (en minutes) entre chaque backup de rétention automatique de données par le scheduler en opération normale. Si vous mettez cette valeur à 0, il ne sauvegardera pas les données de rétention à intervalle régulier, mais seulement à l'arrêt et au redémarrage.

Maximum Host/Service Check Spread

Format:

```
max_service_check_spread=<minutes>  
max_host_check_spread=<minutes>
```

Défaut:

```
max_service_check_spread=30  
max_host_check_spread=30
```

Cette option détermine l'écart de temps maximum (en minutes) pour vérifier tous les hôtes/checks (qui sont planifiés pour être vérifiés régulièrement). Cette option garantira que les vérifications initiales se feront dans la période de temps définie. Par défaut, la valeur est 30 (minutes).

Service/Host Check Timeout

Format:

```
service_check_timeout=<seconds>  
host_check_timeout=<seconds>
```

Défaut:

service_check_timeout=60
host_check_timeout=30

Il s'agit du nombre maximum de secondes qu'autorise Shinken Enterprise pour lancer les hôtes/checks. Si les vérifications dépassent cette limite, elles sont supprimées et un statut CRITICAL est retourné. Une erreur timeout sera également logguée.

Il y a souvent une confusion répandue, sur ce que cette option fait vraiment. Elle est destinée à être utilisée comme un mécanisme de dernier rempart, pour tuer les plugins qui se comportent mal . Elle doit être réglée sur une valeur haute (quelque chose comme 60 secondes ou plus) , de sorte que chaque vérification soit dans ce délai . Si une vérification se fait sur une longueur supérieure à cette limite , Shinken Enterprise va la tuer , pensant que c'est un processus d'emballage.

Timeout Exit Status

Format:

timeout_exit_status={0,1,2,3}

Défaut:

timeout_exit_status=2

Etat défini par Shinken Enterprise en cas de timeout.

Flap History

Format:

flap_history=<int>

Défaut:

flap_history=20

Cette option est utilisée pour définir la taille de l'historique gardé par le scheduler pour réaliser le calcul de l'état flapping .Par défaut, la valeur de 20 états gardés.

La taille mémoire est pour le démon scheduler : 4Bytes flap_history (nb hosts + nb checks). Pour un environnement important, il faut 4 20 (1000+10000) - 900Ko. .

Max Plugins Output Length

Format:

max_plugins_output_length=<int>

Défaut:

max_plugins_output_length=8192

Cette option est utilisée pour définir la taille max en bytes pour les retours de plugins de checks .Si vous voyez des valeurs tronquées, comme par exemple pour des gros checks de disque, augmentez la valeur.

Enable problem/impacts states change

Format:

enable_problem_impacts_states_change=<0/1>

Défaut:

enable_problem_impacts_states_change=0

Cette option est utilisée si on applique le changement d'état lorsqu'un hôte ou un check est impacté par un problème cause. L'état sera changé à UNKNOWN pour un check et UNREACHABLE pour un hôte jusqu'au lancement du prochain check. Le changement d'état ne compte pas comme une tentative, c'est juste pour la console afin que l'utilisateur sache que ces objets ont des problèmes et que leurs états précédents n'étaient pas sûrs.

Disable Old Nagios Parameters Whining

Format:

disable_old_nagios_parameters_whining=<0/1>

Défaut:

disable_old_nagios_parameters_whining=0

Si 1, désactive toutes les notifications et messages à la vérification de la configuration

Timezone Option

Format:

use_timezone=<tz from tz database>

Défaut:

use_timezone=""

Cette option permet d'outrepasser la zone de temps définie par défaut pendant laquelle l'instance Shinken Enterprise tourne . C'est très utile si vous avez plusieurs instances qui doivent tourner sur le même serveur, mais avec des plages horaires différentes . Si rien n'est spécifié, Shinken Enterprise utilisera la plage horaire du système.

Environment Macros Option

Format:

enable_environment_macros=<0/1>

Default:

enable_environment_macros=0

Cette option détermine si le démon Shinken Enterprise va rendre disponibles les :ref:`macros <thebasics/macrolist>` comme variables d'environnement pour les checks, notifications, gestionnaires d'événements, etc. Dans des installations importantes, cela peut être problématique car il nécessite de la CPU supplémentaire pour calculer les valeurs des macros et les rendre disponibles . Cela rajoute également de la communication réseau entre les schedulers et les pollers.

- 0 = ne rend pas les macros disponibles en tant que variables d'environnement
- 1 = rendent les macros disponibles comme variables d'environnement

Initial States Logging Option

Format:

log_initial_states=<0/1>

Défaut:

log_initial_states=1

Cette variable détermine si Shinken Enterprise tous les états initiaux des hôtes et des checks à être loggués, même si le résultat est un état OK . Les état initiaux ne sont normalement loggués que lorsqu'il y a un problème sur le 1er check. Activer cette option est utile si vous avez une application qui scanne le fichier log pour définir des statistiques long terme.

- 0 = ne loggue pas les états initiauxDon't log initial states
- 1 = loggue les états initiaux

Event Handler during downtimes

Format:

no_event_handlers_during_downtimes=<0/1>

Default:

no_event_handlers_during_downtimes=1

Cette option détermine si Shinken Enterprise va lancer :ref:`event handlers <advanced/eventhandlers>` lorsque l'hôte ou le check est maintenance planifiée .

- 0 = Lancement de gestionnaire d'événements (comportement Nagios)
- 1 = Ne lance pas le gestionnaire d'événements

Partie démon Arbitre

Les paramètres suivants sont communs à tous les démons .

Workdir

Format:

workdir=<directory>

Défaut :

workdir=/var/run/shinken/

Cette variable spécifie le répertoire de travail du démon .

Dans le cas de l'arbitre, si la valeur est vide ou absente, c'est le répertoire du paramètre lock_file qui sera pris.

Arbiter Lock File

Format:

lock_file=<file_name>

Exemple:

lock_file=/var/lib/shinken/arbiterd.pid

Cette option spécifie l'emplacement du fichier lock que l'arbitre doit créer lorsqu'il tourne en tant que démon (démarré avec l'argument de la ligne de commande "-d"). Ce fichier contient le numéro de l'id du process (PID) de l'arbitre qui tourne.

Local Log

Format:

local_log=<filename>

Défaut:

local_log=/var/log/shinken/arbiterd.log'

Cette variable spécifie le fichier log file pour le démon.

Log Level

Format:

log_level=[DEBUG,INFO,WARNING,ERROR,CRITICAL]

Défaut:

log_level=WARNING

Cette variable spécifie quelles logs seront remontées par l'arbiter . POur les autres démons, cela peut être défini dans leur fichier local \ d.ini files.

Arbiter Daemon User

Défini dan brokerd.ini, brokerd-windows.ini, pollerd.ini, pollerd-windows.ini, reactionnerd.ini, schedulerd.ini and schedulerd-windows.ini.

Format:

shinken_user=username

Défaut:

shinken_user=<current user>

C'est utilisé pour définir le user actif. Le process de l'arbiter (process principal) va tourner en tant que user actif. Après le démarrage initial du programme, Shinken Enterprise va laisser tomber ses privilèges et tournera en tant que ce user actif.

Arbiter Daemon user Group

Format:

shinken_group=groupname

Défaut:

shinken_group=<current group>

C'est utilisé pour définir le bon groupe utilisateur pour lancer l'arbiter.

Modules directory

Format:

modules_dir=<dirname>

Défaut:

`modules_dir=/var/lib/shinken/modules`

Chemin vers le répertoire modules

Daemon Enabled

Format:

`daemon_enabled=[0/1]`

Défaut:

`daemon_enabled=1`

Validé à 0 si vous souhaitez que ce démon (arbitrer) NE FONCTIONNE PAS

Use SSL

Format:

`use_ssl=[0/1]`

Défaut:

`use_ssl=0`

Utiliser SSI ou pas. Vous devez l'activer dans les autres démons également .

Ca Cert

Format:

`ca_cert=<filename>`

Défaut:

`ca_cert=etc/certs/ca.pem`

Certificat de Certification Authority (CA) certificate.

Server Cert

Format:

`server_cert=<filename>`

Défaut:

`server_cert=etc/certs/server.cert`

Certificat serveur pour SSL. Chemin complet mis pour certs

Server Key

Format:

server_key=<filename>

Défaut:

server_key=/etc/certs/server.key

Clé serveur pour SSL. Chemin complet mis pour certs

Hard SSL Name Check

Format:

hard_ssl_name_check=[0/1]

Défaut:

hard_ssl_name_check=0

Active le nom du check SSL .

HTTP Backend

Format:

http_backend=[auto, cherrypy, swsgiref]

Défaut:

http_backend=cherrypy

Specifie quel http_backend utiliser.