

# Fichier de configuration (shinken.cfg)

## Options de configuration

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

- les lignes commençant par le caractère "#" sont considérées 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`, 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.

### Intervalle de sauvegarde automatique de la rétention

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.

### Propagation maximum pour les hôtes et services

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).

### Timeout pour les Checks sur les Services/Hôtes

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 dure plus longtemps , Shinken Enterprise va la tuer, estimant que le process est bloqué.

## Status de sortie du Timeout

Format:

```
timeout_exit_status={0,1,2,3}
```

Défaut:

```
timeout_exit_status=2
```

Valeur de retour définie par Shinken Enterprise en cas de timeout. **0** pour "OK", **1** pour "UNKNOW", **2** pour "WARNING", **3** pour "CRITICAL"

## Historique de changement d'état

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 est 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.

## Longueur maximum de la sortie des plugins

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.

## Activation du changement d'état pour les problèmes/impacts

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.

## Désactivation de la notification des anciens paramètres nagios

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.

## Options de fuseau horaire

Format:

**`use_timezone=<tz from tz database>`**

Défaut:

**`use_timezone=""`**

Cette option permet d'outrepasser le fuseau horaire défini par défaut sous lequel l'instance Shinken Enterprise tourne. C'est très utile si vous avez plusieurs instances qui doivent tourner sur le même serveur, mais avec des fuseaux horaires différentes. Si rien n'est spécifié, Shinken Enterprise utilisera le fuseau horaire du système.

## Macros d'environnement

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 augmente également 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

## Log des états initiaux

Format:

**log\_initial\_states=<0/1>**

Défaut:

**log\_initial\_states=1**

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

- 0 = ne loggue pas les états initiaux
- 1 = loggue les états initiaux

## Gestionnaire d'événements pendant les périodes de maintenance

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 en maintenance planifiée.

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

## Affichage des logs au format date

Format:

**human\_timestamp\_log=<0/1>**

Défaut :

**human\_timestamp\_log=1**

Cette option permet d'afficher les logs avec un format de date différent:

- 1 = lisible par un humain
- 0 = format timestamp

## Partie démon Arbitre

Les paramètres suivants sont communs à tous les démons.

## Répertoire de travail

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.

## Fichier de verrouillage

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 PID de l'arbitre qui tourne.

## Fichier de log local

Format:

***local\_log=<filename>***

Défaut:

***local\_log=/var/log/shinken/arbiterd.log'***

Cette variable spécifie l'emplacement du fichier log file pour le démon.

## Niveau de log

Format:

***log\_level=[DEBUG,INFO,WARNING,ERROR,CRITICAL]***

Défaut:

***log\_level=WARNING***

Cette variable spécifie le niveau de logs remonté par l'arbitre. Pour les autres démons, cela peut être défini dans leur fichier `/etc/shinken/daemons/*.ini` local

## Utilisateur du daemon

Défini dans `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>***

User sous lequel le process de l'arbitre (process principal) va tourner.

Après le démarrage initial du programme, Shinken Enterprise va réduire ses privilèges et tournera en tant que cet user.

## Groupe d'utilisateur du daemon

Format:

***shinken\_group=groupname***

Défaut:

***shinken\_group=<current group>***

Groupe sous lequel le process de l'arbitre (process principal) va tourner.

## Répertoire des modules

Format:

***modules\_dir=<dirname>***

Défaut:

***modules\_dir=/var/lib/shinken/modules***

Chemin vers le répertoire stockant les modules

## Daemon activé

Format:

***daemon\_enabled=[0/1]***

Défaut:

***daemon\_enabled=1***

Si 0, le démon (arbitre) NE FONCTIONNE PAS

## Utilisation du SSL

Format:

***use\_ssl=[0/1]***

Défaut:

***use\_ssl=0***

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

## Fichier de l'autorité de certification

Format:

***ca\_cert=<filename>***

Défaut:

***ca\_cert=/etc/certs/ca.pem***

Certificat de la Certification Authority (CA) avec le chemin complet.

## Certificat serveur

Format:

***server\_cert=<filename>***

Défaut:

***server\_cert=/etc/certs/server.cert***

Certificat serveur pour SSL avec le chemin complet.

## Clé du certificat

Format:

***server\_key=<filename>***

Défaut:

***server\_key=/etc/certs/server.key***

Clé serveur pour SSL avec le chemin complet

## Vérification stricte de la correspondance de nom

Format:

***hard\_ssl\_name\_check=[0/1]***

Défaut:

***hard\_ssl\_name\_check=0***

Si à 1, active la vérification stricte de la correspondance entre le nom du serveur et le nom défini dans le certificat.

## Moteur HTTP

Format:

***http\_backend=[auto, cherrypy, swsfiref]***

Défaut:

***http\_backend=cherrypy***

Spécifie quel moteur HTTP utiliser.