

Fichier de configuration (shinken.cfg)

Sommaire

- Concept
- Configuration
 - Exemple de fichier de configuration
- Options de configuration
- Options utilisées par défaut
 - Cfg dir et Cfg files
 - Intervalle de sauvegarde automatique de la rétention
 - La valeur par défaut du paramètre de condition de suppression automatique de la prise en compte dans l'API et la WebUI
 - Propagation maximum pour les hôtes et services
 - Timeout pour les Checks sur les Services/Hôtes
 - Status de sortie du Timeout
 - Historique de changement d'état
 - Longueur maximum de la sortie des sondes (plugins)
 - Activation du changement d'état pour les problèmes/impacts
 - Désactivation de la notification des anciens paramètres nagios
 - Options de fuseau horaire
 - Macros d'environnement
 - Log des états initiaux
 - Lancement de la commande du gestionnaire d'événements pendant les périodes de maintenance
 - Lancement de la commande du gestionnaire d'événements à chaque changement d'état
 - Affichage des logs au format date
- Partie démon Arbiter
 - Répertoire de travail
 - Fichier de verrouillage
 - Fichier de log local
 - Niveau de log
 - Utilisateur du daemon
 - Groupe d'utilisateur du daemon
 - Répertoire des modules
 - Daemon activé
 - Utilisation du SSL
 - Fichier de l'autorité de certification
 - Certificat serveur
 - Clé du certificat
 - Vérification stricte de la correspondance de nom
 - Moteur HTTP
 - Interface d'écoute
- Options d'export des données
 - Mot de passe

Concept

Cette page explique les paramètres présents dans le fichier **/etc/shinken/shinken.cfg**

Pour apporter des modifications à ces paramètres, il est conseillé de modifier le fichier suivant : **/etc/shinken-user/configuration/daemons/arbiters/larbiter_cfg_overload.cfg**

- Il est fortement conseillé de les surcharger dans le fichier **arbiter_cfg_overload.cfg**.
- Ils sont présents dans ce fichier pour indiquer qu'ils existent, mais leur modification dans le fichier de surcharge permettra une meilleure gestion (*ne pas modifier ce que l'on livre par défaut, mais le surcharger*).

Configuration

Exemple de fichier de configuration

```
#*****
# This file contains values for:
#   - the global parameters
#   - the arbiter parameters
#
# IMPORTANT : DO NOT EDIT THIS FILE. This file will be overwritten at every update
#             with the default values
#
#*****
```

```

=====
#==== logging =====

local_log=/var/log/shinken/arbiterd.log                                # Log file for the arbiter daemon

the schedulers logs                                                    # You can enable the logging of elements states into
log_initial_states=0                                                    # disabled by default

# If you disable, the timestamp will be an epoch integer instead of a human date
human_timestamp_log=1

# Set logging level to info by default for the
Arbiter daemon (info, warning and error logs will be shown)
log_level=INFO

=====

#==== Modules and architecture =====

cfg_dir=modules
cfg_dir=arbiters
cfg_dir=schedulers
cfg_dir=pollers
cfg_dir=reactionners
cfg_dir=brokers
cfg_dir=receivers
cfg_dir=realms
cfg_dir=synchronizers

=====

#==== Default elements (host, checks, ...) properties =====

# Load default properties for types (like hosts).
# Do not edit these files, they are overwritten at
every update, you can overload them                                     # in a directory listed at the end of this file.

cfg_dir=_default/elements_default_properties
#END =====

#==== Global DATA =====

# You will find global DATA into this directory
cfg_dir=resource.d

#END =====

#==== Acknowledge =====

# The default value for the automatic deletion condition of an acknowledge
# ( warning in the Module receiver-module-webservice => it corresponds to the parameter "sticky" )
# Possible values are :
#      on_OK              : the acknowledgement will be deleted if the status becomes "Ok"

```

```

#           on_status_change : the acknowledgement will be deleted if the status changes ( Example : "Warning"
=> "Critical", "Warning" => "Ok", "Critical" => "Warning" )
#           EXCEPTION : the acknowledgement will NOT be deleted if the status becomes "Unknown" ( Example :
"Warning" => "Unknown" => "Warning", "Critical" => "Unknown"=> "Critical", "Warning" => "Unknown",
"Critical" => "Unknown" )
#global__acknowledge__automatic_deletion_condition=on_OK

#END
===== #=====
=====
#===== Monitoring and scheduling =====

of an element the first time
start of the element presence in your
in UNKNOWN state (not checked yet)
by default
5 minutes by default
max_service_check_spread=5
max_host_check_spread=5

clusters and checks) is allowed to run.
the element will be set in CRITICAL state
check_running_timeout=60

allowed to run. after this limit,
set in CRITICAL state

# /\ USELESS host_check_timeout PARAMETER
# The parameter "host_check_timeout" is no more used in Shinken. Use check_running_timeout instead.
#host_check_timeout=60

clusters and checks) is allowed to consume cpu
performance.
warning_threshold_cpu_usage=5

changes we keep looking for
flap_history=20

execution by the pollers
max_plugins_output_length=65536

the state of dependent elements (like checks on a host)
their final state returned by the plugin
enable_problem_impacts_states_change=1

# Number of minutes to randomly spread the scheduling
# a scheduler schedule this element. During the first
# daemons and this spread limit, the element will be
# - max_service_check_spread => for checks, 5 minutes
# - max_host_check_spread => for hosts and clusters,

# How many seconds a command check (for hosts,
# after this limit, the process will be killed and
# by default: 60

# How many seconds a host check (for hosts only) is
# the process will be killed and the element will be
# by default: 60

# How many seconds a command check (for hosts,
# before raising a warning in check scheduler
# by default: 5

# flap_history is the length of history states
# flapping.
# by default: 20

# Max plugin output size allowed for the plugins
# by default: 65536 (64K)

# If enabled, an element that goes wrong will change
# until they are checks again and then they will take

```

```

specific nagios parameters
disable_old_nagios_parameters_whining=1

# If enabled, won't print warning messages about

# If enabled, set a timezone for the daemons

# If you need to set a specific language for daemons
output (supported: fr; anything else is english), uncomment it
#language=fr

# If your plugins need NAGIOS_ environnement
variables enable them for all your commands.
# If only some commands need them, you can enable it
on these specific commands instead.
# Note: enable this will have a huge impact on your
schedulers CPU consumption.
enable_environment_macros=0

# By default don't launch even handlers during
downtime.
# Note: the default Nagios behaviour is to allow
them. If you want to keep this
# Nagios behavior, then set this parameter to 0
no_event_handlers_during_downtimes=1

# Activate this option to allow Event Handler to
execute command on any status change,
# in addition to its normal behavior.
#           0 : Disable (Default)
#           1 : Enable
event_handler__hard_state__trigger_on_any_status_change=0

# Number of minutes between forced scheduler
retention save
# retention: scheduler backup of elements state so a
daemon restart
# can start with previous states and not UNKNOWN
# by default 60 (minutes)
retention_update_interval=60

# Minimal delay before state of elements start to
expire and elements state change to no data (unknown).
# You can change this parameter if some of your
element state expires when you push a new configuration in production.
#minimal_time_before_an_element_become_missing_data=0

# Maximum duration (in seconds) of the initial
daemons check (schedulers, brokers, ...) by the arbiter when start
# do check which is ALIVE or DEAD. If a daemon didn't
respond at least once during this duration, it will
# be declared as DEAD and a spare will take its
configuration at startup
configuration_dispatch__initial_daemons_check__max_duration=30

```

```

#=====

```

```

=====
===== System arbiter daemon parameters (user, group, pid, ...) =====

shinken_user=shinken                                     # Which user/group to run the daemon as.
shinken_group=shinken

lock_file=/var/run/shinken/arbiterd.pid                 # Lock file (with pid) for arbiter daemon

daemon_enabled=1                                         # Enable or not the arbiter to run.
                                                         # default: 1 (enabled)

modules_dir=/var/lib/shinken/modules                    # Path to the modules code directory.

=====
===== Arbiter daemon HTTP(S) listening =====

instead of HTTP protocol.                               # If enabled, the arbiter daemon will listen in HTTPS
only. You need to generate                              # Note: default pem/cert and key files are for sample
                                                         # your own with your PKI.
                                                         # by default: 0 (disabled)

use_ssl=0
ca_cert=/etc/shinken/certs/ca.pem
server_cert=/etc/shinken/certs/server.cert
server_key=/etc/shinken/certs/server.key

certificates name checks.                               # Should the arbiter connections will force the HTTPS
same as the daemon address, then                        # If enabled and a distant certificate is not the
                                                         # the connection will be refused.

hard_ssl_name_check=0

with.                                                    # Which HTTP backend to start the listening daemon
                                                         # Currently only auto is managed.

http_backend=auto

bind_addr=0.0.0.0                                       # Which addr to bind for the arbiter daemon
                                                         # by default: 0.0.0.0 (means all interfaces)

=====
===== Daemon data export =====

data ( like host name, realm name, ... )                # Set a private password to allow dump with sensible
                                                         # (if void or missing, only anonymous dump is
allowed)
#daemon__export_data__password=

=====
===== User parameters customizations =====

# If you need to customise a value in this file, set it in this file.

```

```

cfg_file=/etc/shinken-user/configuration/daemons/arbiters/arbiter_cfg_overload.cfg

                                # For your custom global DATA.
cfg_dir=/etc/shinken-user/configuration/global-data

                                # For the monitoring properties
cfg_dir=/etc/shinken-user/configuration/monitoring_policy
#=====

```

Options de configuration

En créant ou éditant un fichier de configuration, garder à l'esprit que :

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

Le fichier principal de configuration est "shinken.cfg".

- Il est situé dans le répertoire **"/etc/shinken/"**.
- ⚠ L'édition de ce fichier doit être faite avec l'encodage utf-8

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

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'Arbiter les considère comme des ordres pour ouvrir d'autre(s) fichier(s) de configuration. Pour le `cfg_dir`, l'Arbiter lit uniquement les fichiers se terminant par ".cfg". L'Arbiter 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'Arbiter gère le fichier principal de configuration différemment des autres fichiers.



Ces deux options ne permettent plus d'importer des éléments de configuration (*hôtes, Checks, clusters, etc...*)

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 sauvegarde de rétention automatique de données par le Scheduler en opération normale.

Si l'on met cette valeur à 0, il ne sauvegardera pas les données de rétention à intervalles réguliers, mais seulement à l'arrêt et au redémarrage.

La valeur par défaut du paramètre de condition de suppression automatique de la prise en compte dans l'API et la WebUI

Format :

```

global__acknowledge__automatic_deletion_condition=<valeur>

```

Cette option détermine la valeur par défaut du paramètre de condition de suppression automatique de la prise en compte dans l'API et la WebUI.

Ce paramètre, défini sous quelle condition, la prise en compte sera supprimé automatiquement si un changement de statut intervient.

Valeurs possibles :

- on_OK (*valeur par défaut*)

cela signifie que la prise en compte sera automatiquement supprimée seulement si le statut passe à .

- on_status_change

la prise en compte sera automatiquement supprimée pour n'importe quel changement de statut (*Exemple* :  ).



Attention : le statut  est ignoré pour déterminer la suppression (Exemple :   ).

- La logique du statut  est de ne pas savoir quel état est l'élément, donc Shinken considère qu'on ne peut pas perdre une **Prise en compte**, en passant dans ce statut.

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 vérifié 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 :

```
check_running_timeout=<seconds>
```

Défaut :

```
check_running_timeout=60
```

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 loggué.

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 "UNKNOWN", **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 des changements d'états ou non gardés par le Scheduler pour réaliser le calcul de l'état flapping. Par défaut, la valeur est de 20 changements d'états ou non gardés.

- La taille mémoire est pour le démon Scheduler est de : $(\text{nombre d'hôtes} + \text{nombre cluster} + \text{nombre checks}) * ((\text{flap_history} * 240) + 2960)$.
- Par exemple, pour 1000 hôtes + 10000 checks : $(1000 + 10000) * ((20 * 24) + 296) = 8.14\text{Mo}$.

Longueur maximum de la sortie des sondes (plugins)

Format :

max_plugins_output_length=<int>

Défaut :

max_plugins_output_length=65536

Cette option est utilisée pour définir la taille max en bytes pour les retours de plugins de checks.

- Si des résultats de checks sont incomplets ou des métriques manquantes, cela peut être lié à cette taille maximum.
- Augmenter cette limite pourrait être la solution, mais garder à l'esprit que des retours de checks plus grands augmenteront la taille de consommation RAM de certains démons, en fonction de ce que retournent les sondes (*comme le Broker ou le Scheduler*).

Activation du changement d'état pour les problèmes/impacts

Format :

enable_problem_impacts_states_change=<0/1>

Défaut :

enable_problem_impacts_states_change=1

Cette option est utilisée si on applique le changement d'état lorsqu'un hôte ou un check est impacté par un problème causé. 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 lorsque plusieurs instances doivent tourner sur le même serveur, mais avec des fuseaux horaires différents. Si rien n'est spécifié, Shinken Enterprise utilisera le fuseau horaire du système.

Macros d'environnement

Format :

enable_environment_macros=<0/1>

Défaut :

`enable_environment_macros=0`

Cette option détermine si le démon Shinken Enterprise va rendre disponibles les variables comme variables d'environnement pour les checks, notifications, gestionnaires d'événements, etc. Dans des installations importantes, cela peut être problématique, car il nécessite du CPU supplémentaire pour calculer les valeurs des macros et les rendre disponibles. Cela augmente également la communication réseau entre les Schedulers et les Pollers. (Voir la page [Les Variables \(Remplacement dynamique de contenu - Anciennement les Macros \)](#))

- 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 dans le cas de l'utilisation d'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

Lancement de la commande du gestionnaire d'événements pendant les périodes de maintenance

Format :

`no_event_handlers_during_downtimes=<0/1>`

Défaut :

`no_event_handlers_during_downtimes=1`

Cette option détermine si Shinken Enterprise va lancer la commande du gestionnaire d'événements lorsque l'hôte ou le check est en maintenance planifiée. (Voir la page [Gestionnaire d'événements](#))

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

Lancement de la commande du gestionnaire d'événements à chaque changement d'état

Format :

`event_handler_hard_state_trigger_on_any_status_change=<0/1>`

Défaut :

`event_handler_hard_state_trigger_on_any_status_change=0`

Autorise le gestionnaire d'événements à exécuter sa commande lors du changements de statut des hôtes, clusters et checks quand le statut est confirmé ("**HARD**"), en plus de son fonctionnement normal.

(Voir la page [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 Arbiter

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'Arbiter, 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'Arbiter 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'Arbiter 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'Arbiter. 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'Arbiter (*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'Arbiter (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 (*Arbiter*) NE FONCTIONNE PAS

Utilisation du SSL

Format :

use_ssl=[0/1]

Défaut :

use_ssl=0

Utiliser SSL ou pas. Il faudra 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]

Défaut :

http_backend=auto

Spécifie quel moteur HTTP utiliser.
Actuellement, seule auto est gérée.

Interface d'écoute

Format :

bind_addr=[Adresse IP]

Défaut :

bind_addr=0.0.0.0

Définis l'interface qui sera en écoute sur le réseau. La valeur 0.0.0.0 (par défaut) permet d'écouter sur toutes les interfaces.

Options d'export des données

Mot de passe

Format :

daemon__export_data__password=<chaîne de caractère>

Défaut :

daemon__export_data__password=

Permet de définir un mot de passe pour les appels d'extraction des données des démons.

Par défaut, il n'y a pas de mot de passe, ce qui interdit l'export de donnée avec des données sensibles (*comme les noms d'élément, le nom des royaumes, ...*)

- Voir la page [shinken-scheduler-export-data - export des données du Scheduler](#)

Attention : le caractère " ; " est interprété comme le début d'un commentaire donc le mot de passe sera mal interprété s'il contient un " ; ".