

Poller - GLOBAL - Les logs de gestion des modules - chapitre [MODULES-MANAGER]

Sommaire

- Import des modules depuis /var/lib/shinken/modules - chapitre [CODE-LOADING]
 - Chargement du code
 - Chargement du module en tant que package Python
 - Vérification des propriétés du module
- Création du module - chapitre [CREATION]
 - Création de l'instance du module
 - Démarrage du module
 - Démarrage des modules avec des workers
- Changement de configuration ou d'état du module - chapitre [UPDATE]
 - Rajout d'un nouveau module
 - Suppression d'un module
 - Changement de configuration d'un module (et sa relance)
 - Le module n'a pas réussi à se mettre à jour
- Arrêt du module - chapitre [SHUTDOWN]
 - Le démon éteint ses modules
 - Le module s'éteint
 - Le démon a fini d'éteindre ses modules
- Arrêt inopiné du module - chapitre [CRASH]
 - Le module s'est arrêté de façon inattendu
 - Le module a un comportement anormal

Import des modules depuis /var/lib/shinken/modules - chapitre [CODE-LOADING]

Chaque démon (*ou module qui possède d'autres modules*) va démarrer son gestionnaire de module "modules-manager" qui va charger chaque module présent dans **/var/lib/shinken/modules**.

Pour qu'un module soit chargé, il faut qu'il soit présent dans un répertoire.

Chargement du code

Si le "modules-manager" rencontre un problème durant le chargement du module, un log en **ERROR** avec l'exception Python sera affiché et le module ne sera pas importé.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=name_directory ] Import module [module_name] failed: EXCEPTION.
```

Chargement du module en tant que package Python

Lors du chargement du code, le "modules-manager" va vérifier si un module peut être importé en tant que package Python. En cas d'échec un log en **INFO** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=module_name ] Failed to import the directory /var/lib/shinken/modules/module_name as a python module.
```

Si cet échec est dû à un problème d'import de fichier en Python, le module ne sera pas chargé et 3 logs en **ERROR** seront affichés :

- Le premier log indique que l'import a échoué et que le module ne sera pas chargé.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CODE-LOADING ] [ directory=name_directory ] Failed to import the directory /var/lib/shinken/modules/directory_name as a python module. The module won't be loaded.
```

- Le second log indique que le fichier '`__init__.py`' dans le dossier du module est manquant (sans ce fichier, il est impossible d'importer le module en tant que module Python).

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=name_directory ] - because of missing file : /var/lib/shinken/modules/module_name/__init__.
py.
```

- Le troisième log précise quel fichier n'a pas réussi à être importé dans le code du module. Dans notre exemple, c'est le fichier `module_name.my_file` qui n'a pas été importé.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=name_directory ] - The line 'import module_name.my_file' The module won't work in your
module.py file.
```

Vérification des propriétés du module

Une fois que le "modules-manager" a chargé un module en tant que package Python, il va vérifier si le dictionnaire "properties" est existant dans le module.

Dans le cas contraire, le module ne sera pas importé et le log en **ERROR** suivant sera affiché :

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=name_directory ] Missing properties dict in your module file /var/lib/shinken/modules/module_name
/module.py. The module won't be loaded.
```

Création du module - chapitre [CREATION]

Une fois, le code python chargé, on demande au "modules-manager" de créer les modules, de les démarrer et de démarrer son ou ses Worker(s).

Création de l'instance du module

Le "modules-manager" va essayer de créer le module :

- Si le code du module ne contient pas la fonction "get_instance()" ou qu'elle ne retourne rien, un log en **ERROR** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [
module_name] [ module-type=module_type ] The module get_instance() call did not return any instance
or does not exist.
```

- Si le "get_instance()" du module rencontre une erreur, le log en **ERROR** sera affiché avec l'exception Python concernée:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [
module_name ] [ module-type=module_type ] [0.046s] The module creation did fail with an exception:
EXCEPTION. Will try to recreate it in the future.
```

Si le module a bien été créé alors un log en **INFO** indiquera son temps de création.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [
module-type=module_type ] [0.046s] The module is created.
```

Démarrage du module

Une fois le module créé, le "modules-manager" va essayer de le démarrer.

Ce log sera en **INFO**.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [
module-type=module_type ] Trying to init module.
```

Si le "modules-manager" a réussi à créer son module, mais n'a pas réussi à le démarrer alors 3 logs apparaîtront :

1. Un log en **ERROR** avec l'exception Python rencontrée :

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] The module failed to init and he raised an exception: EXCEPTION.
```

2. Un log en **WARNING** indiquant le nombre de fois que le "modules-manager" à essayé de démarrer son module sans succès. Le compteur se remet à zéro dès lors que le démon a réussi à démarrer son module.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] The module tried to start: 5 times.
```

3. Un log en **WARNING** indiquant quand le "modules-manager" va essayer de redémarrer son module (*maximum une minute*).

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] The module will try to restart at 16:32:18.
```

Un log en **INFO** apparaîtra si le "modules-manager a réussi à démarrer son module.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] The module is started.
```

Démarrage des modules avec des workers

Une fois le module démarré, s'il utilise des Workers, le "modules-manager" va essayer de les démarrer.

- S'il n'arrive pas à les démarrer, le module redémarrera dans le futur (*max 1 minute*) afin de redémarrer les Workers.
- Ce log sera en **INFO**.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] Starting a worker(s).
```

Si le "modules-manager" n'a pas réussi à démarrer le module :

1. Et qu'il a levé une exception lors du démarrage, alors un log en **ERROR** sera affiché disant que le "modules-manager" n'a pas réussi à démarrer les Workers du module avec le message de l'exception. Le module redémarrera dans le futur (*max 1 minute*).

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] The worker(s) failed to init and raised an exception: EXCEPTION. The module will be restarted in the future.
```

2. Un log en **ERROR** sera affiché disant que le "modules-manager" n'a pas réussi à démarrer les Workers du module, et que le module redémarrera dans le futur (*max 1 minute*).

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] The worker(s) failed to init. The module will be restarted in the future.
```

Si les Workers du module se sont bien démarrés alors ce log en **INFO** apparaîtra.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CREATION ] [ module_name ] [ module-type=module_type ] The worker(s) started.
```

Changement de configuration ou d'état du module - chapitre [UPDATE]



Ce chapitre ne concerne pas le démon Synchronizer ni le démon Arbiter.

Rajout d'un nouveau module

Lorsqu'un démon ou un module reçoit un nouveau module, le "modules manager" affiche un log en **INFO** expliquant qu'il va lancer le nouveau module.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ UPDATE ] [ module_name ]  
Launched as a NEW module as it was added in the configuration.
```

Suppression d'un module

Lorsqu'un module est enlevé de la configuration du démon ou de son module, le "modules-manager" affichera un log en **INFO** informant que l'on supprime le module, car il n'est plus dans la configuration de son démon ou de son module.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ UPDATE ] [ module_name ]  
Removing the module because it is removed in the configuration.
```

Si le "modules-manager" a essayé de supprimer le module, mais qu'il n'a pas été trouvé dans sa liste de module en cours d'exécution alors un log en **WARNING** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ UPDATE ] [ module_name ]  
Trying to remove the module module_name but it is not found in the current running modules:  
LIST_CURRENT_RUNNING_MODULES
```

Où `LIST_CURRENT_RUNNING_MODULES` est la liste des modules en cours d'exécution.

Quand le "modules-manager" éteint les modules, si un module ne réussit pas à s'éteindre alors un log en **ERROR** sera affiché, suivi de la traceback de l'exception.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ UPDATE ] [ module_name ]  
The module MODULE_NAME failed to stop and raised an exception : EXCEPTION
```

Changement de configuration d'un module (et sa relance)

Le démon ou le module a détecté que la configuration de son module en cours d'exécution a changé. Le "modules-manager" indique avec un log **INFO** qu'il va redémarrer le module afin de prendre en compte la nouvelle configuration.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ UPDATE ] [ module_name ]  
Configuration of the module did change so we restart it.
```

Le module n'a pas réussi à se mettre à jour

Le démon ou le module n'arrive pas à mettre à jour son module. Le "modules-manager" informe qu'il faut contacter le support avec un log en **ERROR**.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ UPDATE ] [ module_name ]  
We cannot update 'module_name' module. Please contact your Support.
```

Arrêt du module - chapitre [SHUTDOWN]

Lorsqu'un démon s'éteint, il va d'abord demander au "modules-manager" d'éteindre ses modules.

Le démon éteint ses modules

Ce log en **INFO** indique que le "modules-manager" va commencer à éteindre tous ses modules.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ SHUTDOWN ] Start to shutdown  
all modules.
```

Le module s'éteint

Ce log en **INFO** indique que le "modules-manager" est en train d'arrêter le module.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ SHUTDOWN ] Stopping module module_name.
```

Si une exception Python a été remontée par le "modules-manager" lors de l'arrêt du module, un log apparaîtra en **ERROR** avec le nom du module et le message de l'exception Python.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ SHUTDOWN ] Stopping module failed. The quit() function on module [ module_name ] failed with error: EXCEPTION.
```

Si le "modules-manager" a essayé de supprimer un module, mais qu'il ne l'a pas trouvé dans sa liste de modules en cours d'exécution alors un log en **ERROR** sera affiché.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ SHUTDOWN ] Trying to remove the module module_name but it is not found in the current running modules: LIST_CURRENT_RUNNING_MODULES
```

Le démon a fini d'éteindre ses modules

Le "modules-manager" informe avec un log **INFO** qu'il a terminé d'arrêter tous les modules.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ SHUTDOWN ] Stopping all modules ended.
```

Arrêt inopiné du module - chapitre [CRASH]

Le module s'est arrêté de façon inattendu

Lorsqu'un module n'est plus en cours d'exécution et qu'il s'est éteint de façon inattendue, le "modules-manager" affichera un log en **ERROR**.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CRASH ] [ module_name ] [ module-type=module_type ] The module module_name goes down unexpectedly!
```

Lorsqu'un ou plusieurs Worker(s) d'un module ne sont plus en cours d'exécution et qu'ils se sont éteint(s) de manière inattendue, le "modules-manager" affichera un log en **ERROR**.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CRASH ] [ module_name ] [ module-type=module_type ] The module module_name worker(s) goes down unexpectedly!
```

Le module a un comportement anormal

Lorsqu'un module externe a une taille de liste d'attente (*de commande interne ou de retour de commande interne*) **plus élevée** que la taille maximale définie dans le fichier de configuration **ini** du démon (*max_queue_size*), un log sera affiché en **ERROR**.

- **QUEUE_SIZE** = Taille (*en nombre d'éléments*) de la liste d'attente. Éléments possibles : Commande(s) ou retour(s) de commande(s).
- **QUEUE_MAX_SIZE** = Taille (*en nombre d'éléments*) maximale de la Queue définie dans le fichier **ini** du démon (*paramètre max queue size*).

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ daemon_or_module_name ] [ MODULES-MANAGER ] [ CRASH ] [ module_name ] [ module-type=module_type ] The external module module_name got a too high brok queue size ( QUEUE_SIZE > QUEUE_MAX_SIZE )!
```