

Stats Disks by SSH

- Contexte
- Paramétrage
 - Données utilisées provenant du modèle
 - Données communes pour les checks des modèles
 - Authentification
 - Fichiers temporaires
 - Données utilisées provenant du check
- Résultat
 - Exemple
 - Interprétation des données
 - Statut
 - Résultat
- Métriques
 - Définition
 - Exemple
- Les Erreurs
 - Erreurs spécifiques à ce check
 - MONITORED HOST - BAD STATE – Permission denied
 - Erreurs de connexion (communes à tous les checks)
 - UNKNOWN – Username/PublicKey combination invalid
 - UNKNOWN – Unable to extract public key from private key file : Unable to open private key file
 - UNKNOWN – Unable to extract public key from private key file : Wrong passphrase or invalid/unrecognized private key file format
 - UNKNOWN – Connection refused (os error 111)
 - UNKNOWN – Name or service not known

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

Chaque démon va démarrer son "modules-manager" qui va lister les répertoires dans **/var/lib/shinken/modules** et va charger chaque module présent.

Chargement du code

Le "modules-manager" va commencer à charger son module et ne logguera que si il rencontre un problème.

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 : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=architecture-export ] Import module [architecture-export] failed: EXCEPTION.
```

Chargement du module en tant que module Python (Spécifique Python)

Une fois que le "modules-manager" à charger le code du module, il va vérifier si le module peut être importé en tant qu'un module Python.

Si le "modules-manager" ne peut pas importer le code du module en tant que module python alors des logs en **WARNING** formant un bloc apparaîtrons.

- Un premier log expliquant que le "modules-manager" n'a pas réussi à importer le module en tant que python module.

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

- S'il manque le fichier '__init__.py' dans le dossier du module (Sans ce fichier il n'est pas possible d'importer le module en tant que module Python) :

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=my_broker_module ] - because of missing file : /var/lib/shinken/modules/my_broker_module
/__init__.py.
```

- Un troisième log informant qu'il n'est pas possible de faire un import : 'import my_broker_module.my_file '. Car le module n'a pas été importé en tant que module python.

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=my_broker_module ] - 'import my_broker_module.my_file' will not work in your module.py file.
```

Vérification des propriétés du module

Une fois que le modules-manager a importé son module , il va vérifier si le dictionnaire "properties" existe.

S'il n'existe pas, le log en **ERROR** suivant sera affiché:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CODE-LOADING ] [
directory=my_broker_module ] Missing properties dict in your module file /var/lib/shinken/modules
/my_broker_module/module.py. The module will not be imported.
```

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 ces 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 : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-
type=webui ] FAIL The module get_instance() call did not return any instance.
```

- 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 : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-
type=webui ] [0.046s] FAIL The module creation did fail with an exception: EXCEPTION. Will try to
recreate it in the future.
```

Si le module a bien été créée alors un log en **INFO** sera affiché son temps de création.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-
type=webui ] [0.046s] SUCCESS 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 : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-
type=webui ] 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 rencontré:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-
type=webui ] 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" a essayer 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 : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-
type=webui ] 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 : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-type=webui ] 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 : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-type=webui ] SUCCESS The module is started.
```

Démarrage du/des Worker(s) du module

Une fois le module démarré, s'il utilise des Workers, le "modules-manager" va essayer de les démarrer. Ce log sera en **INFO**.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-type=webui ] Starting a worker(s).
```

Si le "modules-manager" n'a pas réussi à démarrer le/les Worker(s) du 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 le/les Worker(s) du module avec le message de l'exception.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-type=webui ] The worker(s) failed to init and raised an exception: EXCEPTION.
```

2. Alors un log en **ERROR** sera affiché disant que le "modules-manager" n'a pas réussi à démarrer le/les Worker(s) du module.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-type=webui ] The worker(s) failed to init.
```

Si le/les Worker(s) du module ce sont bien démarré alors ce log en **INFO** apparaîtra.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ CREATION ] [ WebUI ] [ module-type=webui ] SUCCESS The worker(s) is created.
```

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

Rajout d'un nouveau module dans un démon

Lorsqu'un démon reçoit un nouveau module, le "modules-manager" affichera un log en **INFO** comme quoi le module a été ajouté à la configuration du démon.

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

Suppression d'un module dans un démon

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

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ UPDATE ] [ WebUI ] Removing the module because it is removed from the configuration.
```

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

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

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

Le démon a détecté que la configuration de son module n'est plus la même qu'avant le "modules-manager" annonce avec un log **INFO** qu'il va redémarrer son module afin de prendre la nouvelle configuration en compte.

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

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

Le démon à trouver une nouvelle configuration venant d'un ancien ou d'un nouveau module, mais il n'arrive pas à le mettre à jour. Le "modules-manager" l'informe avec un log en **ERROR**.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ UPDATE ] [ WebUI ] Cannot find previous or new module definition for the module name WebUI, so cannot update it.
```

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 : [ broker-master ] [ 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 : [ broker-master ] [ MODULES-MANAGER ] [ SHUTDOWN ] Stopping module WebUI.
```

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 : [ broker-master ] [ MODULES-MANAGER ] [ SHUTDOWN ] Stopping module FAILED. The quit() function on module [ WebUI ] failed with error: EXCEPTION.
```

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

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ SHUTDOWN ] Trying to remove the module WebUI 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 à terminé d'arrêter tous les modules.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ MODULES-MANAGER ] [ SHUTDOWN ] End of stopping all modules.
```

Arrêt inopiné du module - chapitre [CRASH]

Le module s'est arrêté de façon inattendu

Lorsqu'un module externe 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 : [ broker-master ] [ MODULES-MANAGER ] [ CRASH ] [ my_broker_module ] [ module-type=my_broker_module_type ] The external module my_broker_module goes down unexpectedly!
```

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

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CRASH ] [ my_broker_module ] [ module-type=my_broker_module_type ] The module my_broker_module worker(s) goes down unexpectedly!
```

Si un module a pu prévenir lors de son arrêt (par exemple la WebUI détecte qu'elle a un problème avec ses fichiers *.html* ou *.js*, elle peut prévenir le démon avant de s'arrêter), on aura le log explicatif en **ERROR** issu du module.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ broker-master ] [ MODULES-MANAGER ] [ CRASH ] [ WebUI ] [ module_type=webui ] The __shinken_lang__ variable was not found in the index.html file: there is a critical error with your installation. Please open a ticket to your support.
```

Le module à un comportement anormal

Lorsqu'un module externe à une taille de liste d'attente commande interne ou de retour d'internal commande **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 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 : [ broker-master ] [ MODULES-MANAGER ] [ CRASH ] [ synchronizer-module-database-backup ] [ module-type=synchronizer_module_database_backup ] The external module synchronizer-module-database-backup got a too high brok queue size (QUEUE_SIZE > QUEUE_MAX_SIZE)!
```