

Activer les webservices pour les checks passifs

Quest-ce qu'un webservice pour les checks passifs

Le receiver a un module spécial pour les [checks passifs](#). Ce module est appelé ws-arbiter et va ouvrir par défaut le port 7760 TCP/HTTP. Vous pourrez inscrire vos checks dans le receiver grâce à ce module. L'Arbiter viendra alors récupérer cet état et le fera passer au Scheduler.

Configuration du module

La configuration de ce module se trouve dans le fichier `/etc/shinken/modules/ws-arbiter.cfg`

Par défaut ce sera :

```
#####
# ws-arbiter (webservice)
#####
# Daemons that can load this module:
# - receiver
# - arbiter
# This module is a webservice that can be used to send checks to Shinken Enterprise
# as POST HTTP(s)
#####

define module {

    ##### Module identity #####
    # Module name. Must be unique
    module_name          ws-arbiter
    # Module type (to load module code). Do not edit.
    module_type          ws_arbiter

    ##### Listening address #####
    # host: IP address to listen to.
    #       note: 0.0.0.0 = all interfaces.
    host                 0.0.0.0

    # port to listen
    port                 7760

    # HTTPs part, enable if you want to set the listening for HTTPS instead of default HTTP.
    # disabled by default. Set your own certificates.
    use_ssl              0
    ssl_cert             /etc/shinken/certs/server.cert
    ssl_key              /etc/shinken/certs/server.key

    ##### HTTP authentication #####
    # You can use HTTP basic authentication method for this module.
    # If username is set to anonymous and password is commented, then
    # no authentication will be required.
    username             anonymous
    #password            secret

}

```

Les valeurs peuvent être :

- `module_name`: définit un nom unique pour le module
- `module_type`: doit être `ws_arbiter`
- `host`: l'adresse à écouter IP . 0.0.0.0 signifie "toutes les interfaces"
- `port`: port TCP à écouter
- `use_ssl` et `ssl_cert` / `ssl_key` : permet que le module écoute sur le port en SSL (protocole HTTPS) et d'utiliser des certificats
- `username`&`password`: si mis à "anonymous" aucune accréditation nécessaire. Si vous mettez un nom utilisateur/mot de passe, une authentification est nécessaire

Exemple d'un push de check passif :

Pour passer le résultat d'un check d'un hôte:

```
curl -u user:password -d "host_name=mon-hote&service_description=check-de-mon-hote&return_code=0&output=Everything OK" http://ip-du-receiver:7760/push_check_result
```

- (optionnel) time_stamp= horodatage du check
- host_name= nom de l'hôte auquel vous voulez envoyer le check
- (si check) service_description = nom du check pour lequel vous voulez pousser un résultat
- return_code= [0,1,2,3] valeur pour le code retour. (pour rappel 0 = OK, 1 = WARNING, 2 = CRITIQUE, 3 = UNKNOWN)
- output= texte simple pour le check
- -u user:password : nécessaire que si le module demande une authentification

Voici un autre exemple de curl avec un résultat qui contient des caractères spéciaux :

```
curl -d "host_name=mon-hote&return_code=2&output=*DOWN:état incorrect*" http://ip-du-receiver:7760/push_check_result
```