

email - Méthode de notification

Sommaire

[Contexte](#)
[Exemple de notifications](#)
 [Notify-host-by-email](#)
 [Notify-service-by-email](#)

Description

Le module `named-pipe` permet à Shinken de récupérer des informations via un fichier "passe plat" (*un pipe*) que l'on peut remplir de la manière de notre choix.

Par exemple, ce module peut envoyer des informations à des hôtes possédant des checks passifs (*checks qui au lieu d'effectuer une vérification régulière, va attendre de recevoir des données, plus d'informations sur les checks passifs ici : [Mode actif et mode passif](#)*).

Pour un exemple d'utilisation du module, veuillez vous référer à la section [exemple d'utilisation](#).

Activation du module

Le module `named-pipe` est un module qui peut être activé sur le démon Receiver et de l'Arbiter.

- L'activation du module s'effectue en ajoutant le nom de ce module dans le fichier de configuration du démon Receiver (*ou de l'Arbiter*).
- Pour ce faire, ouvrir le fichier de configuration du Receiver à l'emplacement `/etc/shinken/receivers/nom_du_receiver.cfg`, et ajouter le nom de votre module "`named-pipe`".

Exemple: par défaut, nous livrons un module dont le nom est "`named-pipe`":

```
define receiver {
    [...]
    modules          Module 1, Module 2, named-pipe
    [...]
}
```

Pour prendre en compte le changement de configuration, redémarrer l'Arbiter:

```
service shinken-arbiter restart
```

Configuration

Nous ne livrons pas de configuration par défaut. Vous en trouverez une ici : [named-pipe.cfg](#)

Il faut ensuite placer ce fichier avec les autres modules dans `/etc/shinken/modules/`



Pensez à modifier les droits de lecture du fichier `/etc/shinken/modules/named-pipe.cfg` et de changer son propriétaire :

Donner les droits d'écriture :

```
chmod 644 /etc/shinken/modules/named-pipe.cfg
```

Changer de propriétaire :

```
chown shinken:shinken /etc/shinken/modules/named-pipe.cfg
```

```
## Module:      named-pipe
## Loaded by:   Receiver
# Receive passive host and service results, typically from check_mk plugins.
# No other commands or inputs accepted (Restricted to host and service results)
define module {
    module_name      named-pipe
    module_type      named_pipe
    command_file     /var/lib/shinken/shinken.cmd
}
```

Détails des sections composant le fichier de configuration

Identification du module

Il est possible de définir plusieurs instances de module de type `named-pipe` dans votre architecture Shinken.

- Chaque instance devra avoir un nom unique.

Nom	Type	Unité	Défaut	Commentaire
<code>module_name</code>	Texte	---	<code>named-pipe</code>	Nous vous conseillons de choisir un nom en fonction de l'utilisation du module pour que votre configuration soit simple à maintenir. Doit être unique.
<code>module_type</code>	Texte	---	<code>named-pipe</code>	Ne peut être modifié.

Configuration du fichier passe-plat

Nom	Type	Unité	Défaut	Commentaire
<code>command_file</code>	Chemin	---	<code>/var/lib/shinken/shinken.cmd</code>	Endroit où le module ira lire les informations (comme des traps à envoyer aux éléments concernés).

Exemple d'utilisation du module

Création d'un check passif

Dans l'interface de configuration, il faut :

- Créer un check passif à associer à un hôte, non actif et volatile, car si nous souhaitons recevoir une notification dès un changement d'état. L'expiration de l'état de fraîcheur doit également être paramétrée.

- Appliquer le modèle d'hôte à un hôte accessible sur le réseau (le paramètre "adresse" doit être rempli avec un nom d'hôte ou IPv4 accessible).

Voici un exemple de fichier de configuration d'un check passif et d'un modèle d'hôte :

```
define service{
    service_description    TRAP
    check_command          check-host-alive
    host_name              TRAP-modele
    is_volatile            1
    passive_checks_enabled 1
    active_checks_enabled  0
    check_freshness        1
    freshness_threshold    300
    register               0
    check_interval         1
    retry_interval         1
}

define host{
    name                  TRAP-modele
    register              0
}
```

Maintenant, il faut pousser ces éléments en production afin de pouvoir observer le changement d'état de notre hôte.

Création d'un script pour envoyer des traps SNMP

- Maintenant, il faut un script qui va se charger d'écrire les futurs traps SNMP dans le pipe précisé dans la configuration du module (dans notre cas, `/var/lib/shinken/shinken.cmd`).
- Ajouter le script suivant que l'on appellera "submit_check_result" dans le dossier des plugins Shinken (`/var/lib/shinken-user/libexec/`):

```
#!/bin/bash

# Arguments:
# ${1} = host_name (Short name of host that the service is associated with)
# ${2} = svc_description (Description of the service)
# ${3} = return_code (An integer that determines the state of the service check, 0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN).
# ${4} = plugin_output (A text string that should be used as the plugin output for the service check)

# Ensuring we use the correct commands by using their full absolute path
echoCmd="/bin/echo"
commandfile="/var/lib/shinken/shinken.cmd"

# get the current date/time in seconds since UNIX epoch
datetime="$(date +%s)"

# create the command line to add to the command file
cmdline="[${datetime}] PROCESS_SERVICE_CHECK_RESULT;${1};${2};${3};${4}"

# append the command to the end of the command file
${echoCmd} "${cmdline}" >> "${commandfile}"
```



Ne pas oublier de rendre le script exécutable et de donner les droits du fichier à l'utilisateur Shinken :

```
chown shinken:shinken /var/lib/shinken-user/libexec/submit_check_result
chmod +x /var/lib/shinken-user/libexec/submit_check_result
```

Pour tester le script et simuler une réception d'un trap translaté au format Shinken, il suffit d'exécuter la commande suivante qui va faire passer le service en état critique :

```
/var/lib/shinken-user/libexec/submit_check_result TRAP 2 "test envoi trap - CRITIQUE"
```

Le check devrait passer en critique, et si au bout de la période du seuil de fraîcheur, aucun nouveau trap n'a été reçu, alors la commande check-host-alive fera repasser le check à OK (*si bien sûr l'hôte est accessible via le réseau*).