

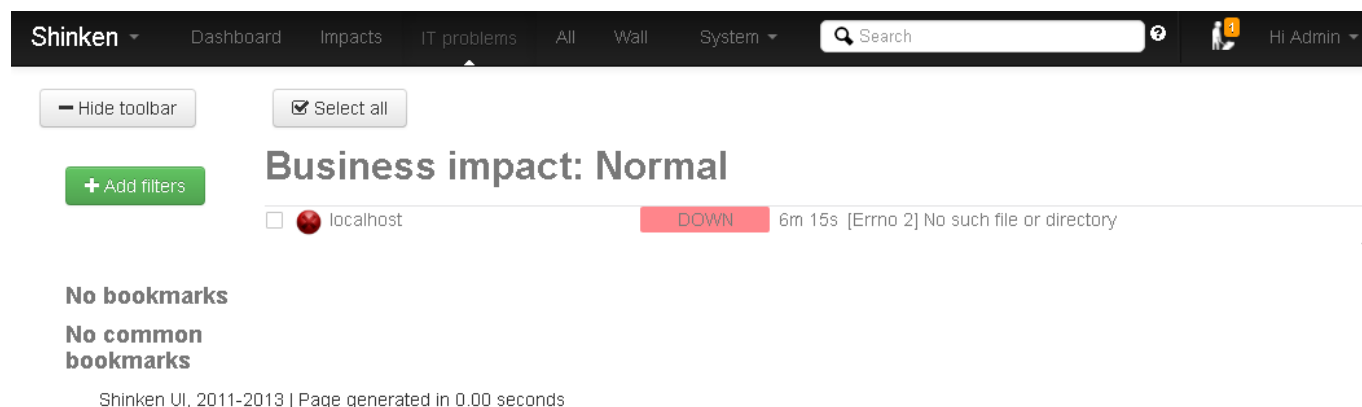
# Activité : Pourquoi et comment une machine est-elle supervisée ?

## Présentation

Ouvrez les différents fichiers présentés, regardez leur contenu afin de comprendre leurs relations et effectuer les actions permettant de tester **localhost**.

## Webui

**Webui** (l'interface utilisateur Web proposée par défaut par Shinken) montre que la supervision ne fonctionne pas. Pour Shinken l'hôte localhost est **down** ce qui **n'est pas possible** car vous avez l'interface Web.



Les différents états gérés par Shinken sont :

- Pour un hôte :
  - Un hôte est une machine accessible par son adresse IP pouvant prendre 4 états (**UP, DOWN, UNREACHABLE, PENDING**) :
    - **Up** → l'hôte répond,
    - **Down** → l'hôte ne répond pas,
    - **Unreachable** → l'hôte est injoignable car il se trouve derrière un autre hôte qui ne répond pas,
    - **Pending** → l'hôte n'est pas encore testé (au démarrage généralement).
- Pour un service :
  - Un service est un élément supervisé sur un hôte (**qui doit donc être UP**) pouvant prendre 5 états (**OK, WARNING, CRITICAL, UNKNOWN, PENDING**) :
    - **Warning** → des problèmes non bloquants,
    - **Critical** → des problèmes bloquants,
    - **Unknown** → état non testable car la commande (le plugin) a un problème,
    - **Pending** → non encore testé (généralement au démarrage).

L'hôte ne répond pas dans notre cas, mais pourquoi devrait-il répondre et à quoi ne répond-il pas ?

# Les hôtes

## Pourquoi Shinken supervise la machine localhost ? :

- Parce que l'installation a créé un fichier **localhost.cfg** dans le répertoire **\$ETC/hosts**.
- Tous les hôtes présents (soit manuellement, soit dynamiquement, comme on verra dans d'autres activités) dans ce répertoire seront donc supervisés.

Si on ouvre le fichier **localhost.cfg** (/etc/shinken/hosts/localhost.cfg) on y lit :

```
define host{
    use                generic-host
    contact_groups     admins
    host_name          localhost
    address            localhost
}
```

Le mot clé **define** permet de définir un **objet** gérable par Shinken, ici un hôte (**host**) identifié par son nom (**localhost**). Cet identifiant est important car il peut être utilisé dans d'autres fichiers de configuration.

Son adresse est **localhost**, elle sera utilisée comme on le verra, par la variable **\$HOSTADDRESS\$** (on utilise un nom ou une adresse IP, dans notre cas **localhost** ou **127.0.0.1**, mais si on utilise un nom il faut une méthode de résolution de noms, dans ce cas le fichier **/etc/hosts**), et le groupe à contacter en cas de problème (**contact\_groups**).

## Les modèles (templates)

Il y a peu de choses dans le fichier précédent car la définition utilise un **modèle (template)** par l'intermédiaire de la directive **use**.

Le modèle (**template**) utilisé est **generic-host**.

La plupart des modèles sont décrits (définis) dans le sous-répertoire **\$ETC/templates/** ou comme on le verra plus loin dans **\$ETC/packs/**

Dans ce répertoire on va trouver le fichier **generic-host.cfg** :

```
# Generic host definition template - This is NOT a real host, just a
template!
# Most hosts should inherit from this one
define host{
    name                generic-host

    # Checking part
    check_command        check_host_alive
    max_check_attempts   2
    check_interval       5
```

```
# Check every time
active_checks_enabled      1
check_period                24x7

# Notification part
# One notification each day (1440 = 60min* 24h)
# every time, and for all 'errors'
# notify the admins contactgroups by default
contact_groups             admins,users
notification_interval      1440
notification_period        24x7
notification_options        d,u,r,f
notifications_enabled      1

# Advanced option. Look at the wiki for more informations
event_handler_enabled      0
flap_detection_enabled     1
process_perf_data          1

# Maintenance period
#maintenance_period        workhours

# Dispatching
#poller_tag                DMZ
#realm                     All

# For the WebUI
#icon_set                  server ; can be database, disk, network_service,
server

# This said that it's a template
register                    0
}
```

Ce fichier définit un modèle identifié par le nom **generic-host**, la directive **register** indique qu'il s'agit d'un modèle grâce à la valeur 0. Un modèle n'est pas instancié, il doit être appelé par un objet.

Ce modèle appelle la commande **check\_host\_alive**.

## Les commandes

La commande **check\_host\_alive** est définie dans **\$ETC/commands/check\_host\_alive.cfg**.

```
define command {
command_namecheck_host_alive
command_line    $NAGIOSPLUGINDIR$/check_ping -H $HOSTADDRESS$ -w 1000,100%
               -c 3000,100% -p 1
}
```

Comme on peut voir la commande **check\_host\_alive** est une référence logique vers la commande réelle **check\_ping**.

Cette commande se situe dans le répertoire désignée par la variable **\$NAGIOSPLUGINDIR\$**, cette variable est définie dans le fichier **\$ETC/resource.d/path.cfg** .

```
# Nagios legacy macros
$USER1$=$NAGIOSPLUGINDIR$
$NAGIOSPLUGINDIR$=/usr/lib/nagios/plugins

#-- Location of the plugins for Shinken
$PLUGINDIR$=/var/lib/shinken/libexec
```

Cette commande est une commande faisant partie des plugins nagios.

Mais si on regarde dans le répertoire **/usr/lib** on ne trouve pas de sous répertoire **nagios**. Il faut donc installer le plugin.

```
root@ctShinken:~# apt-get install nagios-plugins
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
...
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Paramétrage de nagios-plugins (1.4.16-1) ...
Paramétrage de nagios-plugins-contrib (4.20120702) ...
Traitement des actions différées (« triggers ») pour « menu »
```

Les plugins installés sont nombreux :

```
root@ctShinken:~# ls /usr/lib/nagios/plugins/
check_apt          check_imap_receive_epncheck_rbl
check_backuppc    check_ipmi_sensor    check_real
check_breeze      check_ircd           check_rpc
check_by_ssh      check_jabber         check_rta_multi
check_cert_expire check_ldap            check_running_kernel
check_clamd       check_ldaps          check_sensors
check_cluster     check_libs           check_simap
check_dhcp        check_lm_sensors     check_smtp
check_dig         check_load           check_smtp_send
check_disk        check_log            check_smtp_send_epn
check_disk_smb    check_mailq          check_snmp
check_dns         check_memcached      check_snmp_environment
check_dnssec_delegation check_mrtg          check_soas
check_dummy      check_mrtgtraf       check_spop
check_email_delivery check_multipath      check_ssh
check_email_delivery_epn check_mysql          check_ssl_cert
check_entropy     check_mysql_health   check_ssmtp
check_file_age    check_mysql_query    check_statusfile
check_flexlm      check_nagios         check_swap
check_fping       check_nntp           check_tcp
```

check_ftp	check_nntp	check_time
check_game	check_nt	check_udp
check_haproxy	check_ntp	check_ups
check_host	check_ntp_peer	check_users
check_hpasm	check_ntp_time	check_wave
check_hpjd	check_nwstat	check_webinject
check_http	check_oracle	check_whois
check_httpd_status	check_overcr	check_zone_auth
check_icmp	check_packages	check_zone_rrsig_expiration
check_ide_smart	check_pgsql	imap_ssl_cert
check_ifoperstatus	check_ping	imap_ssl_cert_epn
check_ifstatus	check_pop	negate
check_imap	check_printer	urlize
check_imap_quota	check_procs	utils.pm
check_imap_quota_epn	check_radius	utils.sh
check_imap_receive	check_raid	

On y trouve notamment la commande **check\_ping**.

Cette commande est documentée ici : [https://www.monitoring-plugins.org/doc/man/check\\_ping.html](https://www.monitoring-plugins.org/doc/man/check_ping.html)

L'option

- -w indique un "warning",
- -c un événement "critical",
- rta signifie "round trip average travel"
- et -p le nombre de paquets "icmp echo" envoyés.

On peut d'ailleurs exécuter cette commande manuellement pour la tester.

```
root@ctShinken:~# cd /usr/lib/nagios/plugins/
root@ctShinken:/usr/lib/nagios/plugins# ./check_ping -H localhost -w
1000,100% -c 3000,100% -p 1
PING OK - Paquets perdus = 0%, RTA = 0.29
ms|rta=0.291000ms;1000.000000;3000.000000;0.000000 pl=0%;100;100;0
```

Cela fonctionne. Que dit maintenant webui ?

The screenshot shows the Shinken web interface. At the top, there's a navigation bar with 'Shinken' and various tabs like 'Dashboard', 'Impacts', 'IT problems', 'All', 'Wall', 'System'. A search bar and user profile 'Hi Admin' are also present. Below the navigation bar, there are buttons for 'Hide toolbar' and 'Select all'. The main content area displays 'Business impact: Normal' in large blue text. Below this, there's a status bar for 'localhost' showing 'UP' in a green box, '1m 51s' uptime, and 'PING OK - Packet loss = 0%, RTA = 0.04 ms'. On the left side, there are buttons for '+ Add filters', 'No bookmarks', and 'No common bookmarks'. At the bottom, a footer indicates 'Shinken UI, 2011-2013 | Page generated in 0.00 seconds'.

L'hôte **localhost** est up !



Remarque : **\$HOSTADDRESS\$** a été remplacé par l'adresse trouvée dans le fichier **\$ETC/hosts/localhost.cfg**, directive **address**.

```
address localhost
```

## Je reviens au menu Shinken



- [Supervision des services avec Shinken](#)

From:

<https://siocours.lycees.nouvelle-aquitaine.pro/> - **Les cours du BTS SIO**

Permanent link:

[https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/reseau/supervision/shinken\\_08](https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/reseau/supervision/shinken_08)

Last update: **2016/11/08 23:41**

