



C05-net-std-supp - TP LINUX - MTN

Guillaume ASTIER

26/02/16



Table des matières

| | |
|--|----------|
| Network | 1 |
| Configuration | 1 |
| Systemd VS systemV | 1 |
| systemd | 2 |
| systemV | 2 |
| Server Web python | 2 |
| Server Web nginx | 2 |
| Supervision | 2 |
| Client de supervision (monitor) | 2 |
| Machine a superviser (client et server) | 3 |
| Man Page | 3 |
| modification de la conf sur le client et le server (monithor-server) | 3 |
| Ajoutez la sonde nginx | 3 |
| Ajoutez l'interrogation depuis monitor | 3 |
| Sonde/script nombre de process | 4 |

Network

Configuration

Les instances docker sont déjà configuré avec un adressage IP

Vous devez cependant modifier vots fichiers hosts de cette manière (sur les 3 instnaces):

```
1 10.10.10.3 astier-g-client client
2 10.10.10.4 astier-g-server server
3 10.10.10.5 astier-g-monitor monitor
```

Puis vérifier que vous pouvez vous y connecter avec la commande netcat

Systemd VS systemV

Pour démarrez un service vous devez (sur les distribution "récente") utilisez systemctl.

Cependant, sous docker, seul systemV fonctionnera car ce dernier n'est pas dépendant du PID 1

Essayez de démarrer le service cron sur server avec systemctl (systemd) puis avec service (systemV)

Les commande pour vérifier / démarrer / arrêter un service sont :

systemd

```
1 systemctl [start/stop/status/restart] SERVICE_NAME
```

systemV

```
1 service SERVICE_NAME [start/stop/status/restart]
2 # ou
3 /etc/init.d/SERVICE_NAME [start/stop/status/restart]
```

Server Web python

Le principe d'un serveur web est de pouvoir proposer en lecture des pages écrite en html

Sur serveur lancez la commande ci dessous en vous plaçant dans le home de isen (en root)

```
1 root@astier-g-server:/home/isen# python3 -m http.server 80
```

Vérifiez la connexion depuis client

Server Web nginx

Installez le paquet nginx sur server

Depuis client ou monitor utilisez le navigateur web elinks pour vérifier que le service tourne

Que devez vous faire pour que le service fonctionne

Supervision

Client de supervision (monitor)

Dans /opt/deb/ vous trouverez le paquet 'monithor-client_1.14-rebuild_all.deb'

Ce dernier récupère les résultats des sondes envoyé aux instances a superviser

Il doit donc être installé sur la machine : monitor

Pour installer ce paquet il vous faudra utiliser la commande dpkg (regardez le man)

Machine a superviser (client et server)

Dans /opt/deb/ vous trouverez le paquet 'monithor-server_1.14-rebuild_all.deb'

Ce dernier permet de réceptionner des demande d'un client de supervision et renvoyé le résultat des scripts associés au sondes.

Il doit donc être installé sur les machines à supervier : server & client

Pour installer ce paquet il vous faudra utiliser la commande dpkg (regardez le man)

Man Page

Les manpages sont disponibles

```
1 man monithor-client
2 man monithor-server
```

modification de la conf sur le client et le server (monithor-server)

Faite un démarrage avec service ou les scripts init de monithor server

Que constatez vous.

Modifiez la conf en conséquences et relancez le service

Ajoutez la sonde nginx

Ajoutez une sonde et le script associé sur le client afin qu'il vérifie que le service nginx de la machine server est UP (gestion de netcat ou curl)

Ajoutez une sonde et le script associé sur le server afin qu'il vérifie que son service nginx est UP (gestion de ps)

Ajoutez l'interrogation depuis monitor

Sur l'instance monitor faite une manipulation (cf exemple) pour intéroger les instance server et client sur la sonde nginx

Sonde/script nombre de process

Faites les mêmes actions pour pouvoir déterminer si une instance à des processus en trop.