

Mission I2 - Supervision des serveurs ap31-prod et ap31-test

Objectifs

On souhaite superviser le fonctionnement des deux serveurs **ap3x-prod** et **ap3x-test**.
Pour ce faire, on utilisera le logiciel de supervision **nagios** hébergé sur la machine Debian **ap3x-mon**.

Le logiciel de supervision devra :

- s'assurer que les deux machines sont joignables (**i2-c1**)
- et accessibles par SSH (**i2-c2**)
- et accessibles par web pour **ap3x-test** (**i2-c3**)
- vérifier sur ces deux machines au moyen de SNMP : charge CPU, occupation mémoire, espace disque, débit réseau ... (**i2-c4**)
- envoyer une notification par SMTP en cas de problème (**i2-c5**)
- la passerelle **gwsiox** devra également être supervisée ainsi que les serveur **px** et **pxlabx** et **gwlab** (**i2-c6**)
- gérer les dépendances (**px**, **gwsiox**, **gwlab**, **pxlabx**) (**i2-c7**)

Pour installer **Nagios4** et configurer la supervision des serveurs **ap31-prod** et **ap31-test** sur la machine Debian **ap31-mon**, nous avons suivi ces étapes.

1. Installation des paquets nécessaires :

```
apt-get update  
apt-get install -y vim curl wget apache2 php nagios4 nagios-plugins-contrib snmpd
```

2. Copie de la configuration d'Apache pour Nagios :

```
cp /etc/nagios4/apache2.conf /etc/apache2/sites-available/nagios4.conf
```

3. Activation le site Nagios dans Apache et activez les modules nécessaires :

```
a2ensite nagios4.conf  
a2enmod rewrite cgi
```

4. Recharge de la configuration d'Apache :

```
systemctl reload apache2
```

On a vérifié l'état du service Nagios4 :

```
root@ap31-mon:~# systemctl status nagios4  
* nagios4.service - nagios4  
   Loaded: loaded (/lib/systemd/system/nagios4.service; enabled; preset: enabled)  
   Active: active (running) since Thu 2023-10-12 08:29:14 CEST; 3min 31s ago  
     Docs: man:nagios4  
  Main PID: 8533 (nagios4)  
    Tasks: 6 (limit: 38386)  
   Memory: 2.4M  
      CPU: 66ms  
   CGroup: /system.slice/nagios4.service  
           |-8533 /usr/sbin/nagios4 /etc/nagios4/nagios.cfg  
           |-8535 /usr/sbin/nagios4 --worker /var/lib/nagios4/rw/nagios.qh  
           |-8536 /usr/sbin/nagios4 --worker /var/lib/nagios4/rw/nagios.qh  
           |-8537 /usr/sbin/nagios4 --worker /var/lib/nagios4/rw/nagios.qh  
           |-8538 /usr/sbin/nagios4 --worker /var/lib/nagios4/rw/nagios.qh  
           ~8869 /usr/sbin/nagios4 /etc/nagios4/nagios.cfg  
  
Oct 12 08:29:14 ap31-mon nagios4[8533]: wproc: Registry request: name=Core Worker 8535;pid=8535  
Oct 12 08:29:14 ap31-mon nagios4[8533]: wproc: Registry request: name=Core Worker 8535;pid=8535  
Oct 12 08:29:54 ap31-mon nagios4[8533]: Successfully launched command file worker with pid 8869  
Oct 12 08:29:54 ap31-mon nagios4[8533]: Successfully launched command file worker with pid 8869  
Oct 12 08:29:54 ap31-mon nagios4[8533]: HOST ALERT: localhost;DOWN;SOFT;1;/bin/ping -4 -n -U -w 10 -c 1 127.0.0.1  
Oct 12 08:30:29 ap31-mon nagios4[8533]: HOST ALERT: localhost;DOWN;SOFT;2;/bin/ping -4 -n -U -w 10 -c 1 127.0.0.1  
Oct 12 08:31:06 ap31-mon nagios4[8533]: HOST ALERT: localhost;DOWN;SOFT;3;/bin/ping -4 -n -U -w 10 -c 1 127.0.0.1  
Oct 12 08:31:44 ap31-mon nagios4[8533]: SERVICE ALERT: localhost;PING;UNKNOWN;HARD;1;/bin/ping -4 -n -U -w 10 -c 5 127.0.0.1  
Oct 12 08:32:06 ap31-mon nagios4[8533]: HOST ALERT: localhost;DOWN;SOFT;4;/bin/ping -4 -n -U -w 10 -c 1 127.0.0.1  
Oct 12 08:32:21 ap31-mon nagios4[8533]: HOST ALERT: localhost;DOWN;SOFT;5;/bin/ping -4 -n -U -w 10 -c 1 127.0.0.1
```

Mission I2 - Supervision des serveurs ap31-prod et ap31-test

5. Sur chaque machine à superviser :

```
apt update
apt install snmpd
```

6. On modifie le fichier `/etc/snmp/snmpd.conf` pour avoir une couche snmp opérationnelle (écoute sur tout les interfaces en IPV4 et IPV6 udp/161 et autoriser l'accès général à la communauté « public ») :

On commente `agentAddress udp:127.0.0.1:161` et on ajoute `agentAddress udp:161,udp6:[::1]:161`, on commente `rocommunity public localhost` et on ajoute `rocommunity public default` ou `(rocommunity public default #-V systemonly)`

7. A cette étape là, nous avons configuré **Nagios** pour superviser les serveurs **ap31-prod** et **ap31-test**. Pour ce faire, nous avons créé des fichiers de configuration spécifiques.

On a commencé par créer un fichier **hostgroups.cfg** pour définir les différents groupes (fichier dans le répertoire **/etc/nagios4/objects/**):

```
nano /etc/nagios4/objects/hostgroups.cfg
```

8. Ajout du contenu suivant au fichier **hostgroups.cfg**:

```
#Configuration pour groupe SNMP servers
define hostgroup {
    hostgroup_name    snmp-servers
    alias             SNMP Servers
    members           ap31-prod, ap31-test, ap31-wiki
}

# Configuration pour groupe web servers
define hostgroup {
    hostgroup_name    web-servers
    alias             Web Servers
    members           ap31-test, ap31-wiki
}

# Configuration pour groupe linux servers
define hostgroup {
    hostgroup_name    linux-servers
    alias             Linux Servers
    members           localhost, ap31-prod, ap31-test, ap31-wiki, gwsio5, gwlab, px, pqlab1, pqlab2, pqlab3
}
```

7. On a créé des fichiers par hôtes :

ap31-prod
<pre>define host { use linux-server ; Name of host template t> host_name ap31-prod alias ap31-prod Server address 172.16.0.82 parents pqlab1 }</pre>
ap31-test
<pre>define host { use linux-server ; Name of host template t> host_name ap31-test alias ap31-test Server address 172.16.0.95 parents pqlab2 }</pre>

Mission I2 - Supervision des serveurs ap31-prod et ap31-test

ap31-wiki

```
define host {  
    use                linux-server ; Name of host template t>  
    host_name          ap31-wiki  
    alias              ap31-wiki Server  
    address            172.16.0.87  
    parents            pxlab1  
}
```

gwlab

```
define host {  
    use                linux-server ; Name of host template t>  
    host_name          gwlab  
    alias              gwlab  
    address            10.121.38.253  
}
```

gwsio5

```
define host {  
    use                linux-server ; Name of host template t>  
    host_name          gwsio5  
    alias              gwsio5 Gateway  
    address            10.121.38.35  
    parents            gwlab  
}
```

px

```
define host {  
    use                linux-server ; Name of host template t>  
    host_name          px  
    alias              px Gateway  
    address            10.121.38.2  
    parents            gwlab  
}
```

pxlab1

```
define host {  
    use                linux-server ; Name of host template t>  
    host_name          pxlab1  
    alias              pxlab1  
    address            172.16.0.10  
}
```

pxlab2

```
define host {  
    use                linux-server ; Name of host template t>  
    host_name          pxlab2  
    alias              pxlab2  
    address            172.16.0.11  
}
```

pxlab3

```
define host {  
    use                linux-server ; Name of host template t>  
    host_name          pxlab3  
    alias              pxlab3  
    address            172.16.0.12  
}
```

Mission I2 - Supervision des serveurs ap31-prod et ap31-test

8. On a édité le fichier `/etc/nagios4/nagios.cfg` pour prendre en compte les fichiers `.cfg`

```
cfg_file=/etc/nagios4/objects/localhost.cfg
cfg_file=/etc/nagios4/objects/hostgroups.cfg
cfg_file=/etc/nagios4/objects/ap31-test.cfg
cfg_file=/etc/nagios4/objects/ap31-prod.cfg
cfg_file=/etc/nagios4/objects/services.cfg
cfg_file=/etc/nagios4/objects/ap31-wiki.cfg
cfg_file=/etc/nagios4/objects/gwsio5.cfg
cfg_file=/etc/nagios4/objects/gwlab.cfg
cfg_file=/etc/nagios4/objects/px.cfg
cfg_file=/etc/nagios4/objects/pxlab1.cfg
cfg_file=/etc/nagios4/objects/pxlab2.cfg
cfg_file=/etc/nagios4/objects/pxlab3.cfg
```

9. On crée un fichier pour définir les services (`services.cfg`) pour vérifier les informations SNMP. (fichier dans le même répertoire `/etc/nagios4/objects/`) :

```
nano /etc/nagios4/objects/commands.cfg
```

8. Ajout du contenu suivant au fichier `services.cfg` :

```
# Define service for web servers
define service{
    use                generic-service
    hostgroup_name     web-servers
    service_description WWW
    check_command       check_http
}

define service{
    use                generic-service
    hostgroup_name     linux-servers
    service_description SSH
    check_command       check_ssh
}

define service {
    use                local-service ; Name of service template to use
    hostgroup_name     linux-servers
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
}

# Define service for Load (Charge CPU)
define service{
    use                generic-service ; Use generic-service template
    hostgroup_name     snmp-servers ; S'applique au groupe en question
    service_description Charge CPU ; Nom du service
    check_command       check_lin_load!public!-v2c!nets!3,2!4,3,2 ; Command
    # servicegroups     ser-lin-base ;Service belong servicegroup
}

# Définition du service de supervision du débit réseau
define service {
    use                generic-service
    hostgoup_name      snmp-servers
    service_description Débit Réseau
    check_command       check_snmp_int!public!-v2c!eth0!80!90
}
```

Quelques commentaires :

- Dans les définitions de services, la ligne ``check_command`` spécifie la commande de vérification à utiliser, et les chiffres dans l'argument SNMP OID définissent la métrique SNMP spécifique à surveiller.

Mission I2 - Supervision des serveurs ap31-prod et ap31-test

- Un **OID (Object Identifier)** est une chaîne de nombres qui identifie de manière unique un objet dans une structure hiérarchique de gestion réseau, souvent utilisée pour accéder à des données via SNMP (Simple Network Management Protocol).

9. Ajout du contenu suivant au fichier `commands.cfg` :

```
# Definition de la commande check_lin_load (Charge CPU)
define command {
    command_name check_lin_load
    command_line $USER1$/check_snmp_load.pl -H $HOSTADDRESS$ -C $ARG1$ $ARG2$ -T $ARG3$ -w
$ARG4$ -c $ARG5$
}

# Definition de la commande check_snmp_storage (Espace Disque)
define command{
    command_name check_snmp_storage
    command_line $USER1$/check_snmp_storage.pl -H $HOSTADDRESS$ -C $ARG1$ $ARG2$ -m $ARG3$ -w
$ARG4$ -c $ARG5$
}

# Definition de la commande check_snmp_mem (Memoire RAM)
define command{
    command_name check_snmp_mem
    command_line $USER1$/check_snmp_mem.pl -H $HOSTADDRESS$ -C $ARG1$ $ARG2$ $ARG3$ -w $ARG4$ -
c $ARG5$
}

# Définition de la commande pour la supervision du débits réseau
define command{
    command_name check_snmp_int
    command_line $USER1$/check_snmp_int.pl -H $HOSTADDRESS$ -C $ARG1$ $ARG2$ -n $ARG3$ -c $ARG4$
$ARG5$
}
```

9. A partir d'ici on enregistre et quitte l'éditeur de texte.

10. On redémarre Nagios pour appliquer les modifications de configuration :

```
systemctl restart nagios4
```

11. On s'est assuré que Nagios est démarré automatiquement au démarrage du système :

```
systemctl enable nagios4
```

12. On a configuré la gestion des notifications NTFY avec la commande ci-dessous (fichier `/etc/nagios4/objects/commands.cfg`):

Cette commande permet d'envoyer des messages HTTP en mode push grâce à une API REST

```
# 'notify-service-by-ntfy' command definition
define command{
    command_name  notify-service-by-ntfy
    command_line  curl -H "Title: ** $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is
$SERVICESTATE$ **" -d "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService:
$SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time:
$LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$\n $NOTIFICATIONCOMMENT$\n" https://ntfy.lyc-
lecastel.fr/nagios
}
```

Maintenant, Nagios est bien installé et configuré pour superviser les hôtes qui nous intéressent en utilisant SNMP pour surveiller la charge CPU, l'occupation mémoire, l'espace disque et le débit réseau et d'autres paramètres.

On peut accéder à l'interface web de Nagios en utilisant l'adresse IP de la machine ap31-mon dans un navigateur et en ajoutant "/nagios" à l'URL (par exemple, <http://ap31-mon/nagios> // <http://172.16.0.97/nagios>).

Voici l'interfaces des hôtes supervisés & map des hôtes

Limit Results:

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
ap31-prod	Charge CPU	OK	11-16-2023 07:57:56	27d 18h 53m 17s	1/3	Load (CPUs: 2) : 0.60 0.38 0.25 : OK
	Charge Memoire - RAM	OK	11-16-2023 08:00:11	27d 22h 51m 2s	1/3	Ram : 11%, Swap : 0% : OK
	Debit Reseau	OK	11-16-2023 08:02:25	26d 23h 0m 14s	1/3	eth0:UP:1 UP: OK
	Espace disque	OK	11-16-2023 08:04:39	27d 23h 0m 58s	1/3	/: 13% used (1056MB/7963MB) (<90%) : OK
	PING	UNKNOWN	11-16-2023 07:58:07	27d 23h 38m 52s	3/3	/bin/ping -4 -n -U -w 10 -c 5 172.16.0.82
	SSH	OK	11-16-2023 08:00:23	27d 23h 47m 47s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2 (protocol 2.0)
ap31-test	Charge CPU	OK	11-16-2023 08:02:37	27d 18h 50m 52s	1/3	Load (CPUs: 2) : 0.13 0.17 0.18 : OK
	Charge Memoire - RAM	OK	11-16-2023 08:04:52	27d 22h 46m 3s	1/3	Ram : 11%, Swap : 0% : OK
	Debit Reseau	OK	11-16-2023 07:58:23	26d 22h 57m 59s	1/3	eth0:UP:1 UP: OK
	Espace disque	OK	11-16-2023 08:00:38	27d 22h 59m 40s	1/3	/: 11% used (916MB/7963MB) (<90%) : OK
	PING	UNKNOWN	11-16-2023 08:02:51	27d 23h 37m 16s	3/3	/bin/ping -4 -n -U -w 10 -c 5 172.16.0.95
	SSH	OK	11-16-2023 08:05:06	34d 21h 11m 39s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2 (protocol 2.0)
ap31-wiki	WWW	OK	11-16-2023 07:58:34	34d 21h 9m 59s	1/3	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.000 second response time
	Charge CPU	OK	11-16-2023 08:00:49	27d 18h 48m 28s	1/3	Load (CPUs: 2) : 0.44 0.36 0.26 : OK
	Charge Memoire - RAM	OK	11-16-2023 08:03:05	27d 22h 8m 58s	1/3	Ram : 4%, Swap : 0% : OK
	Debit Reseau	OK	11-16-2023 08:05:20	26d 22h 55m 44s	1/3	eth0:UP:1 UP: OK
	Espace disque	OK	11-16-2023 07:58:48	27d 22h 7m 37s	1/3	/: 11% used (844MB/7963MB) (<90%) : OK
	PING	UNKNOWN	11-16-2023 08:01:03	27d 22h 16m 16s	3/3	/bin/ping -4 -n -U -w 10 -c 5 172.16.0.87
gwlabs	SSH	OK	11-16-2023 08:03:18	27d 22h 14m 56s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2 (protocol 2.0)
	WWW	OK	11-16-2023 08:05:33	27d 22h 13m 35s	1/3	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time
	PING	UNKNOWN	11-16-2023 07:59:01	27d 21h 56m 15s	3/3	/bin/ping -4 -n -U -w 10 -c 5 10.121.38.253
gwsio5	SSH	OK	11-16-2023 08:01:16	27d 21h 54m 28s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u1 (protocol 2.0)
	PING	UNKNOWN	11-16-2023 08:03:31	27d 21h 59m 40s	3/3	/bin/ping -4 -n -U -w 10 -c 5 10.121.38.35
localhost	SSH	OK	11-16-2023 08:05:39	1d 19h 22m 17s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u1 (protocol 2.0)
	Current Load	OK	11-16-2023 08:04:16	35d 0h 45m 34s	1/4	LOAD OK - total load average: 0.05, 0.13, 0.16
	Current Users	OK	11-16-2023 08:05:39	35d 0h 44m 59s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	11-16-2023 08:05:39	35d 0h 44m 22s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.001 second response time
	PING	UNKNOWN	11-16-2023 08:05:39	27d 23h 37m 44s	4/4	/bin/ping -4 -n -U -w 10 -c 5 127.0.0.1
	Root Partition	OK	11-16-2023 08:04:28	35d 0h 43m 7s	1/4	DISK OK - free space: / 6175MiB (81% inode=92%):
	SSH	OK	11-16-2023 08:05:39	35d 0h 42m 29s	1/4	SSH OK - OpenSSH_9.2p1 Debian-2 (protocol 2.0)
	Swap Usage	OK	11-16-2023 08:05:39	35d 0h 41m 52s	1/4	SWAP OK - 100% free (1024MB out of 1024MB)
px	Total Processes	OK	11-16-2023 08:05:39	35d 0h 41m 14s	1/4	PROCS OK: 0 processes with STATE = RSZDT
	PING	UNKNOWN	11-16-2023 07:59:42	27d 21h 27m 32s	3/3	/bin/ping -4 -n -U -w 10 -c 5 10.121.38.2
pxlab1	SSH	OK	11-16-2023 08:01:57	1d 17h 35m 59s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u1 (protocol 2.0)
	PING	UNKNOWN	11-16-2023 08:04:12	27d 21h 0m 48s	3/3	/bin/ping -4 -n -U -w 10 -c 5 172.16.0.10
pxlab2	SSH	OK	11-16-2023 08:05:39	27d 21h 6m 28s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u1 (protocol 2.0)
	PING	UNKNOWN	11-16-2023 07:59:56	27d 21h 6m 48s	3/3	/bin/ping -4 -n -U -w 10 -c 5 172.16.0.11
pxlab3	SSH	OK	11-16-2023 08:02:10	27d 21h 1m 39s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u1 (protocol 2.0)
	PING	UNKNOWN	11-16-2023 08:04:25	27d 21h 0m 48s	3/3	/bin/ping -4 -n -U -w 10 -c 5 172.16.0.12
	SSH	OK	11-16-2023 08:05:39	27d 21h 6m 14s	1/3	SSH OK - OpenSSH_9.2p1 Debian-2+deb12u1 (protocol 2.0)

Mission I2 - Supervision des serveurs ap31-prod et ap31-test

MAP :

