

TP4 : Interactions automatisées entre des processus de gestion d'infrastructure

Thomas Dumond & Ethan Huret

Lien du dépôt Git :

<https://github.com/EthanAndreas/ITILAutomatedInteractions>

Table des matières

Table des matières	2
I. Installation de l'agent fusioninventory.....	3
II. Utilisation de l'API de GLPI	4
III. Inventaire dynamique	4
IV. Supervision avec Nagios.....	5
V. Check Nagios	5
VI. SNMP	6
VII. Check SNMP dans Nagios.....	7

Toutes les commandes sont précisées dans le README.md du Git.

I. Installation de l'agent fusioninventory

L'installation de l'agent *fusioninventory* se fait via *ansible*, afin de l'installer de manière automatisée sur chaque machine.

Un fichier *hosts.ini* permet de déclarer les machines (pc1, pc2 et pc3) et un playbook ansible permet de lancer les tâches sur ces machines.

```
1 - name: Déploiement de l'agent FusionInventory
2 hosts: all
3 become: true
4 tasks:
5   - name: Mise à jour du système via apt update avec --allow-releaseinfo-change
6     command: apt update --allow-releaseinfo-change
7
8   - name: Installation du paquet FusionInventory-Agent
9     package:
10      name: fusioninventory-agent
11      state: present
12
13   - name: Configuration de l'agent FusionInventory
14     lineinfile:
15      path: /etc/fusioninventory/agent.cfg
16      regex: '^server'
17      line: 'server = http://192.168.57.98/glpi/plugins/fusioninventory/'
18
19   - name: Redémarrage de l'agent FusionInventory
20     command: fusioninventory-agent
```

Figure 1 - Playbook ansible de l'installation de fusioninventory (playbook.yaml)

De plus, un fichier *.ansible.cfg* est utilisé pour paramétrer ansible, spécialement pour exécuter les commandes en mode root.

```
1 [defaults]
2 host_key_checking = False
3 remote_user = root
4 inventory = /home/tprli/hosts
```

Figure 2 - Fichier de configuration ansible

Via l'API GLPI sur la page <http://192.168.57.98/glpi/front/computer.php>, on peut voir que les machines et leurs configurations ont été ajoutées.

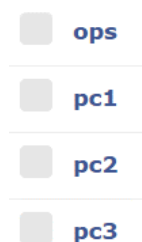


Figure 3 - Liste machines de l'API GLPI

II. Utilisation de l'API de GLPI

Il est possible d'obtenir l'inventaire via l'API *GLPI* de manière automatisée, cependant, cela nécessite une authentification.

- Dans un premier temps, il faut créer une API client via

<http://192.168.57.98/glpi/front/config.form.php>

Après l'avoir créée, on obtient un token d'application, défini au moment de la création ou bien généré automatiquement :

Application token (app_token)

i05sner75WB5vffRQXHAux

☐ Regenerate

Cet token d'application est une clé qui est nécessaire pour s'authentifier.

- Ensuite, il faut utiliser un utilisateur et son mot de passe (dans notre cas c'est l'utilisateur par défaut *glpi*).
- A l'aide d'un script bash, on envoie une requête à l'API en lui communiquant les informations nécessaires, et elle renvoie un token de session. Et, à l'aide de ce token de session, il est possible d'envoyer une requête à l'application session pour récupérer les données de l'inventaire.

```
1 user=glpi
2 password=glpi
3 authtoken=$(echo -n "$user:$password" | base64)
4 APPTOKEN='i05sner75WB5vffRQXHAuxnf0K5y$5tx1wx6x3L5'
5 APIURL='http://localhost/glpi/apirest.php'
6
7 response=$(curl -s -X GET \
8   -H 'Content-Type: application/json' -H "Authorization: Basic $authtoken" \
9   -H "App-Token: $APPTOKEN" "$APIURL/initSession")
10
11 sessiontoken=$(echo "$response" | grep -o "session_token":"[^"]*" | sed 's/"session_token"://')
12
13 if [ -n "$sessiontoken" ]; then
14   curl -s -X GET \
15     -H 'Content-Type: application/json' -H "Session-Token: $sessiontoken" \
16     -H "App-Token: $APPTOKEN" "$APIURL/Computer" | jq -r '.[.] | .name'
17 fi
```

Figure 4 - Script bash récupérant le nom des machines via l'API GLPI (static-inventory.sh)

Ce script nous permet de récupérer les noms des machines comme indiqué sur la Figure 3.

III. Inventaire dynamique

Désormais, on souhaite mettre l'inventaire obtenu au format ansible. Pour cela, le script bash a simplement été modifié pour retourner :

```
{
  "all": {
    "hosts": [
      "pc2",
      "pc1",
      "pc3"
    ]
  }
}
```

Figure 5 - Inventaire fourni par le script bash dynamic-inventory.sh

Il faut bien penser à exclure la machine *ops* qui est la machine gérant les autres et où se trouve le playbook ansible. L'inventaire peut ainsi être précisé comme *-i dynamic-inventory.sh*.

IV. Supervision avec Nagios

On souhaite maintenant surveiller les machines via *Nagios*. Ainsi, il faut définir les machines à surveiller dans sa configuration. Pour cela, on réutilise le contenu le premier script bash auquel on ajoute des commandes. On crée un fichier *pc.cfg* dans */etc/nagios4/objects/*, et on définit chaque machine dans ce fichier de la manière suivante :

```
1 define host
2 {
3     use                linux-server
4     host_name          pc1
5     check_interval 1
6 }
```





Figure 6 - Format des machines dans la configuration Nagios

Puis, le fichier de configuration est ajouté dans */etc/nagios4/nagios.cfg* :





```
1 cfg_file=/etc/nagios4/objects/pc.cfg
```

Figure 7 - Ajout du fichier de configuration des machines dans le fichier de configuration principal

Cette opération est réalisée uniquement sur *ops*. Ainsi, dans le playbook ansible, un second bloc concernant la configuration et le redémarrage de Nagios est fait pour l'host *ops*. On peut ainsi voir que les machines sont désormais présentes en tant qu'hosts sur l'API de Nagios.

localhost		UP
pc1		UP
pc2		UP
pc3		UP

Dans le cas où se connecte sur *pc1* et qu'on arrête le système. Nagios considère la machine comme *down*.

localhost		UP
pc1		DOWN
pc2		UP
pc3		UP

En redémarrant le conteneur, Nagios reconsidère *pc1* comme *up*.

V. Check Nagios

On souhaite vérifier via *SMTP* tourne sur la machine *ops*. Pour cela, on peut ajouter un plugin à Nagios.

- Il faut créer le fichier de configuration */etc/nagios-plugins/config/smtp.cfg*. Il faut indiquer la commande à réaliser pour le test, puis il faut définir le service autour du test. Dans ce cas-ci, on fait simplement un netcat sur le port 25 pour voir s'il y a du contenu.

```
1  define command {
2      command_name check_smtp
3      command_line /home/tparli/check_smtp
4  }
5
6  define service {
7      use generic-service
8      host_name localhost
9      service_description Check SMTP
10     check_command check_smtp
11     check_interval 1
12 }
```

- Comme pour l'ajout des pc en tant que hosts pour Nagios, il faut ajouter la fichier de configuration au fichier principal de configuration */etc/nagios4/nagios.cfg*.

VI. SNMP

Nous avons voulu configurer *SNMP* afin de vérifier l'état de certaines machines. Pour cela, il faut configurer *SNMP* sur chaque machine. Il faut installer le daemon *snmpd* et l'outil command-line *snmp*. Dans le fichier de configuration du daemon, il faut mettre toutes les interfaces à l'écoute et mettre la communauté *public* en *example*.

```
1  - name: Installation du daemon SNMP
2      package:
3          name: snmpd
4          state: present
5
6  - name: Installation du command-line SNMP
7      package:
8          name: snmp
9          state: present
10
11 - name: Mettre en écoute sur toutes les interfaces
12     lineinfile:
13         path: /etc/snmp/snmpd.conf
14         regexp: '^#?(\\s*)agentAddress'
15         line: 'agentAddress udp:161,udp6:[::1]:161'
16
17 - name: Modification de la communauté SNMP
18     lineinfile:
19         path: /etc/snmp/snmpd.conf
20         regexp: '^#?(\\s*)com2sec'
21         line: 'com2sec readonly default example'
22
23 - name: Redémarrage de SNMP
24     command: systemctl restart snmpd
```

Nous avons rencontré des problèmes lorsque nous avons voulu vérifier que le service était bien configuré. En effet, la variable *hrSystemProcesses* n'est pas définie ou non-reconnue. Ainsi, nous n'avons pas pu déterminer l'OID de *hrSystemProcesses*, ni l'interroger.

VII. Check SNMP dans Nagios

Etant les problèmes précédents, nous n'avons pas pu faire cette partie. Les lignes de cette partie sont commentées dans le playbook.