



TP Ansible Base (corrections)

Table des matières

Module 2 : Installation sur le serveur de gestion	2
Exercice 2.1 : Prise en main des VM de TP	2
Exercice 2.2 : Vérification de l'installation	2
Module 3 : Authentification par clef	5
Exercice 3.1 : Créer une bi-clefs	5
Exercice 3.2 : Tests d'authentification	5
Module 4 : Utilisation en ligne de commande	7
Module 5 : Les modules	8
Module 6 : Les playbooks	13

Objectifs

- Mettre en oeuvre Ansible ;
- Appliquer des changements de configuration sur un serveur ;
- Créer des playbooks Ansible.

Module 2 : Installation sur le serveur de gestion

Objectifs

- Vérifier l'installation d'ansible sur le serveur

Exercice 2.1 : Prise en main des VM de TP

Vous allez travailler sur 2 VM :

- La VM de gestion : **server-X**.
 - Vous installerez **ansible** sur ce serveur.
 - Adresse IP : 10.1.1.X
- La VM cliente : **client-X**.
 - Cette VM n'a aucun logiciel spécifique.
 - Adresse IP : 10.1.1.X

Pour accéder à ces serveurs, vous passerez par une adresse IP publique et une redirection de port vers votre serveur.

Exemple :

```
ssh -p 2222 aftec@109.238.4.96 # Acces au serveur
ssh -p 2223 aftec@109.238.4.96 # Acces au client
```

Le mot de passe est **@ns!bl\$4@ft\$c**



Installez ansible en suivant les instructions fournies à la page suivante : http://docs.ansible.com/ansible/latest/intro_installation.html

Uniquement sur le serveur de gestion :

```
$ sudo yum install -y epel-release # Pour avoir ansible en version 2.7
$ sudo yum install -y ansible
```

Exercice 2.2 : Vérification de l'installation

- Vérifier la connectivité entre les serveurs :

Se connecter avec l'utilisateur **aftec** sur le serveur, puis faire un **ping** vers le client :

```
[aftec] $ ping 10.1.1.11
```

- Vérifier qu'Ansible est installé sur le serveur

```
[aftec] $ rpm -qa ansible  
ansible-2.7.5-1.el7.noarch
```

- Vérifier la version d'Ansible :

```
[aftec] $ ansible --version  
ansible 2.7.5  
  config file = /etc/ansible/ansible.cfg  
  configured module search path = [u'/home/aftec/.ansible/plugins/modules',  
u'/usr/share/ansible/plugins/modules']  
  ansible python module location = /usr/lib/python2.7/site-packages/ansible  
  executable location = /usr/bin/ansible  
  python version = 2.7.5 (default, Jul 13 2018, 13:06:57) [GCC 4.8.5 20150623 (Red Hat  
4.8.5-28)]
```

- Visualiser le fichier de configuration :

```
$ less /etc/ansible/ansible.cfg
```

- Visualiser le fichier d'inventaire :

```
$ less /etc/ansible/hosts
```



Plus d'informations sur http://docs.ansible.com/ansible/latest/intro_inventory.html

- Créer deux entrées dans ce fichier inventaire : une votre serveur et une autre pour votre client.

```
[aftec] $ sudo vim /etc/ansible/hosts  
[ansiblesrv]  
10.1.1.10  
  
[ansiblecli]  
10.1.1.11
```

- Créer un groupe **tp-ansible** reprenant les deux entrées précédemment créées :

```
[aftec] $ sudo vim /etc/ansible/hosts  
[tp-ansible:children]  
ansiblesrv  
ansiblecli
```

Module 3 : Authentification par clef

Objectifs

- Déployer une clef privée/publique sur les clients

Exercice 3.1 : Créer une bi-clefs

Sur le serveur de gestion :

```
[aftec] $ ssh-keygen
```

Copier la clef publique sur les 2 VM :

```
[aftec] $ ssh-copy-id aftec@10.1.1.10  
[aftec] $ ssh-copy-id aftec@10.1.1.11
```

Tester une connexion SSH vers chacun des serveurs.



Aucun mot de passe ne doit être demandé :

```
[aftec] $ ssh 10.1.1.10  
Last login: Wed Jan 3 15:23:00 2018 from 10.1.1.10
```

```
[aftec] $ ssh 10.1.1.11  
Last login: Wed Jan 3 15:23:00 2018 from 10.1.1.10
```

Exercice 3.2 : Tests d'authentification

- Utiliser le module **ping** pour tester le bon fonctionnement de votre plateforme :

```
[aftec] $ ansible tp-ansible -m ping
10.1.1.11 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
10.1.1.10 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

- Utiliser le module **shell** pour récupérer le **uptime** de vos VM :

```
[aftec] $ ansible tp-ansible -m shell -a "uptime"
10.1.1.11 | CHANGED | rc=0 >>
22:11:31 up 39 min, 1 user, load average: 0.00, 0.01, 0.03

10.1.1.10 | CHANGED | rc=0 >>
22:11:31 up 41 min, 2 users, load average: 0.13, 0.04, 0.05
```


Module 4 : Utilisation en ligne de commande

Objectifs

- Utiliser ansible en lignes de commandes



Plus d'informations sur <http://docs.ansible.com/ansible/latest/ansible.html>

- Lister les hôtes appartenants au groupe **tp-ansible** :

```
[aftec] $ ansible tp-ansible --list-hosts
hosts (2):
 10.1.1.10
 10.1.1.11
```

- Afficher les **facts** pour la VM **ansiblecli** :

```
[aftec] $ ansible ansiblecli -m setup | less
10.1.1.11 | SUCCESS => {
  "ansible_facts": {
    "ansible_all_ipv4_addresses": [
      "10.1.1.11"
    ],
    "ansible_all_ipv6_addresses": [],
    "ansible_apparmor": {
      "status": "disabled"
    },
    "ansible_architecture": "x86_64",
    "ansible_bios_date": "03/23/2017",
    "ansible_bios_version": "4.4.1-xs137600",
    "ansible_cmdline": {
      "BOOT_IMAGE": "/vmlinuz-3.10.0-327.28.3.el7.x86_64",
      "LANG": "en_US.UTF-8",
      "crashkernel": "auto",
      "quiet": true,
      "rd.lvm.lv": "vg01/root",
      "rhgb": true,
      "ro": true,
      "root": "/dev/mapper/vg01-root",
      "vconsole.font": "latarcyrheb-sun16",
      "vconsole.keymap": "fr"
    },
  },
  ...
```

Module 5 : Les modules

Objectifs

- Utiliser les principaux modules Ansible
- Chercher dans la documentation Ansible



Plus d'informations sur http://docs.ansible.com/ansible/modules_by_category.html

En utilisant Ansible en ligne de commande, vous allez réaliser les actions suivantes :

- Créer les groupes **rennes**, **lille** et **paris**
- Créer un utilisateur **core**
- Modifier l'utilisateur **core** pour qu'il ait l'UID 10000
- Modifier l'utilisateur **core** pour qu'il soit dans le groupe **rennes**
- Installer le logiciel **tree**
- Stopper le service **crond**
- Désactiver le service **atd**
- créer un fichier vide **/tmp/test** avec les droits 644
- Mettre à jour le système sur la VM cliente
- Redémarrer la VM cliente (une petite recherche s'impose !)



Utilisez les modules appropriés plutôt que le module shell. Pour rappel, les modules disponibles sont listés à cette adresse : http://docs.ansible.com/ansible/modules_by_category.html



Vérifier sur le client que les actions sont bien effectuées.

- Créer les groupes **rennes**, **lille** et **paris**

```
$ ansible tp-ansible --become --ask-become-pass -m group -a "name=rennes"
SUDO password:
10.1.1.11 | CHANGED => {
    "changed": true,
    "gid": 1001,
    "name": "rennes",
    "state": "present",
    "system": false
}
10.1.1.10 | CHANGED => {
    "changed": true,
    "gid": 1001,
    "name": "rennes",
    "state": "present",
    "system": false
}

# grep rennes /etc/group
rennes:x:1001
```

```
$ ansible tp-ansible --become --ask-become-pass -m group -a "name=lille gid=1005"
SUDO password:
10.1.1.11 | CHANGED => {
    "changed": true,
    "gid": 1005,
    "name": "lille",
    "state": "present",
    "system": false
}
10.1.1.10 | CHANGED => {
    "changed": true,
    "gid": 1005,
    "name": "lille",
    "state": "present",
    "system": false
}
```

```
$ ansible tp-ansible --become --ask-become-pass -m group -a "name=paris gid=1010"
SUDO password:
10.1.1.11 | CHANGED => {
  "changed": true,
  "gid": 1010,
  "name": "paris",
  "state": "present",
  "system": false
}
10.1.1.10 | CHANGED => {
  "changed": true,
  "gid": 1010,
  "name": "paris",
  "state": "present",
  "system": false
}
```

- Créer un utilisateur **core**

```
$ ansible tp-ansible --become --ask-become-pass -m user -a "name=core"
SUDO password:
10.1.1.11 | CHANGED => {
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1011,
  "home": "/home/core",
  "name": "core",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1001
}
10.1.1.10 | CHANGED => {
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1011,
  "home": "/home/core",
  "name": "core",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1001
}
...
```

- Modifier l'utilisateur **core** pour qu'il ait l'UID 10000

```
$ ansible tp-ansible --become --ask-become-pass -m user -a "name=core uid=10000"
SUDO password:
10.1.1.11 | CHANGED => {
  "append": false,
  "changed": true,
  "comment": "",
  "group": 1011,
  "home": "/home/core",
  "move_home": false,
  "name": "core",
  "shell": "/bin/bash",
  "state": "present",
  "uid": 10000
}
10.1.1.10 | CHANGED => {
  "append": false,
  "changed": true,
  "comment": "",
  "group": 1011,
  "home": "/home/core",
  "move_home": false,
  "name": "core",
  "shell": "/bin/bash",
  "state": "present",
  "uid": 10000
}
```

- Modifier l'utilisateur **core** pour qu'il soit invité dans le groupe **rennes**

```
$ ansible tp-ansible --become --ask-become-pass -m user -a "name=core groups=rennes"
10.1.1.11 | SUCCESS => {
  "changed": true,
  ...
  "name": "core",
  "groups": "rennes",
  ...
  "uid": 10000
}
...
```

- Installer le logiciel **tree**

```
$ ansible tp-ansible --become --ask-become-pass -m yum -a "name=tree"
```

- Stopper le service **crond**

```
$ ansible tp-ansible --become --ask-become-pass -m systemd -a "name=crond state=stopped"
```

- Désactiver le service **auditd**

```
$ ansible tp-ansible --become --ask-become-pass -m systemd -a "name=auditd enabled=no"
```

- créer un fichier vide **/tmp/test** avec les droits 644

```
$ ansible tp-ansible --become --ask-become-pass -m copy -a "content='' dest=/tmp/test force=no mode=644"
```

- Mettre à jour le système sur la VM cliente

```
$ ansible ansiblecli --become --ask-become-pass -m yum -a "name=* state=latest"
```

- Redémarrer la VM cliente

```
$ ansible ansiblecli --become --ask-become-pass -m shell -a "reboot"
```

Module 6 : Les playbooks

Objectifs

- Ecrire ses premiers playbooks



Plus d'informations sur <http://docs.ansible.com/ansible/latest/playbooks.html> et sur <http://docs.ansible.com/ansible/latest/ansible-playbook.html>.



Modifier la configuration de **sudo** avec la commande **visudo** pour que le mot de passe de votre utilisateur ne soit plus demandé lors de l'escalade de privilège.

- Ecrire un playbook simple permettant de mettre à jour une VM CentOS 7 et de la redémarrer

Le playbook patchmanagement.yml

```
---
- hosts: ansiblecli
  remote_user: aftec
  become: true

  tasks:
  - name: update all packages
    yum: name=* state=latest
  - name: reboot the vm
    shell: sleep 2 && shutdown -r now 'Ansible reboot triggered'
```

```
$ ansible-playbook patchmanagement.yml
```

- Ecrire un playbook simple permettant de déployer le compte **supervision** (UID: 5000, groupe: rennes)

Le playbook supervision.yml

```
---
- hosts: ansiblecli
  remote_user: afterc
  become: true

  tasks:
  - name: creer le compte supervision
    user:
      name: supervision
      uid: 5000
      group: rennes
```

```
$ ansible-playbook supervision.yml
```

- Ecrire un playbook permettant de déployer apache/php/mariadb.

Le playbook lamp.yml

```
---
- hosts: ansiblecli
  remote_user: afterc
  become: true

  tasks:
  - name: ensure apache and php are at the latest version
    yum:
      name: httpd,php,php-mysqli
      state: latest
  - name: ensure httpd is started
    systemd:
      name: httpd
      state: started
  - name: ensure mariadb-server is at the latest version
    yum:
      name: mariadb-server
      state: latest
  - name: ensure mariadb is started
    systemd:
      name: mariadb
      state: started
```

```
$ ansible-playbook lamp.yml
```