

TP2 – Gestion des hôtes cibles - correctement

Brahim Hamdi

Commandes ad-hoc

1. En utilisant le module « setup », collectez des informations détaillées sur les cibles (gathering facts)

ansible all -m setup

2. Installez les paquet net-tools, mariadb-server et nginx sur tous les hôtes cibles

ansible host1 --become -m yum -a "name=net-tools state=present"

ansible host1 --become -m yum -a "name=mariadb-client state=present"

ansible host1 --become -m yum -a "name=nginx state=present"

ansible host2 --become -m yum -a "name=net-tools state=present"

ansible host2 --become -m yum -a "name=mariadb-client state=present"

ansible host2 --become -m yum -a "name=nginx state=present"

ou

ansible all --become -m package -a "name=net-tools state=present"

ansible all --become -m package -a "name=mariadb-client state=present"

ansible all --become -m package -a "name=nginx state=present"

ou

ansible all --become -m package -a "name=net-tools,mariadb-server,nginx"

- Quels sont les modules utilisés ?

- apt (host1) et yum (host2)

- package (all)

- Vérifiez que les paquets ont été bien installés sur tous les hôtes

Ok

3. Arrêtez les services mariadb et nginx

ansible all --become -m service -a "name=mariadb state=stopped"

ansible all --become -m service -a "name=nginx state=stopped"

- Supprimez ces paquets.

ansible all --become -m package -a "name=nginx,mariadb-server,mariadb-server-10.3 state=absent"

- Supprimez le paquet git de votre hôte local (ansible)

ansible localhost -c local --become -m package -a "name=git state=absent"

- Vérifiez que tous ces paquets ont été bien supprimés.

```
ssh 192.168.201.11 "dpkg -l | grep nginx"
```

```
ssh 192.168.201.12 "rpm -qa | grep nginx"
```

4. Refaire les mêmes tâches en utilisant un fichier inventaire que vous créez dans le répertoire courant.

```
ansible all -i inventaire --become -m package -a "name=net-tools,mariadb-server,nginx"
```

...

Premier playbook

5. Dans le fichier inventory, affectez host1 au groupe web et host2 au groupe db

```
[web]
```

```
host1 ansible_host=192.168.201.11 ansible_user=vagrant ansible_ssh_private_key_file=~/.ssh/id_rsa
```

```
[db]
```

```
host2 ansible_host=192.168.201.12 ansible_user=vagrant ansible_ssh_private_key_file=~/.ssh/id_rsa
```

6. Créez un playbook nommé playbook1.yml

```
vim playbook.yml
```

- Dans ce playbook nous allons définir les tâches suivantes :

- Installer nginx sur host1 (groupe web)

- Installer mariadb-server sur host2 (groupe db)

```
- name: installer nginx sur les serveurs web
```

```
hosts: web
```

```
gather_facts: false
```

```
become: yes
```

```
tasks:
```

- name: Installer nginx

```
package:
```

```
name: nginx
```

```
state: latest
```

```
- name: installer mariadb sur les serveurs de base de données
```

```
hosts: db
```

```
gather_facts: false
```

```
become: yes
```

```
tasks:
```

- name: Installer mariadb-server

```
package:
```

```
name: mariadb-server
```

```
state: latest
```

- Appliquez playbook1.

```
ansible-playbook -i inventaire playbook1.yml
```

- Combien de tâches sont exécutées sur chaque hôte ?

1 tâche sur chaque groupe

- Les tâches sont exécutées correctement ?

Oui, ok=1

- Y a-t-il des changements ?

Oui

- Vérifiez le fonctionnement des services web et base de données.
- Que fait la tâche « Gathering Facts » ? Comment enlever cette tâche ?

Gathering facts : collecte des faits (une tâche supplémentaire)

Ajouter la ligne suivante :

gather_facts: false

- Ré-appliquez le même playbook.

- Que remarquez-vous ?

Pas de changements

7. Ajoutez un handler qui démarre le service mariadb. Il est exécuté suite à l'installation de mariadb-server.

- name: installer nginx sur les serveurs web

hosts: web

gather_facts: false

become: yes

tasks:

- name: Installer nginx

package:

name: nginx

state: latest

- name: installer mariadb sur les serveurs de base de données

hosts: db

gather_facts: false

become: yes

tasks:

- name: Installer mariadb-server

package:

name: mariadb-server

state: latest

notify: start mariadb

handlers:

- name: start mariadb

service:

name: mariadb

state: restarted

- Exécutez le playbook1. Le handler a-t-il exécuté ? Pourquoi ?

Non, parce que y a pas de changement ; mariadb est déjà installé

- Désinstallez mariadb-server, puis exécutez de nouveau playbook1. Le handler a t-il exécuté cette fois ?

Cette fois oui, parce qu'il a un changement.

Variables et templates

8. Ajoutez les 2 tâches suivantes à playbook1 :

- La première tâche cherche les fichiers « *.html » dans /usr/share/nginx/
- La deuxième copie les fichiers trouvés en ajoutant l'extension « .orig » dans le même répertoire.

- name: installer nginx sur les serveurs web

hosts: web

gather_facts: false

become: yes

tasks:

- name: Installer nginx

package:

name: nginx

state: latest

- name: Recherche de fichiers html

command: find /usr/share/nginx -iname ".html"*

register: find_output

- name: sauvegarder les html

copy:

src: "{{ item }}"

dest: "{{ item }}.orig"

remote_src: yes

with_items:

- "{{ find_output.stdout_lines }}"

- name: installer mariadb sur les serveurs de base de données

hosts: db

gather_facts: false

become: yes

tasks:

- name: Installer mariadb-server

package:

name: mariadb-server

state: latest

notify: start mariadb

handlers:

- name: start mariadb

service:

name: mariadb

state: restarted

- Appliquer le playbook
ansible-playbook -i inventaire playbook1.yml
- Vérifiez la présence de « index.html.orig »
ssh 192.168.201.11 "ls /usr/share/nginx/html"

9. Ajoutez une liste qui contient les informations suivantes :

- le titre : Formation Ansible
- votre nom
- votre email
- votre profession.
- Ajouter une tâche qui édite le fichier */var/www/html/index.html* de groupe web en ajoutant le contenu de la liste.
- Appliquer le playbook, et vérifier le contenu du fichier */var/www/html/index.html* sous host1.
- En utilisant le navigateur, tester la page d'accueil de host1.

hosts: web

gather_facts: false

become: yes

vars:

liste:

- **Formation Ansible**
- **Brahim Hamdi**
- **brahim.hamdi.consult@gmail.com**
- **formateur**

tasks:

- *name: Installer nginx*

package:

name: nginx

state: latest

- *name: Recherche de fichiers html*

command: find /usr/share/nginx -iname ".html"*

register: find_output

- *name: sauvegarder les html*

copy:

src: "{{ item }}"

dest: "{{ item }}.orig"

remote_src: yes

with_items:

- *"{{ find_output.stdout_lines }}"*

- *name: Editer index.html*

copy:

content: "{{ liste }}"

dest: /var/www/html/index.html

notify: reload nginx

handlers:

- name: reload nginx

service:

name: nginx

state: reloaded

- name: installer mariadb sur les serveurs de base de données

hosts: db

gather_facts: false

become: yes

tasks:

- name: Installer mariadb-server

package:

name: mariadb-server

state: latest

notify: start mariadb

handlers:

- name: start mariadb

service:

name: mariadb

state: restarted

10. Refaire le même travail en utilisant un modèle (template) jinja2 (fichier index.html.j2)

index.html.j2 :

Parcours DevOps

{% for item in liste %}

**{{ item }}
**

{% endfor %}

**
jeudi 5 août 2021**

Playbook1.yml :

- name: installer nginx sur les serveurs web

hosts: web

gather_facts: false

become: yes

vars:

liste:

- Formation Ansible

- Brahim Hamdi

- brahim.hamdi.consult@gmail.com

- formateur

tasks:

- name: Installer nginx

package:

name: nginx

```

    state: latest
- name: Recherche de fichiers html
  command: find /usr/share/nginx -iname "*.html"
  register: find_output
- name: sauvegarder les html
  copy:
    src: "{{ item }}"
    dest: "{{ item }}.orig"
    remote_src: yes
  with_items:
    - "{{ find_output.stdout_lines }}"
- name: Editer index.html
  template:
    src: index.html.j2
    dest: /var/www/html/index.html
  notify: reload nginx

```

handlers:

```

- name: reload nginx
  service:
    name: nginx
    state: reloaded

```

- name: installer mariadb sur les serveurs de base de données

hosts: db

gather_facts: false

become: yes

tasks:

```

- name: Installer mariadb-server
  package:
    name: mariadb-server
    state: latest
  notify: start mariadb

```

handlers:

```

- name: start mariadb
  service:
    name: mariadb
    state: restarted

```

Conditions et boucles

11. En utilisant une boucle, ajouter l'installation des paquets mariadb-client, php-mbstring, php-fpm, php-xml, php-intl et php-mysql à la tâche d'installation de nginx.

....

vars:

liste:

- Formation Ansible
- Brahim Hamdi

```
- brahim.hamdi.consult@gmail.com
- formateur
package: ['nginx','mariadb-client','php-mbstring','php-fpm','php-xml','php-intl','php-mysql']
```

tasks:

```
- name: Installer des paquets
```

```
package:
```

```
name: "{{ package }}"
```

```
state: latest
```

....

- Y a-t-il une autre méthode pour installer tous ces paquets ?

...

tasks:

```
- name: Installer des paquets
```

```
package:
```

```
name: nginx,mariadb-client,php-mbstring,php-fpm,php-xml,php-intl,php-mysql
```

```
state: latest
```

...

12. Fusionnez les 2 plays en ajoutant une condition qui indique la distribution linux utilisée.

Utilisez la notion de blocks.

- Appliquez le playbook. Quelle opération doit être faite avant d'exécuter les tâches (vérifier les conditions) ?
- Activez la collecte des faits (Gathering facts) dans le playbook, puis ré-appliquez le.

```
- name: installer nginx sur les serveurs web
```

```
hosts: all
```

```
gather_facts: true
```

```
become: yes
```

```
vars:
```

```
liste:
```

```
- Formation Ansible
```

```
- Brahim Hamdi
```

```
- brahim.hamdi.consult@gmail.com
```

```
- formateur
```

```
package: ['nginx','mariadb-client','php-mbstring','php-fpm','php-xml','php-intl','php-mysql']
```

tasks:

```
- name: Installer des paquets
```

```
block:
```

```
- package:
```

```
name: "{{ package }}"
```

```
state: latest
```

```
- command: find /usr/share/nginx -iname "*.html"
```

```
register: find_output
```

```
- copy:
```



```

    src: "{{ item }}"
    dest: "{{ item }}.orig"
    remote_src: yes
with_items:
  - "{{ find_output.stdout_lines }}"
- template:
    src: index.html.j2
    dest: /var/www/html/index.html
  notify: reload nginx

```

when: ansible_distribution=='Ubuntu'

```

- name: installer mariadb sur les serveurs de base de données
  block:
    - package:
        name: mariadb-server
        state: latest
      notify: start mariadb

```

when: ansible_distribution=='CentOS'

handlers:

```

- name: reload nginx
  service:
    name: nginx
    state: reloaded
- name: reload nginx
  service:
    name: nginx
    state: reloaded

```

Les étiquettes

13. Ajoutez les étiquettes suivantes :

- paquets (tâche d'installation des paquets sous hôte1)
- template_jinja2 (tâche de template)

...

```

    state: latest
tags: paquets
- command: find /usr/share/nginx -iname "*.html"
  register: find_output
- copy:
    src: "{{ item }}"
    dest: "{{ item }}.orig"
    remote_src: yes
with_items:
  - "{{ find_output.stdout_lines }}"

```

- *template:*
 src: index.html.j2
 dest: /var/www/html/index.html
tags: template_jinja2
notify: reload nginx

...

- Listez les étiquettes utilisées dans le playbook
ansible-playbook --list-tags playbook1.yml
- Faire une modification dans le template index.html.j2 et appliquez uniquement cette tâche.
ansible-playbook -i inventaire --tags='template_jinja2' playbook1.yml
- Appliquez tout le playbook sauf l'installation des paquets sous hôte1.
ansible-playbook -i inventaire --tags='paquets' playbook1.yml