

Machines Clientes Debian

--> SSH

Permit root login à yes

```
root@debian:~# grep ^PermitRoot /etc/ssh/sshd_config
PermitRootLogin yes
```

Redemarrer SSHD

```
root@debian:~# systemctl restart sshd
```

--> Dépôt Internet

Mette à jour sources.list en commentant la ligne du cdrom

```
root@debian:~# more /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 9.1.0 _Stretch_ - Official i386 DVD Binary-1 20170722-
12:45]/ stretch contrib main
# deb cdrom:[Debian GNU/Linux 9.1.0 _Stretch_ - Official i386 DVD Binary-1 20170722-
12:45]/ stretch contrib main
deb http://ftp.fr.debian.org/debian/ stretch main
deb-src http://ftp.fr.debian.org/debian/ stretch main
deb http://security.debian.org/debian-security stretch/updates main contrib
deb-src http://security.debian.org/debian-security stretch/updates main contrib
# stretch-updates, previously known as 'volatile'
deb http://ftp.fr.debian.org/debian/ stretch-updates main contrib
deb-src http://ftp.fr.debian.org/debian/ stretch-updates main contrib
```

--> Installer sudo

```
root@debian:~# apt-get install sudo
```

EXERCICES ANSIBLE

Installation Ansible

Exercice 1

Installation Ansible.

```
[root@mars ~]# yum install -y epel-release
[root@mars ~]# yum install -y ansible
```

Vérification.

```
[root@mars ~]# ansible --version
ansible 2.5.2
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.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, Aug 4 2017, 00:39:18) [GCC 4.8.5 20150623 (Red Hat 4.8.5-16)]
```

Mise à jour du fichier /etc/hosts.

```
[root@mars LABS]# more /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.14 client11
192.168.1.15 client12
192.168.1.17 debian11
192.168.1.27 debian12
```

Exercice 2

Génération de la clef SSH.

```
[root@mars ~]# ssh-keygen -t rsa
```

Envoi des clefs sur les serveurs.

```
[root@mars ~]# ssh-copy-id -i /root/.ssh/id_rsa.pub 192.168.1.14
[root@mars ~]# ssh-copy-id -i /root/.ssh/id_rsa.pub 192.168.1.15
[root@mars ~]# ssh-copy-id -i /root/.ssh/id_rsa.pub 192.168.1.17
[root@mars ~]# ssh-copy-id -i /root/.ssh/id_rsa.pub 192.168.1.27
```

Se connecter sur les serveurs avec leur nom de machine.

```
[root@mars ~]# ssh debian11
[root@mars ~]# ssh debian12
[root@mars ~]# ssh client11
[root@mars ~]# ssh client12
```

Manip 2 : Premiers pas avec Ansible

Exercice 1

Mise à jour du fichier /etc/ansible/hosts

```
[root@mars ~]# more /etc/ansible/hosts
[clients]
client11
client12

[debiens]
debian11
debian12
```

Test avec le module ping. Tout doit fonctionner

```
[root@mars ~]# ansible -m ping all
```

Exercice 2

Copie du fichier d'inventaire

```
[root@mars ~]# cp /etc/ansible/hosts /etc/ansible/inventaire.inv
```

Modification du fichier d'inventaire

```
[root@mars ~]# more /etc/ansible/inventaire.inv
[all:vars]
ansible_user=root

[clients]
client11
client12

[debiens]
debian11 nom=stretch
debian12 nom=jessie

[domaine1]
client13 domain=cloud.formation.fr
client14
debian13

[domaine2]
poste1 ansible_user=theo
poste2
poste3

[Windows]
Win10
Win8

[linux:children]
domaine1
domaine2

[linux:vars]
domain=formation.fr
```

Lister les hotes de l'inventaire.

```
[root@mars ~]# ansible all -i /etc/ansible/inventaire.inv --list-hosts
hosts (12):
  client13
  client14
  debian13
  debian11
  debian12
  Win10
  Win8
  poste1
  poste2
  poste3
  client11
  client12
```

Afficher la valeur de la variable domain.

```
[root@mars ansible]# ansible all -i inventaire.inv -m debug -a
"var=domain" --one-line
client13 | SUCCESS => {    "changed": false,    "domain": "cloud.formation.fr"}
debian12 | SUCCESS => {    "changed": false,    "domain": "VARIABLE IS NOT DEFINED!"}
debian11 | SUCCESS => {    "changed": false,    "domain": "VARIABLE IS NOT DEFINED!"}
client14 | SUCCESS => {    "changed": false,    "domain": "formation.fr"}
debian13 | SUCCESS => {    "changed": false,    "domain": "formation.fr"}
Win10 | SUCCESS => {    "changed": false,    "domain": "VARIABLE IS NOT DEFINED!"}
poste1 | SUCCESS => {    "changed": false,    "domain": "formation.fr"}
Win8 | SUCCESS => {    "changed": false,    "domain": "VARIABLE IS NOT DEFINED!"}
poste3 | SUCCESS => {    "changed": false,    "domain": "formation.fr"}
poste2 | SUCCESS => {    "changed": false,    "domain": "formation.fr"}
client11 | SUCCESS => {    "changed": false,    "domain": "VARIABLE IS NOT DEFINED!"}
client12 | SUCCESS => {    "changed": false,    "domain": "VARIABLE IS NOT DEFINED!"}
```

Afficher la valeur de la variable ansible_user.

```
[root@mars ansible]# ansible all -i inventaire.inv -m debug -a
"var=ansible_user" --one-line
poste1 | SUCCESS => {    "ansible_user": "theo",    "changed": false}
poste3 | SUCCESS => {    "ansible_user": "root",    "changed": false}
debian11 | SUCCESS => {    "ansible_user": "root",    "changed": false}
poste2 | SUCCESS => {    "ansible_user": "root",    "changed": false}
debian12 | SUCCESS => {    "ansible_user": "root",    "changed": false}
Win10 | SUCCESS => {    "ansible_user": "root",    "changed": false}
client14 | SUCCESS => {    "ansible_user": "root",    "changed": false}
debian13 | SUCCESS => {    "ansible_user": "root",    "changed": false}
Win8 | SUCCESS => {    "ansible_user": "root",    "changed": false}
client11 | SUCCESS => {    "ansible_user": "root",    "changed": false}
client13 | SUCCESS => {    "ansible_user": "root",    "changed": false}
client12 | SUCCESS => {    "ansible_user": "root",    "changed": false}
```

Afficher la valeur de la variable nom.

```
[root@mars ansible]# ansible all -i inventaire.inv -m debug -a "var=nom"
--one-line
poste1 | SUCCESS => {    "changed": false,    "nom": "VARIABLE IS NOT DEFINED!"}
Win10 | SUCCESS => {    "changed": false,    "nom": "VARIABLE IS NOT DEFINED!"}
debian12 | SUCCESS => {    "changed": false,    "nom": "jessie"}
...
client13 | SUCCESS => {    "changed": false,    "nom": "VARIABLE IS NOT DEFINED!"}
client14 | SUCCESS => {    "changed": false,    "nom": "VARIABLE IS NOT DEFINED!"}
debian13 | SUCCESS => {    "changed": false,    "nom": "VARIABLE IS NOT DEFINED!"}
```

Affichage sous forme de graphique.

```
[root@mars ~]# ansible-inventory all -i /etc/ansible/inventaire.inv
--graph
@all:
  |--@Windows:
  |   |--Win10
  |   |--Win8
  |--@clients:
  |   |--client11
  |   |--client12
  |--@debians:
  |   |--debian11
  |   |--debian12
  |--@linux:
  |   |--@domaine1:
  |   |   |--client13
  |   |   |--client14
  |   |   |--debian13
  |   |--@domaine2:
  |   |   |--poste1
  |   |   |--poste2
  |   |   |--poste3
  |--@ungrouped:
```

Affichage des variables.

```
[root@mars ~]# ansible-inventory all -i /etc/ansible/inventaire.inv --
graph --vars
@all:
  |--@Windows:
  |   |--Win10
  |   |--Win8
  |--@clients:
  |   |--client11
  |   |--client12
  |--@debians:
  |   |--debian11
  |   |   |--{nom = stretch}
  |   |--debian12
  |   |   |--{nom = jessie}
  |--@linux:
  |   |--@domaine1:
  |   |   |--client13
  |   |   |   |--{domain = cloud.formation.fr}
  |   |   |--client14
  |   |   |--debian13
  |   |--@domaine2:
  |   |   |--poste1
  |   |   |   |--{ansible_user = theo}
  |   |   |--poste2
  |   |   |--poste3
  |   |--{domain = formation.fr}
  |--@ungrouped:
  |--{ansible_user = root}
```

Affichage plus détaillée sous forme de liste

```
[root@mars ~]# ansible-inventory all -i /etc/ansible/inventaire.inv
--list
{
  "Windows": {
    "hosts": [
      "Win10",
      "Win8"
    ]
  },
  "_meta": {
    "hostvars": {
      "Win10": {
        "ansible_user": "root"
      },
      "Win8": {
        "ansible_user": "root"
      },
      "client11": {
        "ansible_user": "root"
      }
    }
  }
}
...
```

Les commandes Ansible

Exercice 1

Exécuter la commande suivante sur le groupe clients : free -m

```
[root@mars ~]# ansible clients -a 'free -m'
client11 | SUCCESS | rc=0 >>
      total      used      free      shared  buff/cache   available
Mem:      488      106      259         4        123        344
Swap:     1023         0      1023

client12 | SUCCESS | rc=0 >>
      total      used      free      shared  buff/cache   available
Mem:      488      109      259         4        119        340
Swap:     1023         0      1023
```

Remarque : l'option h c'est plus convivial.

```
[root@mars ~]# ansible clients -a 'free -h'
client11 | SUCCESS | rc=0 >>
      total      used      free      shared  buff/cache   available
Mem:     488M     106M     259M       4,3M       123M       344M
Swap:     1,0G         0B       1,0G

client12 | SUCCESS | rc=0 >>
      total      used      free      shared  buff/cache   available
Mem:     488M     109M     259M       4,3M       119M       340M
Swap:     1,0G         0B       1,0G
```

Exercice 2

Exécuter la commande suivante sur le groupe clients : date > /tmp/date.res

```
[root@mars ~]# ansible client -m shell -a 'date > /tmp/date.res'
client12 | SUCCESS | rc=0 >>

client11 | SUCCESS | rc=0 >>
```

Remarque : rc est le code de retour qui peut-être exploité par la suite.

vérifier le contenu de ce fichier après exécution.

```
[root@mars ~]# ssh client11 cat /tmp/date.res
mer. mai  9 13:42:03 CEST 2018
[root@mars ~]# ssh client12 cat /tmp/date.res
mer. mai  9 13:42:03 CEST 2018
```

Exécuter la commande suivante sur le groupe clients : date; sleep 5; date.

```
[root@mars ~]# ansible clients -m shell -a '( date ; sleep 5 ; date )'
client11 | SUCCESS | rc=0 >>
mer. mai  9 13:49:08 CEST 2018
mer. mai  9 13:49:13 CEST 2018

client12 | SUCCESS | rc=0 >>
mer. mai  9 13:49:08 CEST 2018
mer. mai  9 13:49:13 CEST 2018
```

Exercise 3

Pour le groupe clients : copier le fichier /etc/hosts sous /tmp/MACHINES avec les permissions 444, avec le propriétaire user1 et le groupe user1

```
[root@mars ~]# ansible clients -m copy -a "src=/etc/hosts
dest=/tmp/MACHINES mode=444 owner=user1 group=user1"
client11 | SUCCESS => {
  "changed": true,
  "checksum": "6546c2de97ae00b5246ad0708f6ef4369c48a72c",
  "dest": "/tmp/MACHINES",
  "gid": 1000,
  "group": "user1",
  "md5sum": "4502c378984d4e17785505dad0844839",
  "mode": "0444",
  "owner": "user1",
  "secontext": "unconfined_u:object_r:admin_home_t:s0",
  "size": 246,
  "src": "/root/.ansible/tmp/ansible-tmp-1525866835.02-187284774937133/source",
  "state": "file",
  "uid": 1000
}
client12 | SUCCESS => {
  "changed": true,
  "checksum": "6546c2de97ae00b5246ad0708f6ef4369c48a72c",
  "dest": "/tmp/MACHINES",
  "gid": 1000,
  "group": "user1",
  "md5sum": "4502c378984d4e17785505dad0844839",
  "mode": "0444",
  "owner": "user1",
  "secontext": "unconfined_u:object_r:admin_home_t:s0",
  "size": 246,
  "src": "/root/.ansible/tmp/ansible-tmp-1525866835.03-99225095518504/source",
  "state": "file",
  "uid": 1000
}
```

Vérification sur client11

```
[root@mars ~]# ssh client11 ls -l /tmp/MACHINES
-r--r--r--. 1 user1 user1 246  9 mai  13:53 /tmp/MACHINES
```

```
[root@mars ~]# ssh client11 cat /tmp/MACHINES
127.0.0.1    localhost localhost.localhost localhost4 localhost4.localhost4
::1         localhost localhost.localhost localhost6 localhost6.localhost6
192.168.1.14 client11
192.168.1.15 client12
192.168.1.17 debian11
192.168.1.27 debian12
```

Reexécuter la commande de copie (constater la différence des messages)

```
[root@mars ~]# ansible clients -m copy -a "src=/etc/hosts
dest=/tmp/MACHINES mode=444 owner=user1 group=user1"
client11 | SUCCESS => {
  "changed": false,
  "checksum": "6546c2de97ae00b5246ad0708f6ef4369c48a72c",
  "dest": "/tmp/MACHINES",
  "gid": 1000,
  "group": "user1",
  "mode": "0444",
  "owner": "user1",
```

```

    "path": "/tmp/MACHINES",
    "secontext": "unconfined_u:object_r:admin_home_t:s0",
    "size": 246,
    "state": "file",
    "uid": 1000
  }
}
client12 | SUCCESS => {
  "changed": false,
  "checksum": "6546c2de97ae00b5246ad0708f6ef4369c48a72c",
  "dest": "/tmp/MACHINES",
  "gid": 1000,
  "group": "user1",
  "mode": "0444",
  "owner": "user1",
  "path": "/tmp/MACHINES",
  "secontext": "unconfined_u:object_r:admin_home_t:s0",
  "size": 246,
  "state": "file",
  "uid": 1000
}

```

Pour le groupe clients : supprimer le fichier /tmp/MACHINES
 Attention: ne pas supprimer /etc/hosts !!!!!

```

[root@mars fictest]# ansible clients -m file -a "dest=/tmp/MACHINES
state=absent"
client11 | SUCCESS => {
  "changed": true,
  "path": "/tmp/MACHINES",
  "state": "absent"
}
client12 | SUCCESS => {
  "changed": true,
  "path": "/tmp/MACHINES",
  "state": "absent"
}

```

Quand la commande est réexécutée on constate que le changed est à false.

```

[root@mars fictest]# ansible clients -m file -a "dest=/tmp/MACHINES
state=absent"
client12 | SUCCESS => {
  "changed": false,
  "path": "/tmp/MACHINES",
  "state": "absent"
}
client11 | SUCCESS => {
  "changed": false,
  "path": "/tmp/MACHINES",
  "state": "absent"
}

```

Exercise 4

Pour le groupe clients : installer le package nmap

```
[root@mars fictest]# ansible clients -m yum -a "name=nmap state=present"
```

Apparaît en jaune. Si on reexécute la commande on constate que c'est en vert.

Vérifier si le package est présent.

```
[root@mars fictest]# ansible clients -m yum -a "list=nmap"
```

```
client12 | SUCCESS => {
  "changed": false,
  "results": [
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-7.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "7.el7",
      "repo": "base",
      "version": "6.40",
      "yumstate": "available"
    },
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-7.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "7.el7",
      "repo": "installed",
      "version": "6.40",
      "yumstate": "installed"
    }
  ]
}
client11 | SUCCESS => {
  "changed": false,
  "results": [
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-7.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "7.el7",
      "repo": "base",
      "version": "6.40",
      "yumstate": "available"
    },
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-7.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "7.el7",
      "repo": "installed",
      "version": "6.40",
      "yumstate": "installed"
    }
  ]
}
```

Pour le groupe clients : supprimer le package nmap

```
[root@mars fictest]# ansible clients -m yum -a "name=nmap state=absent"
```

En vérifiant on constate que le package est disponible sur les dépôts mais il n'est pas installé.

```
[root@mars fictest]# ansible clients -m yum -a "list=nmap"
client11 | SUCCESS => {
  "changed": false,
  "results": [
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-7.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "7.el7",
      "repo": "base",
      "version": "6.40",
      "yumstate": "available"
    }
  ]
}
client12 | SUCCESS => {
  "changed": false,
  "results": [
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-7.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "7.el7",
      "repo": "base",
      "version": "6.40",
      "yumstate": "available"
    }
  ]
}
```

Exercice 5

Pour le groupe clients : créer l'utilisateur user2 (uid 1234) :

- il appartiendra au groupe equipe (gid 1212)

```
[root@mars fictest]# ansible clients -m group -a "name=equipe gid=1212"
client12 | SUCCESS => {
  "changed": true,
  "gid": 1212,
  "name": "equipe",
  "state": "present",
  "system": false
}
client11 | SUCCESS => {
  "changed": true,
  "gid": 1212,
  "name": "equipe",
  "state": "present",
  "system": false
}
```

- son répertoire de connexion : /home/user2 (par défaut est créer)
- pas de mot de passe

```
[root@mars fictest]# ansible clients -m user -a "name=user2 uid=1234 group=equipe"
client12 | SUCCESS => {
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1212,
  "home": "/home/user2",
  "name": "user2",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1234
}
client11 | SUCCESS => {
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1212,
  "home": "/home/user2",
  "name": "user2",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1234
}
```

Pour debian

```
[root@mars fictest]# ansible debians -m group -a "name=equipe gid=1212"
debian11 | SUCCESS => {
  "changed": true,
  "gid": 1212,
  "name": "equipe",
  "state": "present",
  "system": false
}
debian12 | SUCCESS => {
  "changed": true,
  "gid": 1212,
  "name": "equipe",
  "state": "present",
  "system": false
}
```

```
[root@mars fictest]# ansible debians -m user -a "name=user2 uid=1234
group=equipe"
debian12 | SUCCESS => {
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1212,
  "home": "/home/user2",
  "name": "user2",
  "shell": "",
  "state": "present",
  "system": false,
  "uid": 1234
}
debian11 | SUCCESS => {
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1212,
  "home": "/home/user2",
  "name": "user2",
  "shell": "",
  "state": "present",
  "system": false,
  "uid": 1234
}
```

Exercise 6

Pour le groupe clients : redémarrer le service crond

```
[root@mars fictest]# ssh client11 pgrep -l crond
633 crond
[root@mars fictest]# ansible clients -m service -a "name=crond
state=restarted"
[root@mars fictest]# ssh client11 pgrep -l crond
2820 crond
```

Exercice 1

Créer un répertoire `apache_install`. Positionnez-vous sur ce répertoire.

```
[root@mars ~]# mkdir apache_install
[root@mars ~]# cd $_
```

Créer le playbook `apache_install.yml` qui doit réaliser les opérations suivantes sur les postes du groupe clients :

- Installation du package Apache (nom du package : `httpd`).
- Copie du fichier de configuration `httpd.conf` sous le répertoire `/etc/httpd/conf`.
- Installation de la page de garde du site :
copie du fichier `index_install.html` sous `/var/www/html/index.html`
propriétaire : `apache` groupe propriétaire : `apache`
- Redémarrer le service `apache`.

```
[root@mars apache_install]# cat apache_install.yml
---
# Creation de mon premier playbook
#####

# Liste des machines concernées

- hosts: clients

# Taches a effectuer

tasks:

  # Installation du package Apache

  - name: Installer le serveur WEB Apache
    yum:
      name: httpd
      state: latest

  # Copie des fichiers de configuration d'Apache

  - name: Copie du fichier httpd.conf
    copy:
      src: /tmp/httpd.conf
      dest: /etc/httpd/conf

  - name: Copie du fichier index.html
    copy:
      src: /tmp/index_install.html
      dest: /var/www/html/index.html
      owner: apache
      group: apache

  # Redemarrage du seueur Apache
  - name: Redemarrage du serveur WEB Apache
    service:
      name: httpd
      state: restarted

  # Arret du pare-feu
  - name: Arret du pare-feu
    service:
      name: firewalld
```

```
state: stopped
enabled: no
```

Remarque : Plusieurs sections hosts peuvent être présents dans le fichier.

Vérifier la syntaxe du fichier par : `--syntax-check`.

```
[root@mars apache_install]# ansible-playbook --syntax-check
apache_install.yml

playbook: apache_install.yml
```

Vérifier le fonctionnement par : `--check`

Simuler l' exécution du playbook.

```
[root@mars apache_install]# ansible-playbook --check apache_install.yml

PLAY [clients]
*****
*****

TASK [Gathering Facts]
*****
**
ok: [client11]
ok: [client12]

TASK [Installer le serveur WEB Apache]
*****
ok: [client12]
ok: [client11]

TASK [Copie du fichier httpd.conf]
*****
ok: [client12]
ok: [client11]

TASK [Copie du fichier index.html]
*****
ok: [client11]
ok: [client12]

TASK [Redemarrage du serveur WEB Apache]
*****
changed: [client12]
changed: [client11]

TASK [Arret du pare-feu]
*****
ok: [client11]
ok: [client12]

PLAY RECAP
*****
client11          : ok=6    changed=1    unreachable=0    failed=0
client12          : ok=6    changed=1    unreachable=0    failed=0
```

Execution du playbook.

```
[root@mars apache_install]# ansible-playbook apache_install.yml

PLAY [clients]
*****

TASK [Gathering Facts]
*****
**
ok: [client11]
ok: [client12]

TASK [Installer le serveur WEB Apache]
*****
changed: [client11]
changed: [client12]

TASK [Copie du fichier httpd.conf]
*****
changed: [client11]
changed: [client12]

TASK [Copie du fichier index.html]
*****
changed: [client11]
changed: [client12]

TASK [Redemarrage du serveur WEB Apache]
*****
changed: [client11]
changed: [client12]

PLAY RECAP
*****
client11          : ok=5    changed=4    unreachable=0    failed=0
client12          : ok=5    changed=4    unreachable=0    failed=0
```

Vérifier si httpd est installé sur client11.

```
[root@mars apache_install]# ssh client11 'rpm -qa | grep httpd'
httpd-tools-2.4.6-67.el7.centos.6.x86_64
httpd-2.4.6-67.el7.centos.6.x86_64
```

Vérifier si httpd est démarré sur client11.

```
[root@mars apache_install]# ssh client11 'pgrep -l httpd'
3940 httpd
3941 httpd
3942 httpd
3943 httpd
3944 httpd
3945 httpd
```

Executer un navigateur et connectez-vous sur http://adresse_ip

Exercice 2

Créer un répertoire variables. Positionnez-vous sur ce répertoire.

```
[root@mars ~]# mkdir variables
[root@mars ~]# cd !$
cd variables
```

1 - Votre playbook variables.yml doit afficher le contenu de la variable compte_bdd.

Modification du fichier /etc/ansible/hosts pour déclarer la variables.

```
[root@mars variables]# tail -2 /etc/ansible/hosts
[all:vars]
compte_bdd="Administrateur Maria DB"
```

Création du Playbook.

```
[root@mars variables]# more variables.yml
- hosts: all
  tasks:
    - debug:
        msg: "Le contenu de la variable compte_bdd est:  {{compte_bdd}}"
```

Exécution du Playbook

```
[root@mars variables]# ansible-playbook variables.yml

PLAY [all]
*****
*****

TASK [Gathering Facts]
*****
*
ok: [debian11]
ok: [debian12]
ok: [client12]
ok: [client11]

TASK [debug]
*****
*****
ok: [debian11] => {
    "msg": "Le contenu de la variable compte_bdd est:  Administrateur Maria DB"
}
ok: [debian12] => {
    "msg": "Le contenu de la variable compte_bdd est:  Administrateur Maria DB"
}
ok: [client11] => {
    "msg": "Le contenu de la variable compte_bdd est:  Administrateur Maria DB"
}
ok: [client12] => {
    "msg": "Le contenu de la variable compte_bdd est:  Administrateur Maria DB"
}

PLAY RECAP
*****
*****
client11          : ok=2    changed=0    unreachable=0    failed=0
client12          : ok=2    changed=0    unreachable=0    failed=0
debian11          : ok=2    changed=0    unreachable=0    failed=0
debian12          : ok=2    changed=0    unreachable=0    failed=0
```

Créer deux répertoires `group_vars` et `host_vars` au même niveau que votre playbook.

```
[root@mars variables]# mkdir group_vars host_vars
```

Vous devez exploiter ces répertoires au mieux pour que la variable `compte_bdd` ai la valeur `none` pour tous les postes, sauf pour les machines du groupe `clients` où elle correspond à `opérateur`. Une exception concerne la machine `client1` où la valeur doit être `administrateur`.

Veillez suivre les étapes suivantes :

Faites que la variables ai `none` pour tous les postes.

```
[root@mars variables]# cat group_vars/all
compte_bdd: "none"
```

Test du playbook.

```
[root@mars variables]# ansible-playbook variables.yml

PLAY [all]
*****
*****

TASK [Gathering Facts]
*****
*
ok: [debian11]
ok: [debian12]
ok: [client11]
ok: [client12]

TASK [debug]
*****
*****
ok: [debian11] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
}
ok: [debian12] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
}
ok: [client11] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
}
ok: [client12] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
}

PLAY RECAP
*****
*****
client11      : ok=2    changed=0    unreachable=0    failed=0
client12      : ok=2    changed=0    unreachable=0    failed=0
debian11      : ok=2    changed=0    unreachable=0    failed=0
debian12      : ok=2    changed=0    unreachable=0    failed=0
```

Faites que la variable ait la valeur `opérateur` pour les postes du groupe `clients`

```
[root@mars variables]# cat group_vars/clients
compte_bdd: operateur
```

Vérifier par une exécution du playbook.

```
[root@mars variables]# ansible-playbook variables.yml

PLAY [all]
*****

TASK [Gathering Facts]
*****
*
ok: [debian11]
ok: [debian12]
ok: [client11]
ok: [client12]

TASK [debug]
*****
ok: [debian11] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
}
ok: [debian12] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
}
ok: [client11] => {
    "msg": "Le contenu de la variable compte_bdd est:  operateur"
}
ok: [client12] => {
    "msg": "Le contenu de la variable compte_bdd est:  operateur"
}

PLAY RECAP
*****
client11                : ok=2    changed=0    unreachable=0    failed=0
client12                : ok=2    changed=0    unreachable=0    failed=0
debian11                : ok=2    changed=0    unreachable=0    failed=0
debian12                : ok=2    changed=0    unreachable=0    failed=0
```

Faites que la variable ait la valeur administrateur pour le poste client1.

```
[root@mars variables]# cat host_vars/client1
compte_bdd: administrateur
```

Vérifier par une exécution de votre playbook.

```
[root@mars variables]# ansible-playbook variables.yml

PLAY [all]
*****

TASK [Gathering Facts]
*****
*
ok: [debian11]
ok: [debian12]
ok: [client12]
ok: [client11]

TASK [debug]
*****
ok: [debian11] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
}
ok: [client11] => {
```

```

    "msg": "Le contenu de la variable compte_bdd est:  administrateur"
  }
ok: [debian12] => {
    "msg": "Le contenu de la variable compte_bdd est:  none"
  }
ok: [client12] => {
    "msg": "Le contenu de la variable compte_bdd est:  operateur"
  }
}

PLAY RECAP
*****
*****
client11          : ok=2    changed=0    unreachable=0    failed=0
client12          : ok=2    changed=0    unreachable=0    failed=0
debian11          : ok=2    changed=0    unreachable=0    failed=0

```

2 - Exceptionnellement, par exemple pour un test spécifique, vous devez exécuter votre playbook avec la variable compte_bdd positionnée à TEST.

```

[root@mars variables]# ansible-playbook -e compte_bdd=TEST variables.yml

PLAY [all]
*****
*****

TASK [Gathering Facts]
*****
*
ok: [debian11]
ok: [debian12]
ok: [client12]
ok: [client11]

TASK [debug]
*****
*****
ok: [client11] => {
    "msg": "Le contenu de la variable compte_bdd est:  TEST"
  }
ok: [debian11] => {
    "msg": "Le contenu de la variable compte_bdd est:  TEST"
  }
ok: [client12] => {
    "msg": "Le contenu de la variable compte_bdd est:  TEST"
  }
ok: [debian12] => {
    "msg": "Le contenu de la variable compte_bdd est:  TEST"
  }
}

PLAY RECAP
*****
*****
client11          : ok=2    changed=0    unreachable=0    failed=0
client12          : ok=2    changed=0    unreachable=0    failed=0
debian11          : ok=2    changed=0    unreachable=0    failed=0
debian12          : ok=2    changed=0    unreachable=0    failed=0

```

Exercice 3

Le formateur vous fournit les fichiers httpd.conf et index_template.html.

Créer un répertoire apache_template. Positionnez-vous sur ce répertoire.

```
[root@mars ~]# mkdir apache_template && cd $_
```

1.

Le playbook à créer est apache_template.yml.

Il doit réaliser les mêmes opérations que l'exercice 1. La différence concerne la page de garde (index.html) qui cette fois est un template.

Les étapes :

Mettre à jour le fichier index_template.html pour faire référence aux bonnes variables.

```
[root@mars ~]# more /tmp/index_template.html
<html>
  <head><title>Page de garde de {{ansible_hostname}} </title></head>
  <body>
    <hr>
    <center><h1>Machine   {{ansible_hostname}} </h1>
      du domaine   {{ansible_domain}} </center>
    <br><hr>
    <center><b>
<br> Bonjour {{ansible_user_id}}
<br>
<br> La distribution est : {{ansible_distribution}}
<br> La taille de la memoire est : {{ansible_memtotal_mb}} Mo
<br> La taille de la swap est : {{ansible_swaptotal_mb}} Mo
<br>
<br> La liste des adresses IP :<br>
      {% for addrip in ansible_all_ipv4_addresses %}
      {{addrip}} <br>
      {% endfor %}
<br> La liste des interfaces resaux :<br>
      {% for nic in ansible_interfaces %}
      {{nic}} <br>
      {% endfor %}
<br><hr>
    </b></center>
  </body>
</html>
```

Créer le playbook qui utilisera le template pour la page de garde du site :
/var/www/html/index.html

Copie du playbook créer dans la manip 3

```
[root@mars apache_template]# cp ../apache_install/apache_install.yml  
apache_template.yml
```

```
[root@mars apache_template]# cat apache_template.yml
```

```
---  
# Creation de mon premier playbook  
#####  
  
# Liste des machines concernées  
  
- hosts: clients  
  
# Taches a effectuer  
  
tasks:  
  
  # Installation du package Apache  
  
  - name: Installer le serveur WEB Apache  
    yum:  
      name: httpd  
      state: latest  
  
  # Copie des fichiers de configuration d'Apache  
  
  - name: Copie du fichier httpd.conf  
    copy:  
      src: /tmp/httpd.conf  
      dest: /etc/httpd/conf  
  
  - name: Copie du fichier template en index.html  
    template:  
      src: /tmp/index_template.html  
      dest: /var/www/html/index.html  
      owner: apache  
      group: apache  
  
  # Redemarrage du seveur Apache  
  - name: Redemarrage du serveur WEB Apache  
    service:  
      name: httpd  
      state: restarted  
  
  # Arret du pare-feu  
  - name: Arret du pare-feu  
    service:  
      name: firewalld  
      state: stopped  
      enabled: no
```

Vérifier la syntaxe du fichier par : --syntax-check

```
[root@mars apache_template]# ansible-playbook --syntax-check  
apache_template.yml
```

```
playbook: apache_template.yml
```

Vérifier le fonctionnement par : --check

```
[root@mars apache_template]# ansible-playbook --check apache_template.yml

PLAY [clients]
*****

TASK [Gathering Facts]
*****
*
ok: [client11]
ok: [client12]

TASK [Installer le serveur WEB Apache]
*****
ok: [client12]
ok: [client11]

TASK [Copie du fichier httpd.conf]
*****
ok: [client11]
ok: [client12]

TASK [Copie du fichier template en index.html]
*****
changed: [client11]
changed: [client12]

TASK [Redemarrage du serveur WEB Apache]
*****
changed: [client12]
changed: [client11]

TASK [Arret du pare-feu]
*****
ok: [client11]
ok: [client12]

PLAY RECAP
*****
client11          : ok=6    changed=2    unreachable=0    failed=0
client12          : ok=6    changed=2    unreachable=0    failed=0
```

Exécuter le playbook.

```
[root@mars apache_template]# ansible-playbook apache_template.yml
```

Vérifier par votre navigateur : <http://client1> & <http://client2>

Ca fonctionne bien. Tout est OK.

2.

Mettre à jour le fichier `index_template` pour afficher la valeur de la variable `LANG`.

Sous la ligne traitant de la swap, ajouter la ligne suivante :

`<br>` La variable `LANG` = ... Le contenu de la variable `LANG` ...

Modification du fichier `index_template.html`

```
[root@mars apache_template]# grep -A 1 swap /tmp/index_template.html
<br> La taille de la swap est : {{ansible_swaptotal_mb}} Mo
<br> La variable LANG est initialisee avec la valeur suivante: {{ansible_env.LANG}}
```

Réexécuter votre playbook.

```
[root@mars apache_template]# ansible-playbook apache_template.yml
```

Vérifier par votre navigateur : <http://client1> & <http://client2>

Ca fonctionne bien. Tout est OK

Exercice 4

Créer un répertoire debug_register. Positionnez-vous sur ce répertoire.

```
[root@mars ~]# mkdir debug_register && cd $_
```

Le playbook à créer est affichage.yml. Il concerne toutes les machines.

Le playbook doit récupérer le résultat de la commande : free -h

Pour afficher le résultat de la commande, utiliser le module debug :

Affichage classique .

```
[root@mars debug_register]# cat affichage_vers0.yml
---
- hosts: all
  tasks:
    - command: free -m
      register: resultat
    - debug: var=resultat
```

première version : avec stdout.

```
[root@mars debug_register]# cat affichage_vers1.yml
---
- hosts: all
  tasks:
    - command: free -m
      register: resultat
    - debug: var=resultat.stdout
```

deuxième version : avec stdout_lines.

```
[root@mars debug_register]# cat affichage_vers2.yml
---
- hosts: all
  tasks:
    - command: free -m
      register: resultat
    - debug: var=resultat.stdout_lines
```

troisième version : avec stdout_lines[1].

```
[root@mars debug_register]# cat affichage_vers3.yml
---
- hosts: all
  tasks:
    - command: free -m
      register: resultat
    - debug: var=resultat.stdout_lines[1]
```

Toutes les versions dans le même fichier

```
[root@mars debug_register]# cat affichage.yml
---
- hosts: all
  tasks:
    - command: free -m
      register: resultat
    - debug: var=resultat
    - debug: var=resultat.stdout
    - debug: var=resultat.stdout_lines
    - debug: var=resultat.stdout_lines[1]
```

quatrième version, afficher :

Code de retour=...Le_code_de_retour..., Ligne=...La_ligne_index_1...

```
[root@mars debug_register]# cat affichage_code_retour.yml
---
- hosts: all
  tasks:
    - command: free -h
      register: resultat
    - debug: msg="Code de retour={{resultat.rc}}, Ligne={{resultat.stdout_lines[1]}}"
```

Execution du playbook.

```
[root@mars debug_register]# ansible-playbook affichage_code_retour.yml

PLAY [all]
*****
*****

...

TASK [debug]
*****
*****
ok: [client11] => {
  "msg": "Code de retour=0, Ligne=Mem:          488M          90M          103M
4,4M          294M          356M"
}
ok: [client12] => {
  "msg": "Code de retour=0, Ligne=Mem:          488M          94M          99M
4,4M          294M          352M"
}
ok: [debian12] => {
  "msg": "Code de retour=0, Ligne=Mem:          1,0G          380M          113M
4,5M          511M          474M"
}
ok: [debian11] => {
  "msg": "Code de retour=0, Ligne=Mem:          1,0G          29M          735M
2,9M          239M          835M"
}

PLAY RECAP
*****
*****
client11          : ok=3    changed=1    unreachable=0    failed=0
client12          : ok=3    changed=1    unreachable=0    failed=0
debian11          : ok=3    changed=1    unreachable=0    failed=0
debian12          : ok=3    changed=1    unreachable=0    failed=0
```

Exercice 5

Créer un répertoire `apache_handler`. Positionnez-vous sur ce répertoire.

```
[root@mars ~]# mkdir apache_handler && cd $_
```

Le playbook à créer est `apache_handler.yml`. Il concerne les machines du groupe clients.

```
[root@mars apache_handler]# cp ../apache_template/apache_template.yml  
./apache_handler.yml
```

On reprend l'exercice Apache avec la version des templates (exercice 3).

On décide que le service Apache doit redémarrer si le fichier de configuration ou la page web a changé.

```
[root@mars apache_handler]# cat apache_handler.yml
---
# Creation de mon premier playbook
#####

# Liste des machines concernées

- name: "Test handler pour apache"
  hosts: clients
  handlers:
    - name: "Redemarrage du service httpd"
      service:
        name: httpd
        state: restarted

# Taches à effectuer

tasks:

  # Installation du package Apache

  - name: Installer le serveur WEB Apache
    yum:
      name: httpd
      state: latest

  # Copie des fichiers de configuration d'Apache

  - name: Copie du fichier httpd.conf
    copy:
      src: /tmp/httpd.conf
      dest: /etc/httpd/conf
      notify: [ "Redemarrage du service httpd" ]

  - name: Copie du fichier template en index.html
    template:
      src: /tmp/index_template.html
      dest: /var/www/html/index.html
      owner: apache
      group: apache
      notify: [ "Redemarrage du service httpd" ]

  # Arrêt du pare-feu
  - name: Arrêt du pare-feu
    service:
      name: firewalld
      state: stopped
      enabled: no
```

Pour tester :

- Première exécution Installation ? Modification ? Redémarrage ?

```
[root@mars apache_handler]# ansible-playbook apache_handler.yml

PLAY [Test handler pour apache]
*****

TASK [Gathering Facts]
*****
*
ok: [client11]
ok: [client12]

TASK [Installer le serveur WEB Apache]
*****
ok: [client12]
ok: [client11]

TASK [Copie du fichier httpd.conf]
*****
ok: [client11]
ok: [client12]

TASK [Copie du fichier template en index.html]
*****
ok: [client11]
ok: [client12]

TASK [Arret du pare-feu]
*****
ok: [client12]
ok: [client11]

PLAY RECAP
*****
client11          : ok=5    changed=0    unreachable=0    failed=0
client12          : ok=5    changed=0    unreachable=0    failed=0
```

- Deuxième exécution Installation ? Modification ? Redémarrage ?

On constate que ce n'est pas réinstallé

- Modification d un fichier (ajouter un commentaire) et exécution :

Installation ? Modification ? Redémarrage ?

- Modification des 2 fichiers (ajouter un commentaire) :

```
[root@mars apache_handler]# ansible-playbook apache_handler.yml

PLAY [Test handler pour apache]
*****

TASK [Gathering Facts]
*****
*
ok: [client11]
ok: [client12]

TASK [Installer le serveur WEB Apache]
*****
ok: [client11]
ok: [client12]

TASK [Copie du fichier httpd.conf]
*****
changed: [client12]
changed: [client11]

TASK [Copie du fichier template en index.html]
*****
changed: [client11]
changed: [client12]

TASK [Arret du pare-feu]
*****
ok: [client11]
ok: [client12]

RUNNING HANDLER [Redemarrage du service httpd]
*****
changed: [client12]
changed: [client11]

PLAY RECAP
*****
client11          : ok=6    changed=3    unreachable=0    failed=0
client12          : ok=6    changed=3    unreachable=0    failed=0
```

Remarque : Les handlers ne sont exécutés qu'après les différentes tâches. Si un service fait appel à plusieurs handlers ils sont exécutés dans l'ordre dans lequel ils ont été définis.
Si plusieurs services font appel au même handler celui-ci n'est exécuté qu'une seule fois

Installation ? Modification ? Redémarrage ?

- Nouvelle exécution Installation ? Modification ? Redémarrage ?

Exercice 6

Créer un répertoire boucles. Positionnez-vous sur ce répertoire.

```
[root@mars ~]# mkdir boucles && cd $_
```

Le playbook à créer est boucles.yml. Il concerne les machines du groupe clients.

Installer les packages perl, nmap, php, webmin par une boucle
pour webmin utiliser :

<http://download.webmin.com/download/yum/webmin-1.881-1.noarch.rpm>

Créer les comptes suivants :

jean avec uid=1101, groupe team (gid 1100), répertoire de connexion /home/jean

marc avec uid=1102, groupe team (gid 1100), répertoire de connexion /home/marc

eve avec uid=1103, groupe team (gid 1100), répertoire de connexion /home/eve

celine avec uid=1104, groupe team (gid 1100), répertoire de connexion /home/celine

utiliser une variable pour le nom du groupe et une variable pour le gid.

utiliser un tableau pour le nom et le uid.

```
[root@mars boucles]# cat boucle.yml
---
- hosts: clients
  vars:
    grp_name: team
    grp_gid: 1100
    users:
      - {nom: jean, uid: 1101, groupe: "{{grp_gid}}" }
      - {nom: marc, uid: 1002, groupe: "{{grp_gid}}" }
      - {nom: eve, uid: 1003, groupe: "{{grp_gid}}" }
      - {nom: celine, uid: 1004, groupe: "{{grp_gid}}" }
  tasks:
    - name: Installation des packages
      yum:
        name: "{{item}}"
        state: present
      loop:
        - perl
        - nmap
        - php
        - http://download.webmin.com/download/yum/webmin-1.881-1.noarch.rpm
    - name: Creation du groupe
      group:
        state: present
        name: team
        gid: "{{grp_gid}}"
    - name: Creation des users
      user:
        state: present
        name: "{{item.nom}}"
        uid: "{{item.uid}}"
        group: "{{item.groupe}}"
        home: /home/"{{item.nom}}"
        create_home: yes
      loop: "{{users}}"
```

Execution du playbook

```
[root@mars boucles]# ansible-playbook boucle.yml

PLAY [clients]
*****

TASK [Gathering Facts]
*****
*
ok: [client11]
ok: [client12]

TASK [Installation des packages]
*****
changed: [client12] => (item=perl)
changed: [client11] => (item=perl)
changed: [client12] => (item=nmap)
changed: [client11] => (item=nmap)
changed: [client12] => (item=php)
changed: [client11] => (item=php)
changed: [client12] => (item=http://download.webmin.com/download/yum/webmin-1.881-1.noarch.rpm)
changed: [client11] => (item=http://download.webmin.com/download/yum/webmin-1.881-1.noarch.rpm)

TASK [Creation du groupe]
*****
changed: [client11]
changed: [client12]

TASK [Creation des users]
*****
changed: [client12] => (item={u'groupe': 1100, u'nom': u'jean', u'uid': 1101})
changed: [client11] => (item={u'groupe': 1100, u'nom': u'jean', u'uid': 1101})
changed: [client12] => (item={u'groupe': 1100, u'nom': u'marc', u'uid': 1002})
changed: [client11] => (item={u'groupe': 1100, u'nom': u'marc', u'uid': 1002})
changed: [client12] => (item={u'groupe': 1100, u'nom': u'eve', u'uid': 1003})
changed: [client11] => (item={u'groupe': 1100, u'nom': u'eve', u'uid': 1003})
changed: [client12] => (item={u'groupe': 1100, u'nom': u'celine', u'uid': 1004})
changed: [client11] => (item={u'groupe': 1100, u'nom': u'celine', u'uid': 1004})

PLAY RECAP
*****
client11          : ok=4    changed=3    unreachable=0    failed=0
client12          : ok=4    changed=3    unreachable=0    failed=0
```

Pour tester webmin : <https://hostname:10000> login avec root

Ajouter l'exception de sécurité et se logger avec root/root

Exercice 7

1. Le service cron

Créer un répertoire condition_when. Positionnez-vous sur ce répertoire.

```
[root@mars ~]# mkdir condition_when && cd $_
```

Le playbook à créer est cron_when.yml. Il concerne toutes les machines.

Le playbook doit redémarrer le service cron sur les centos et les debian :
pour une CentOS, le service est crond,
pour une Debian, le service est cron
consulter la variable ansible_distribution
optionnellement, vous pouvez utiliser le filtre lower ou upper

Le playbook

```
[root@mars condition_when]# cat cron_when.yml
```

```
---
- hosts: all
  tasks:
    - name: Redemarre cron sur CentOS
      service: name=crond state=restarted
      when: ansible_distribution|lower == "centos"
    - name: Redemarre cron sur Debian
      service: name=cron state=restarted
      when: ansible_distribution|lower == "debian"
```

L'exécution du playbook

```
[root@mars condition_when]# ansible-playbook cron_when.yml
```

```
PLAY [all]
*****
*****

TASK [Gathering Facts]
*****
*
ok: [debian11]
ok: [debian12]
ok: [client12]
ok: [client11]

TASK [Redemarre cron sur CentOS]
*****
skipping: [debian11]
skipping: [debian12]
changed: [client12]
changed: [client11]

TASK [Redemarre cron sur Debian]
*****
skipping: [client11]
skipping: [client12]
changed: [debian11]
changed: [debian12]

PLAY RECAP
*****
client11          : ok=2    changed=1    unreachable=0    failed=0
client12          : ok=2    changed=1    unreachable=0    failed=0
debian11          : ok=2    changed=1    unreachable=0    failed=0
debian12          : ok=2    changed=1    unreachable=0    failed=0
```

2. Apache

Recopier le répertoire de l'exercice 3 (Apache avec Templates).

Le playbook à créer est `apache_when.yml`. Il concerne les machines du groupe clients.

Le playbook doit redémarrer le service Apache si un fichier a été copié (paramètre `changed`).
(que ce soit le fichier de configuration ou de la page html)

Le Playbook

```
[root@mars condition_when]# cat apache_when.yml
---
# Creation de mon premier playbook
#####

# Liste des machines concernées

- hosts: clients

# Taches a effectuer

tasks:

    # Installation du package Apache

    - name: Installer le serveur WEB Apache
      yum:
        name: httpd
        state: latest

    # Copie des fichiers de configuration d'Apache

    - name: Copie du fichier httpd.conf
      copy:
        src: /tmp/httpd.conf
        dest: /etc/httpd/conf
        register: res1

    - name: Copie du fichier template en index.html
      template:
        src: /tmp/index_template.html
        dest: /var/www/html/index.html
        owner: apache
        group: apache
        register: res2

    # Redemarrage du seueur Apache
    - name: Redemarrage du serveur WEB Apache
      service:
        name: httpd
        state: restarted
        enabled: yes
        when: res1.changed or res2.changed

    # Arret du pare-feu
    - name: Arret du pare-feu
      service:
        name: firewalld
        state: stopped
        enabled: no
```

Nota: pour le service -> skipping si le service n est pas redémarré.
-> ok si le service est redémarré.

```
[root@mars condition_when]# ansible-playbook apache_when.yml

PLAY [clients]
*****

TASK [Gathering Facts]
*****
*
ok: [client11]
ok: [client12]

TASK [Installer le serveur WEB Apache]
*****
ok: [client11]
ok: [client12]

TASK [Copie du fichier httpd.conf]
*****
ok: [client12]
ok: [client11]

TASK [Copie du fichier template en index.html]
*****
ok: [client12]
ok: [client11]

TASK [Redemarrage du serveur WEB Apache]
*****
skipping: [client11]
skipping: [client12]

TASK [Arret du pare-feu]
*****
ok: [client12]
ok: [client11]

PLAY RECAP
*****
client11          : ok=5    changed=0    unreachable=0    failed=0
client12          : ok=5    changed=0    unreachable=0    failed=0
```

LES ROLES

Exercice 1

```
[root@mars ~]# mkdir lab_roles && cd $_
[root@mars lab_roles]# mkdir roles
[root@mars lab_roles]# cd roles
[root@mars lab_roles]# ansible-galaxy init apache
[root@mars roles]# tree
```

```
.
├── apache
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── README.md
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
```

9 directories, 8 files

Après copie et adaptation de fichiers on obtient l'arborescence suivante:

```
[root@mars lab_roles]# tree
```

```
.
├── apache.yml
├── roles
│   └── apache
│       ├── defaults
│       │   └── main.yml
│       ├── files
│       │   ├── apache2.conf
│       │   └── httpd.conf
│       ├── handlers
│       │   └── main.yml
│       ├── meta
│       │   └── main.yml
│       ├── README.md
│       ├── tasks
│       │   ├── centos.yml
│       │   ├── debian.yml
│       │   └── main.yml
│       ├── templates
│       │   └── index_template.html
│       ├── tests
│       │   ├── inventory
│       │   └── test.yml
│       └── vars
│           └── main.yml
```

10 directories, 14 files

Contenu des fichiers du rôle.

```
[root@mars lab_roles]# more apache.yml
```

```
---
- hosts: all
  roles:
    - apache
```

```
[root@mars lab_roles]# more roles/apache/handlers/main.yml
```

```
---
# handlers file for apache

- name: "Redemarrage du service httpd"
  service:
    name: httpd
    state: restarted

- name: "Redemarrage du service apache2"
  service:
    name: apache2
    state: restarted
```

```
[root@mars lab_roles]# more roles/apache/tasks/centos.yml
```

```
---
# Installation du package Apache

- name: Installer le serveur WEB Apache
  yum:
    name: httpd
    state: latest

# Copie des fichiers de configuration d'Apache

- name: Copie du fichier httpd.conf
  copy:
    src: httpd.conf
    dest: /etc/httpd/conf
  notify: [ "Redemarrage du service httpd" ]
```

```
[root@mars lab_roles]# more roles/apache/tasks/debian.yml
```

```
---
- name: Installer les outils de developpement
  apt:
    name: build-essential
    state: present

- name: Installer Apache
  apt:
    name: apache2
    state: latest

- name: Copier le fichier de configuration d Apache
  copy:
    src: apache2.conf
    dest: /etc/apache2/apache2.conf
  notify: "Redemarrage du service apache2"
```

```
[root@mars lab_roles]# more roles/apache/tasks/main.yml
```

```
---
```

```
# tasks file for apache
```

```
- name: Installer et demarrer apache sur les serveurs CentOS
  include_tasks: centos.yml
  when: ansible_distribution == 'CentOS'
- name: Installer et demarrer apache sur les serveurs Debian
  include_tasks: debian.yml
  when: ansible_distribution == 'Debian'
- name: Copie du template
  template:
    src: index_template.html
    dest: /var/www/html/index.html
```

```
[root@mars lab_roles]# more roles/apache/templates/index_template.html
```

```
<html>
  <head><title>Page de garde de {{ansible_hostname}} </title></head>
  <body>
    <hr>
    <center><h1>Machine  {{ansible_hostname}} </h1>
      du domaine  {{ansible_domain}} </center>
    <br><hr>
    <center><b>
<br> Bonjour {{ansible_user_id}}
<br>
<br> La distribution est : {{ansible_distribution}}
<br> La taille de la memoire est : {{ansible_memtotal_mb}} Mega octets
<br> La taille de la swap est : {{ansible_swaptotal_mb}} Mega octets
<br> La variable LANG est initialisee avec la valeur suivante: {{ansible_env.LANG}}
<br>
<br> La liste des adresses IP :<br>
      {% for addrip in ansible_all_ipv4_addresses %}
      {{addrip}} <br>
      {% endfor %}
<br> La liste des interfaces resaux :<br>

      {% for nic in ansible_interfaces %}
      {{nic}} <br>
      {% endfor %}

<br><hr>
    </b></center>
  </body>
</html>
```

Exercice 2

Affichage d'un rôle.

```
[root@mars ~]# ansible-galaxy info barangerjeanmarc.site_lamp
Role: barangerjeanmarc.site_lamp
  description: Un rôle pour un site LAMP
  active: True
  commit: e8b243c7a9023e0c4b389790726558f768c69ba4
  commit_message: Add files via upload
  commit_url:
https://github.com/barangerjeanmarc/site_lamp/commit/e8b243c7a9023e0c4b389790726558f768c69ba4
  company: Sphérius
...
```

Installation du rôle.

```
[root@mars lab_roles]# ansible-galaxy install barangerjeanmarc.site_lamp
--roles-path roles
- downloading role 'site_lamp', owned by barangerjeanmarc
- downloading role from
https://github.com/barangerjeanmarc/site_lamp/archive/master.tar.gz
- extracting barangerjeanmarc.site_lamp to
/root/lab_roles/roles/barangerjeanmarc.site_lamp
- barangerjeanmarc.site_lamp (master) was installed successfully
```

Aborescence après l'installation du rôle.

```
[root@mars lab_roles]# tree
.
├── apache.yml
├── roles
│   ├── apache
│   │   ├── defaults
│   │   │   └── main.yml
│   │   ├── files
│   │   │   ├── apache2.conf
│   │   │   └── httpd.conf
│   │   ├── handlers
│   │   │   └── main.yml
│   │   ├── meta
│   │   │   └── main.yml
│   │   ├── README.md
│   │   ├── tasks
│   │   │   ├── centos.yml
│   │   │   ├── debian.yml
│   │   │   └── main.yml
│   │   ├── templates
│   │   │   └── index_template.html
│   │   ├── tests
│   │   │   ├── inventory
│   │   │   └── test.yml
│   │   ├── vars
│   │   │   └── main.yml
│   └── barangerjeanmarc.site_lamp
│       ├── defaults
│       │   └── main.yml
│       ├── files
│       │   ├── annuaire.sql
│       │   └── page.php
│       ├── handlers
│       │   └── main.yml
│       ├── meta
│       │   └── main.yml
```

```
├── README.md
├── tasks
│   ├── bdd_install.yml
│   ├── bdd_manip.yml
│   ├── bdd_php.yml
│   └── main.yml
├── tests
│   ├── inventory
│   └── test.yml
├── vars
│   └── main.yml
└── site.yml
```

18 directories, 28 files

Création du fichier site.yml dans le répertoire lab_roles.

```
[root@mars lab_roles]# more site.yml
```

```
---
- hosts: clients
  roles:
    - apache
    - barangerjeanmarc.site_lamp
```

Execution du playbook.

```
[root@mars lab_roles]# ansible-playbook site.yml
```

Exercice 1

```
[root@mars ~]# mkdir delegate_to && cd $_

[root@mars delegate_to]# more apache_delegate.yml
---
# Liste des machines concernées
- hosts: client11

  # Taches a effectuer

  tasks:

    # Installation du package Apache

    - name: Installer le serveur WEB Apache
      yum:
        name: httpd
        state: latest

    # Copie des fichiers de configuration d'Apache

    - name: Copie du fichier httpd.conf
      copy:
        src: /tmp/httpd.conf
        dest: /etc/httpd/conf

    # Redemarrage du seueur Apache
    - name: Redemarrage du serveur WEB Apache
      service:
        name: httpd
        state: restarted

- hosts: all

  # Taches a effectuer

  tasks:

    - name: Copie du fichier template en index.html
      template:
        src: /tmp/index_template.html
        dest: /var/www/html/{{inventory_hostname}}.html
        owner: apache
        group: apache
        delegate_to: client11
```

Vérifier par <http://client11/debian11> ou <http://client11/client11> ou ...

Exercise 2

1/

```
[root@mars delegate_to]# more apache_delegate.yml
---
# Liste des machines concernées
- hosts: client11
  gather_facts: no

  # Taches a effectuer

  tasks:

    # Installation du package Apache

    - name: Installer le serveur WEB Apache
      yum:
        name: httpd
        state: latest

    # Copie des fichiers de configuration d'Apache

    - name: Copie du fichier httpd.conf
      copy:
        src: /tmp/httpd.conf
        dest: /etc/httpd/conf

    # Redemarrage du seveur Apache
    - name: Redemarrage du serveur WEB Apache
      service:
        name: httpd
        state: restarted

- hosts: all

  # Taches a effectuer

  tasks:

    - name: Copie du fichier template en index.html
      template:
        src: /tmp/index_template.html
        dest: /var/www/html/{{inventory_hostname}}.html
        owner: apache
        group: apache
      delegate_to: client11
```

Execution du playbook.

```
[root@mars delegate_to]# ansible-playbook apache_delegate.yml
```

On constate bien que les facts n'apparaissent pas.

2/

```
[root@mars delegate_to]# more apache_delegate.yml
```

```
---
```

```
# Liste des machines concernées
```

```
- hosts: client11
```

```
# Taches a effectuer
```

```
tasks:
```

```
  # Installation du package Apache
```

```
  - name: Installer le serveur WEB Apache
```

```
    yum:
```

```
      name: httpd
```

```
      state: latest
```

```
  # Copie des fichiers de configuration d'Apache
```

```
  - name: Copie du fichier httpd.conf
```

```
    copy:
```

```
      src: /tmp/httpd.conf
```

```
      dest: /etc/httpd/conf
```

```
  # Redemarrage du seueur Apache
```

```
  - name: Redemarrage du serveur WEB Apache
```

```
    service:
```

```
      name: httpd
```

```
      state: restarted
```

```
- hosts: all
```

```
  gather_facts: no
```

```
# Taches a effectuer
```

```
tasks:
```

```
  - name: Copie du fichier template en index.html
```

```
    template:
```

```
      src: /tmp/index_template.html
```

```
      dest: /var/www/html/{{inventory_hostname}}.html
```

```
      owner: apache
```

```
      group: apache
```

```
      delegate_to: client11
```

Résultat de l'exécution du playbook.

```
[root@mars delegate_to]# ansible-playbook apache_delegate.yml

PLAY [client11]
*****

TASK [Gathering Facts]
*****
ok: [client11]

TASK [Installer le serveur WEB Apache]
*****
ok: [client11]

TASK [Copie du fichier httpd.conf]
*****
ok: [client11]

TASK [Redemarrage du serveur WEB Apache]
*****
changed: [client11]

PLAY [all]
*****

TASK [Copie du fichier template en index.html]
*****
fatal: [debian11 -> client11]: FAILED! => {"changed": false, "msg":
"AnsibleUndefinedVariable: 'ansible_env' is undefined"}
fatal: [client12 -> client11]: FAILED! => {"changed": false, "msg":
"AnsibleUndefinedVariable: 'ansible_env' is undefined"}
fatal: [debian12 -> client11]: FAILED! => {"changed": false, "msg":
"AnsibleUndefinedVariable: 'ansible_env' is undefined"}
ok: [client11 -> client11]
    to retry, use: --limit @/root/delegate_to/apache_delegate.retry

PLAY RECAP
*****
client11          : ok=5    changed=1    unreachable=0    failed=0
client12          : ok=0    changed=0    unreachable=0    failed=1
debian11          : ok=0    changed=0    unreachable=0    failed=1
debian12          : ok=0    changed=0    unreachable=0    failed=1
```