

ANSIBLE EXERCICES



Votre partenaire formation ...

UNIX - LINUX - WINDOWS - ORACLE - VIRTUALISATION



www.spherius.fr

SOMMAIRE

| | |
|---|-----------|
| INSTALLATION ANSIBLE..... | 4 |
| CONFIGURATION ET UTILISATION D'ANSIBLE..... | 5 |
| LES COMMANDES ET LES MODULES DE BASE ANSIBLE..... | 6 |
| LES PLAYBOOKS..... | 7 |
| Exercice 1 Premier Playbook..... | 7 |
| Exercice 2 Les variables..... | 7 |
| Exercice 3 Les templates..... | 8 |
| Exercice 4 debug et register..... | 8 |
| Exercice 5 Handler..... | 9 |
| Exercice 6 Les boucles..... | 9 |
| Exercice 7 La condition when..... | 10 |
| LES RÔLES..... | 11 |
| Exercice 1 Apache pour CentOS et Débian..... | 11 |
| Exercice 2 LAMP..... | 12 |
| FONCTIONNALITÉS AVANCÉES..... | 13 |
| Exercice 1 delegate_to..... | 13 |
| Exercice 2 Gather_facts..... | 13 |
| CORRECTIONS DES EXERCICES..... | 14 |
| CORRECTION - INSTALLATION ANSIBLE..... | 15 |
| CORRECTION - CONFIGURATION ET UTILISATION D'ANSIBLE..... | 16 |
| CORRECTION - LES COMMANDES ET LES MODULES DE BASE ANSIBLE..... | 17 |
| CORRECTION - LES PLAYBOOKS..... | 20 |
| Exercice 1 Premier Playbook..... | 20 |
| Exercice 2 Les variables..... | 20 |
| Exercice 3 Les templates..... | 21 |
| Exercice 4 debug et register..... | 22 |
| Exercice 5 Handler..... | 22 |
| Exercice 6 Les boucles..... | 23 |
| Exercice 7 La condition when..... | 24 |
| CORRECTION - LES RÔLES..... | 25 |
| Exercice 1 Apache pour CentOS et Débian..... | 25 |
| Exercice 2 LAMP..... | 26 |
| CORRECTION - FONCTIONNALITÉS AVANCÉES..... | 27 |
| Exercice 1 delegate_to..... | 27 |
| Exercice 2 Gather_facts..... | 27 |
| FIN DU SUPPORT DE COURS..... | 28 |

Ce document est sous Copyright :

Toute reproduction ou diffusion, même partielle, à un tiers est interdite sans autorisation écrite de Sphérius. Pour nous contacter, veuillez consulter le site web <http://www.spherius.fr>.

Les logos, marques et marques déposées sont la propriété de leurs détenteurs.

Les auteurs de ce document sont :

- Monsieur Baranger Jean-Marc,
- Monsieur Schomaker Theo.

Installation Ansible

Exercice 1

Installer Ansible en utilisant la commande yum.

Vérifier le bon fonctionnement.

Récupérer les adresses IP des clients (sur une feuille).

Noter si c'est une debian ou centos

Mettre à jour le fichier /etc/hosts.

Attention pour la notation clientX pour les centos, et debianX pour les debian

pour le stagiaire1 : client11, client12, ... et debian11, debian12, ...

pour le stagiaire2 : client21, client22, ... et debian21, debian22, ...

...

pour le stagiaire10 : client101, client102, ... et debian101, debian102, ...

Exercice 2

Création de la clé et déploiement sur les postes clients (CentOS et Débian).

Pour chaque poste, vérifier le bon fonctionnement.

- par : **ssh adresse_IP**

Répondre « yes » à la question du fingerprint.

On ne doit pas vous demander de saisir un mot de passe.

- puis par : **ssh Nom_Machine**

Répondre « yes » à la question du fingerprint.

On ne doit pas vous demander de saisir un mot de passe.

Configuration et utilisation d'Ansible

Exercice 1

Mettre à jour le fichier `/etc/ansible/hosts`

- avec un groupe nommé **clients** pour les clientX
- avec un groupe nommé **debians** pour les debianX

Vérifier le bon fonctionnement avec le module ping.

Exercice 2

Copier `/etc/ansible/hosts` en `/etc/ansible/inventaire.inv`

Pour la suite de cet exercice, vous pouvez utiliser des nom de machines qui n'existent pas.

Ajouter des groupes et des machines à votre fichier `inventaire.inv`

- groupes :
- domaine1 avec 3 machines
 - domaine2 avec 3 machines
 - windows avec 2 machines
 - linux avec domaine1 et domaine2
- ajouter quelques variables

Tester les exemples du support :

- `--lists, -m debug -a "var=groups",.`
- Afficher les valeurs d'une variable.

Tester la commande `ansible-inventory`.

Les commandes et les modules de base

Ansible

Exercice 1

Exécuter la commande suivante sur le groupe clients : `free -h`

Exécuter la commande suivante sur le groupe clients : `date > /tmp/date.res`
vérifier le contenu de ce fichier après exécution

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

Copier le fichier `/etc/hosts` de votre serveur Ansible vers `/tmp/MACHINES` des machines de votre groupe clients avec les permissions 444, avec le propriétaire user1 et le groupe user1.

Vérifier les permissions, le propriétaire et groupe propriétaire du fichier `/tmp/MACHINES`.

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

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

Pour le groupe clients : installer le package nmap
Pour le groupe clients : supprimer le package nmap

Pour toutes les machines : créer l'utilisateur user2 (uid 1234) :

- il appartiendra au groupe equipe (gid 1212)
- son répertoire de connexion : `/home/user2`
- pas de mot de passe

Pour le groupe clients : redémarrer le service crond

Les playbooks

Exercice 1 Premier Playbook

Le formateur vous fournit les fichiers `httpd.conf` et `index_install.html`.

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

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`.

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

Vérifier le fonctionnement par : `--check`

Exécuter le playbook.

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

Exercice 2 Les variables

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

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

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

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 `operateur`. Une exception concerne la machine `clientX1` (`client11`, `client21`, etc) où la valeur doit être `administrateur`.

Veillez suivre les étapes suivantes :

- Faites que la variable ai `none` pour tous les postes.
- Faites votre playbook et vous le testez.
- Faites que la variable ai `operateur` pour les postes du groupe `clients`
- Vérifier par une exécution de votre playbook.
- Faites que la valeur ai `administrateur` pour le poste `clientX1`.
- Vérifier par une exécution de votre playbook.

2 - Exceptionnellement, par exemple pour un test spécifique, vous devez exécuter votre playbook en ligne de commandes avec la variable positionnée à `TEST`.

Exercice 3 Les templates

Le formateur vous fournit les fichiers `httpd.conf` et `index_template.html`.
Créer un répertoire `apache_template`. Positionnez-vous sur ce répertoire.

1.

Le playbook à créer est `apache_template.yml`.

Il doit réaliser les mêmes opérations que l'exercice 1 et concerne le groupe `clients`. 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.

Créer le playbook qui utilisera le template pour la page de garde du site :

`/var/www/html/index.html`

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

Vérifier le fonctionnement par : `--check`

Exécuter le playbook.

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

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 :

`
 La variable LANG = ... Le contenu de la variable LANG ...`

Réexécuter votre playbook.

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

Exercice 4 debug et register

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

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` :

première version : avec `stdout`,

deuxième version : avec `stdout_lines`,

troisième version : avec `stdout_lines[1]`

quatrième version, afficher :

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

Exercice 5 Handler

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

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

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.

Pour tester :

- | | |
|---|---|
| - Première exécution | Installation ? Modification ? Redémarrage ? |
| - Deuxième exécution | Installation ? Modification ? Redémarrage ? |
| - Modification d un fichier (ajouter un commentaire) et exécution : | |
| | Installation ? Modification ? Redémarrage ? |
| - Modification des 2 fichiers (ajouter un commentaire) : | |
| | Installation ? Modification ? Redémarrage ? |
| - Nouvelle exécution | Installation ? Modification ? Redémarrage ? |

Exercice 6 Les boucles

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

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`.

Pour tester `webmin` : `https://hostname:10000` login avec `root`

Exercice 7 La condition when

1. Le service cron

Créer un répertoire `condition_when`. Positionnez-vous sur ce répertoire.
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 Débian, le service est `cron`

consulter la variable `ansible_distribution`
optionnellement, vous pouvez utiliser le filtre `lower` ou `upper`

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)

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

3.

Vous pouvez tester les exemples du support pour les filtres

Les rôles

Exercice 1 Apache pour CentOS et Débian

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

Créer un rôle d'installation d'Apache sur CentOS et Debian.

votre playbook : apache.yml

votre rôle : apache

pour les CentOS (groupe de machines clients) exploiter le playbook de l'exercice 5 des Playbooks concernant le déploiement d'Apache sur CentOS avec Handler.

pour les Débian (groupe de machines debians), le formateur vous fournit le fichier debian.yml à exploiter, ainsi que le fichier de configuration apache2.conf.

pour la copie de la page index :

retirer les lignes owner et group (sinon il y a une erreur pour les Débian).

Vérifier par : <http://clientXX> & <http://debianXX>

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

Exercice 2 LAMP

Taper :

```
# ansible-galaxy info barangerjeanmarc.site_lamp
```

Ce rôle est disponible en complément de votre rôle Apache.

Installer ce rôle :

A partir de votre répertoire racine de votre projet (lab-roles), taper la commande :

```
# ansible-galaxy install barangerjeanmarc.site_lamp --roles-path roles
```

Créer un playbook site.yml qui appelle votre rôle Apache puis celui de site_lamp.

Important :

Votre playbook devra solliciter que les machines du groupe clients (« - hosts : clients »), c'est à dire que les postes de type CentOS. En effet, le rôle site_lamp concerne que ce type d'OS.

Vous obtiendrez un site web LAMP sur chaque poste client CentOS.

Tester avec : `http://...poste.../page.php`

Fonctionnalités avancées

Exercice 1 `delegate_to`

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

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

Le playbook à créer est `apache_delegate.yml`. Il contiendra deux section « `hosts` ».

La première section concerne que le clientX1. Il y sera installé Apache, copié le fichier de configuration `httpd.conf` et enfin le service sera démarré.

La deuxième section concerne toutes les machines. Le template sera sollicité pour créer une page nommée `nom_du_client_en_cours.html` qui sera stockée dans `/var/www/html` du poste clientX1 sur lequel a été installé Apache.

Vérification par : `http://clientX1/clientXX.html` `http://clientX1/debianXX.html`

Exercice 2 `Gather_facts`

1.

Modifier le playbook précédent pour intégrer « `gather_facts: no` » sur la première section de l'installation d'Apache.

Exécuter le playbook. Contrôler le déroulement de cette exécution.

Vérifier le bon fonctionnement du site.

2.

De la modification précédente, intégrer en plus « `gather_facts: no` » sur la deuxième section de la création des pages html.

Exécuter le playbook. Contrôler le déroulement de cette exécution.

Constater le résultat sur le fonctionnement du site.

CORRECTIONS DES EXERCICES

Correction - Installation Ansible

Exercice 1

Installation :

```
# yum install epel-release
```

```
# yum install -y ansible
```

Vérification :

```
# ls /etc/ansible/  
ansible.cfg  hosts  roles
```

```
# ansible --version
```

```
# cat /etc/hosts  
. . .  
x.x.x.x  client11  
y.y.y.y  client12  
z.z.z.z  debian11
```

Exercice 2

Création de la clef :

```
# ssh-keygen -t rsa
```

```
# ls $HOME/.ssh
```

Envoi de la clef publique sur les clients :

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub x.x.x.x
```

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub y.y.y.y
```

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub z.z.z.z
```

```
# ssh x.x.x.x          yes au fingerprint  
# ssh y.y.y.y          yes au fingerprint  
# ssh z.z.z.z          yes au fingerprint
```

```
# ssh client11        yes au fingerprint  
# ssh client12        yes au fingerprint  
# ssh debian11        yes au fingerprint
```

Correction - Configuration et utilisation d'Ansible

Exercice 1

```
# cat /etc/ansible/hosts
. . .
[clients]
client11
client12

[debians]
debian11
```

```
# ansible all -m ping
```

Exercice 2

Vous avez un équivalent de ces manipulations au sein du support de cours.

Correction - Les commandes et les modules de base Ansible

Exercice 1

Exécuter la commande suivante sur le groupe clients : `free -h`

```
# ansible clients -m command -a "free -h"
# ansible client1,client2 -m command -a "free -h"
```

Exécuter la commande suivante sur le groupe clients : `date > /tmp/date.res`
vérifier le contenu de ce fichier après exécution

```
# ansible clients -m shell -a "date > /tmp/date.res"
client1 | SUCCESS | rc=0 >>
client2 | SUCCESS | rc=0 >>

# ssh client1 cat /tmp/date.res
mer. mai  9 13:42:03 CEST 2018
# ssh client1 cat /tmp/date.res
mer. mai  9 13:42:03 CEST 2018
```

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

```
# ansible clients -m shell -a "date; sleep 5; date"
client1 | SUCCESS | rc=0 >>
lun. avril 30 15:41:32 CEST 2018
lun. avril 30 15:41:37 CEST 2018

client2 | SUCCESS | rc=0 >>
lun. avril 30 15:41:32 CEST 2018
lun. avril 30 15:41:37 CEST 2018
```

Copier le fichier `/etc/hosts` de votre serveur Ansible vers `/tmp/MACHINES` des machines de votre groupe clients avec les permissions 444, avec le propriétaire `user1` et le groupe `user1`.

```
# ansible clients -m copy \
-a "src=/etc/hosts dest=/tmp/MACHINES mode=444 owner=user1 group=user1"
client2 | SUCCESS => {
  "changed": true,
  "checksum": "0406e2ba79464be3934d5deb2c410636d8491810",
  "dest": "/tmp/MACHINES",
  "gid": 1000,
  "group": "user1",
  "md5sum": "380ff798b621267fe9c241d60644ddc2",
  "mode": "0444",
  "owner": "user1",
  "secontext": "unconfined_u:object_r:admin_home_t:s0",
  "size": 242,
  "src": "/root/.ansible/tmp/ansible-tmp-1525096422.03-49918029143439/source",
  "state": "file",
  "uid": 1000
}
client1 | SUCCESS => {
  "changed": true,
  "checksum": "0406e2ba79464be3934d5deb2c410636d8491810",
```

```
"dest": "/tmp/MACHINES",
"gid": 1000,
"group": "user1",
"md5sum": "380ff798b621267fe9c241d60644ddc2",
"mode": "0444",
"owner": "user1",
"secontext": "unconfined_u:object_r:admin_home_t:s0",
"size": 242,
"src": "/root/.ansible/tmp/ansible-tmp-1525096422.03-98805615519424/source",
"state": "file",
"uid": 1000
}
```

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

```
# ansible clients -m copy -a "src=/etc/hosts dest=/tmp/MACHINES mode=444
owner=user1 group=user1"
client2 | SUCCESS => {
  "changed": false,
  "checksum": "0406e2ba79464be3934d5deb2c410636d8491810",
  "dest": "/tmp/MACHINES",
  "gid": 1000,
  "group": "user1",
  "mode": "0444",
  "owner": "user1",
  "path": "/tmp/MACHINES",
  "secontext": "unconfined_u:object_r:admin_home_t:s0",
  "size": 242,
  "state": "file",
  "uid": 1000
}
client1 | SUCCESS => {
  "changed": false,
  "checksum": "0406e2ba79464be3934d5deb2c410636d8491810",
  "dest": "/tmp/MACHINES",
  "gid": 1000,
  "group": "user1",
  "mode": "0444",
  "owner": "user1",
  "path": "/tmp/MACHINES",
  "secontext": "unconfined_u:object_r:admin_home_t:s0",
  "size": 242,
  "state": "file",
  "uid": 1000
}
```

Pour le groupe clients : supprimer le fichier /tmp/MACHINES

Attention: ne pas supprimer /etc/hosts !!!!!

```
# ansible clients -m file -a "path=/tmp/MACHINES state=absent"
# ansible clients -m file -a "dest=/tmp/MACHINES state=absent"
client2 | SUCCESS => {
  "changed": true,
  "path": "/tmp/MACHINES",
  "state": "absent"
}
client1 | SUCCESS => {
  "changed": true,
  "path": "/tmp/MACHINES",
  "state": "absent"
}
```

Pour le groupe clients : installer le package nmap.

```
# ansible clients -m yum -a "list=nmap"
# ansible clients -m yum -a "name=nmap state=present"
# ansible clients -m yum -a "list=nmap"
```

Pour le groupe clients : supprimer le package nmap.

```
# ansible clients -m yum -a "name=nmap state=absent"
```

Pour toutes les machines : créer l'utilisateur user2 (uid 1234) :

- il appartiendra au groupe equipe (gid 1212)
- son répertoire de connexion : /home/user2
- pas de mot de passe

```
# ansible all -m group -a "name=equipe gid=1212"
```

```
# ansible clients -m user -a "name=user2 uid=1234 group=equipe"
```

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
```

Correction - Les playbooks

Exercice 1 Premier Playbook

```
# cat apache_install.yml
---
- name: "Installation de Apache"
  hosts: clients
  tasks:
    - name: "Installation du package Apache"
      yum:
        name: "httpd"
        state: "present" # ou state: latest
    - name: "Installation du fichier de configuration Apache"
      copy:
        src: "httpd.conf"
        dest: "/etc/httpd/conf"
        owner: "apache"
        group: "apache"
    - name: "Installation de la page de garde"
      copy:
        src: "index_install.html"
        dest: "/var/www/html/index.html"
        owner: "apache"
        group: "apache"
    - name: "Demarrage du service Apache"
      service:
        name: "httpd"
        state: "restarted"
        enabled: yes
...
```

Exercice 2 Les variables

1.

```
# tree
.
├── group_vars
│   └── all
│       └── clients
├── host_vars
│   └── client1
└── variables.yml
```

```
# cat variables.yml
---
- hosts: all
  tasks:
    - debug: msg="La valeur pour {{inventory_hostname}} est {{compte_bdd}}"
...
```

2.

```
# ansible-playbook variable.yml -e compte_bdd=TEST
```

Exercice 3 Les templates

```
# cat index_template.html
<html>
  <head><title>Page de garde de {{inventory_hostname}}</title></head>
  <body>
    <hr>
    <center><h1>Machine {{inventory_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}}
<br> La taille de la swap est : {{ansible_memory_mb.swap.total}}
<br> La variable LANG = {{ansible_env["LANG"]}} # {{ansible_env.LANG}}
<br>
<br> La liste des adresses IP :<br>
    {% for element in ansible_all_ipv4_addresses %}
      {{element}}<br>
    {% endfor %}
<br> La liste des interfaces resaux :<br>
    {% for element in ansible_interfaces %}
      interface : {{element}}<br>
    {% endfor %}
<br><hr>
    </b></center>
  </body>
</html>
```

```
# cat apache_template.yml
- name: "Installation de Apache"
  hosts: clients
  tasks:
    - name: "Installation du package Apache"
      yum:
        name: "httpd"
        state: "present"
    - name: "Installation du fichier de configuration Apache"
      copy:
        src: "httpd.conf"
        dest: "/etc/httpd/conf/httpd.conf"
        owner: "apache"
        group: "apache"
    - name: "Installation de la page de garde"
      template:
        src: "index_template.html"
        dest: "/var/www/html/index.html"
        owner: "apache"
        group: "apache"
    - name: "Demarrage du service Apache"
      service:
        name: "httpd"
        state: "restarted"
        enabled: yes
```

Exercice 4 debug et register

Le fichier affichage.yml :

```
- hosts: all
  tasks:
    - command: free -h
      register: resultat
    - debug: var=resultat
    - debug: var=resultat.stdout
    - debug: var=resultat.stdout_lines
    - debug: var=resultat.stdout_lines[1]
```

```
- hosts: all
  tasks:
    - command: free -h
      register: resultat
    - debug: msg="Code de retour={{resultat.rc}}, Ligne={{resultat.stdout_lines[1]}}"
```

Exercice 5 Handler

```
# cat apache_handler.yml
- name: "Installation de Apache"
  hosts: clients
  handlers:
    - name: "Redemarrage du service Apache"
      service:
        name: "httpd"
        state: "started"
        enabled: yes
  tasks:
    - name: "Installation du package Apache"
      yum:
        name: "httpd"
        state: "present"
    - name: "Installation du fichier de configuration Apache"
      copy:
        src: "httpd.conf"
        dest: "/etc/httpd/conf/httpd.conf"
        owner: "apache"
        group: "apache"
      notify: "Redemarrage du service Apache"
    - name: "Installation de la page de garde"
      template:
        src: "index_template.html"
        dest: "/var/www/html/index.html"
        owner: "apache"
        group: "apache"
      notify: "Redemarrage du service Apache"
```

Exercice 6 Les 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}}"
```

Exercice 7 La condition when

1. Le service cron

```
# 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"
```

2. Apache

```
# cat apache_when.yml
- name: "Installation de Apache"
  hosts: clients
  tasks:
    - name: "Installation du package Apache"
      yum:
        name: "httpd"
        state: "present"
    - name: "Installation du fichier de configuration Apache"
      copy:
        src: "httpd.conf"
        dest: "/etc/httpd/conf/httpd.conf"
        owner: "apache"
        group: "apache"
      register: fic1
    - name: "Installation de la page de garde"
      template:
        src: "index_template.html"
        dest: "/var/www/html/index.html"
        owner: "apache"
        group: "apache"
      register: fic2
    - name: "Redemarrage du service Apache"
      service:
        name: "httpd"
        state: "started"
        enabled: yes
      when: fic1.changed or fic2.changed
```


Correction - Les rôles

Exercice 1 Apache pour CentOS et Débian

```
# mkdir lab_roles
# cd lab_roles

# mkdir roles
# cd roles

# ansible-galaxy init apache

# cd ..

# cat apache.yml
---
- hosts: all
  roles:
    - apache

# ls roles/apache/files
apache2.conf  httpd.conf

# ls roles/apache/templates
index_template.html

# ls roles/apache/tasks
debian.yml  main.yml  redhat.yml

# cat roles/apache/tasks/main.yml
---
- name: Installer et demarrer apache sur les serveurs Redhat
  include_tasks: redhat.yml
  when: ansible_os_family == 'RedHat'

- name: Installer et demarrer apache sur les serveurs Debian
  include_tasks: debian.yml
  when: ansible_os_family == 'Debian'

- name: "Installation de la page de garde"
  template:
    src: "index_template.html"
    dest: "/var/www/html/index.html"

# cat 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 Apache pour Debian"

# cat roles/apache/tasks/redhat.yml
---
- name: "Installation du package Apache"
  yum:
    name: "httpd"
    state: "present"
- name: "Installation du fichier de configuration Apache"
  copy:
    src: "httpd.conf"
    dest: "/etc/httpd/conf/httpd.conf"
    owner: "apache"
    group: "apache"
    notify: "Redemarrage du service Apache pour CentOS"
```

Exercice 2 LAMP

```
# cd lab_roles

# ansible-galaxy install \
    barangerjeanmarc.site_lamp --roles-path roles

# cat site.yml
---
- hosts: classe_clients
  roles:
    - apache
    - barangerjeanmarc.site_lamp
```

Correction - Fonctionnalités avancées

Exercice 1 delegate_to

```
# cat apache_delegate.yml
- name: "Installation de Apache sur client1"
  hosts: client1
  tasks:
    - name: "Installation du package Apache"
      yum:
        name: "httpd"
        state: "present"
    - name: "Installation du fichier de configuration Apache"
      copy:
        src: "httpd.conf"
        dest: "/etc/httpd/conf/httpd.conf"
        owner: "apache"
        group: "apache"
    - name: "Demarrage du service Apache"
      service:
        name: "httpd"
        state: "started"
        enabled: yes

- name: "Creation des pages"
  hosts: all
  tasks:
    - name: "Creation de la page {{inventory_hostname}}.html"
      template:
        src: "index_template.html"
        dest: "/var/www/html/{{inventory_hostname}}.html"
        owner: "apache"
        group: "apache"
      delegate_to: client1
```

Vérification par : <http://clientX1/clientXX.html> <http://clientX1/debianXX.html>

Exercice 2 Gather_facts

1.

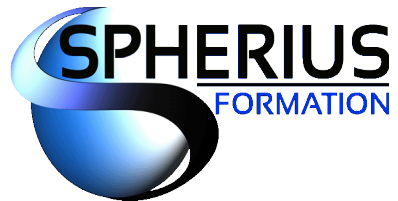
```
# cat apache_delegate.yml
- name: "Installation de Apache sur client1"
  hosts: client1
  gather_facts: no
  tasks:
    . . .
```

2.

En plus :

```
- name: "Creation des pages"
  hosts: all
  gather_facts: no
  tasks:
    . . .
```

Fin du support de cours



Votre partenaire formation ...

UNIX - LINUX - WINDOWS - ORACLE - VIRTUALISATION



www.spherius.fr