

ANSIBLE EXERCICES



SOMMAIRE

ENVIRONNEMENT DE TP.....	5
INSTALLATION ANSIBLE.....	6
Exercice 1: Mise à jour du fichier /etc/hosts.....	6
Exercice 2: Les clefs SSH.....	6
Exercice 3: Installation Ansible.....	6
CONFIGURATION ET UTILISATION D'ANSIBLE.....	7
Exercice 1: Configuration du fichier d'inventaire.....	7
Exercice 2: Afficher le contenu d'un fichier d'inventaire.....	7
LES COMMANDES AD-HOC ET LES MODULES ANSIBLE.....	8
Exercice 1: Exécution d'une commande.....	8
Exercice 2: Exécution de plusieurs commandes.....	8
Exercice 3: Copie de fichiers.....	8
Exercice 4: Suppression de fichiers.....	8
Exercice 5: Installation et suppression de packages.....	8
Exercice 6: Création d'un utilisateur.....	9
Exercice 7: Redémarrage d'un service.....	9
Exercice 8: Ouverture d'un service sur le pare-feu.....	9
LES PLAYBOOKS.....	10
Exercice 1: Installer apache.....	10
Exercice 2: Les variables.....	11
Exercice 3: Les templates.....	12
Exercice 4: debug et register.....	12
Exercice 5: Les handlers.....	13
Exercice 6: Les boucles.....	14
Exercice 7: La condition when.....	15
Exercice 7a: Le service cron.....	15
Exercice 7b: Apache.....	15
Exercice 7c: Les filtres.....	15
Exercice 7d: Installation de netdata.....	16
LES RÔLES.....	17
Exercice 1: Apache pour CentOS et Debian.....	17
Exercice 2: LAMP.....	18
FONCTIONNALITÉS AVANCÉES.....	19
Exercice 1: delegate_to.....	19
Exercice 2: gather_facts.....	19
CORRECTIONS DES EXERCICES.....	20
CORRECTION - INSTALLATION ANSIBLE.....	21
Exercice 1: Mise à jour du fichier /etc/hosts.....	21
Exercice 2: Les clefs SSH.....	21
Exercice 3: Installation Ansible.....	21
CORRECTION - CONFIGURATION ET UTILISATION D'ANSIBLE.....	22
Exercice 1: Configuration du fichier d'inventaire.....	22
Exercice 2: Afficher le contenu d'un fichier d'inventaire.....	22
CORRECTION - LES COMMANDES ET LES MODULES DE BASE ANSIBLE.....	23
Exercice 1: Exécution d'une commande.....	23
Exercice 2: Exécution de plusieurs commandes.....	23

Exercice 3: Copie de fichiers.....	24
Exercice 4: Suppression de fichiers.....	25
Exercice 5: Installation et suppression de packages.....	26
Exercice 6: Création d'un utilisateur.....	29
Exercice 7: Redémarrage d'un service.....	30
Exercice 8: Ouverture d'un service sur le pare-feu.....	30
CORRECTION - LES PLAYBOOKS.....	31
Exercice 1: Installer apache.....	31
Exercice 2: Les variables.....	33
Exercice 3: Les templates.....	36
Exercice 4: debug et register.....	39
Exercice 5: Les handlers.....	41
Exercice 6: Les boucles.....	44
Exercice 7: La condition when.....	45
Exercice 7a: Le service cron.....	45
Exercice 7b: Apache.....	46
Exercice 7d: Installation de netdata.....	47
CORRECTION - LES RÔLES.....	49
Exercice 1: Apache pour CentOS et Débian.....	49
Exercice 2: LAMP.....	52
CORRECTION - FONCTIONNALITÉS AVANCÉES.....	55
Exercice 1: delegate_to.....	55
Exercice 2: gather_facts.....	56
FIN DU SUPPORT D'EXERCICES.....	57

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.

Environnement de TP

Les TP se font dans un environnement virtualisé sous Virtualbox.

Vous avez à votre disposition 4 machines virtuelles :

- Serveur_Ansible_formation Ansible

machine serveur CentOS sur laquelle une installation avec interface graphique a été effectué. Ce sera votre machine de travail.

Les outils yamllint et ansible-lint ont été installés.

Un éditeur de texte graphique plus évolué que gedit a été installé: sublime-text.

La commande suivante permet d'éditer un fichier avec cet éditeur:

```
# subl nom_fichier
```

Le répertoire `/root/tp_sources` contient les fichiers à utiliser durant les TP.

Le répertoire `/root/tp_correction` contient les corrections des TP.

- Centos1_Formation_Ansible et Centos2_formation_Ansible

2 machines clientes CentOS sur laquelle une installation minimale a été effectué.

Le pare-feu est activé.

Les commandes suivantes permettent d'ouvrir un port sur le pare-feu:

```
# firewall-cmd --add-port=80/tcp
# firewall-cmd --reload
```

OU

```
# firewall-cmd --add-service=http
# firewall-cmd --reload
```

- Debian1_Formation_Ansible

1 machine cliente Debian1 sur laquelle une installation minimale a été effectué.

Le pare-feu est désactivé.

Pour modifier le nom d'une machine :

```
# hostnamectl set-hostname new_name
```

Pour afficher l'adresse IP d'une machine :

```
# ip a
```

Installation Ansible

Exercice 1: Mise à jour du fichier /etc/hosts

Connectez-vous sur chaque machine cliente et notez leur adresse IP ainsi que le nom de la machine (Utilisez les commandes suivantes : hostname et ip address).

Une fois que les adresses IP ont été récupérés, mettez à jour le fichier /etc/hosts du serveur.

Exercice 2: Les clefs SSH

Créer une paire de clefs RSA.

Envoyer la clef publique pour le compte root de chaque machine cliente.

Vérification : Exécuter la commande suivante pour chaque client :

```
# ssh nom_machine_client hostname
```

Exercice 3: Installation Ansible

Installer le dépôt epel, qui correspond au package epel-release.

Installer Ansible.

Vérifier la version.

Installer les packages yamllint et ansible-lint.

Configuration et utilisation d'Ansible

Exercice 1: Configuration du fichier d'inventaire

Renommer le fichier `/etc/ansible/hosts` en `/etc/ansible/hosts.origin`

Créer un nouveau fichier `/etc/ansible/hosts` avec :

- un groupe nommé **centos_servers** pour les machines CentOS,
- un groupe nommé **debian_servers** pour les machines Debian.

Vérifier le bon fonctionnement avec le module ping.

Exercice 2: Afficher le contenu d'un fichier d'inventaire

Sur le fichier `/root/tp_sources/moninventaire.inv`, tester les exemples du support de cours :

```
--list  
-m debug -a "var=groups"
```

Tester la commande `ansible-inventory`.

Si vous le souhaitez, vous pouvez modifier ce fichier d'inventaire en intégrant d'autres machines ou groupes de machines ainsi que des variables.

Les commandes Ad-Hoc et les modules Ansible

Exercice 1: Exécution d'une commande

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

Exercice 2: Exécution de plusieurs commandes

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

Exercice 3: Copie de fichiers

Copier le fichier `/etc/hosts` de votre serveur Ansible vers un fichier `/tmp/MACHINES` des machines de votre groupe centos_servers.

Le fichier `/tmp/MACHINES` aura les caractéristiques suivantes:

- permissions: 444
- propriétaire: user1
- groupe propriétaire: user1

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

Reexécuter la commande de copie et constater la différence des messages.

Exercice 4: Suppression de fichiers

Pour le groupe centos_servers: supprimer le fichier `/tmp/MACHINES`

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

Exercice 5: Installation et suppression de packages

Pour le groupe centos_servers:

- vérifier si une version du package nmap est disponible.
- vérifier si une version du package xxx est disponible (xxx est un package inexistant).
- installer le package `nmap-<numero_de_version>`.
- supprimer le package nmap.

Exercice 6: Création d'un utilisateur

Pour toutes les machines créer l'utilisateur user2 avec les caractéristiques suivantes :

- uid: 1234
- groupe primaire: equipe (gid 1212)
- répertoire de connexion : /home/user2
- pas de mot de passe

Exercice 7: Redémarrage d'un service

Pour le groupe centos_servers: redémarrer le service crond.

Exercice 8: Ouverture d'un service sur le pare-feu

Pour le groupe centos_servers:

ajouter une règle permanente pour laisser passer le service http à travers le pare-feu.

Consulter la documentation du module firewalld pour connaître les paramètres du module.

Recharger la configuration du pare-feu pour une application immédiate.

Pour cela utiliser le module service pour 'firewalld'.

Les playbooks

Exercice 1: Installer apache

Créer le répertoire /root/tp/apache_install. Positionnez-vous sur ce répertoire.
Copier les fichiers httpd.conf et index_install.html depuis le répertoire /root/tp_sources.

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

- 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:
copier le fichier index_install.html sous /var/www/html/index.html.
- Redémarrer le service apache.

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

Vérifier le fonctionnement par: `--check`

Exécuter le playbook.

Vérifier avec votre navigateur: <http://centos1> & <http://centos2>

Question subsidiaire, à la fin du playbook ajouter les actions suivantes :

- Ouvrir le service http sur le pare-feu.
- Recharger la configuration du pare-feu pour une application immédiate.

Exercice 2: Les variables

Créer le répertoire /root/tp/variables. Positionnez-vous sur ce répertoire.

Etape 1 :

Le 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 ait la valeur none pour tous les postes, sauf pour les machines du groupe centos_servers où elle correspond à operateur. Une exception concerne la machine centos2 où la valeur doit être administrateur.

Veillez suivre les étapes suivantes:

Faites que la variable ait none pour tous les postes.

Faites votre playbook et tester le.

Faites que la variable ait operateur pour les postes du groupe centos_servers.

Vérifier par une exécution de votre playbook.

Faites que la valeur ait administrateur pour le poste centos2.

Vérifier par une exécution de votre playbook.

Etape 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

Créer un répertoire /root/tp/apache_template. Positionnez-vous sur ce répertoire.
Copier les fichiers httpd.conf et index_template.html depuis le répertoire /root/tp_sources.

Etape 1 :

Créer le playbook apache_template.yml pour le groupe centos_servers.
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.
- 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`
- Exécuter le playbook.
- Vérifier par votre navigateur: <http://centos1>
 <http://centos2>

Etape 2 :

Mettre à jour le fichier index_template.html 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://centos1>
<http://centos2>

Exercice 4: debug et register

Créer un répertoire /root/tp/debug_register. Positionnez-vous sur ce répertoire.
Créer le playbook affichage.yml pour toutes les machines.

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

Le playbook doit utiliser le module debug pour:

- afficher stdout,
- afficher stdout_lines,
- afficher stdout_lines[1],
- afficher un message indiquant:
 - la commande utilisée,
 - l'option de la commande utilisée,
 - le code de retour.

Exercice 5: Les handlers

Créer le répertoire /root/tp/apache_handler. Positionnez-vous sur ce répertoire.

On reprend l'exercice Apache avec la version des templates (**exercice 3**). Copiez les fichiers depuis le répertoire /root/tp/apache_template et adaptez les. (nota : il y a une version de apache_template.yml au sein du répertoire /root/tp_sources).

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

On décide que:

- le service Apache ne doit redémarrer que si le fichier de configuration httpd.conf ou si la page web a changé.
- le firewall doit être rechargé si sa configuration a été modifiée ou si le fichier de configuration httpd.conf a changé.

Pour tester :

Exécuter le playbook selon l'ordre ci-dessous et observer ce qui est installé, modifié et redémarré.

- **1^{ère} exécution.**
- **2^{ème} exécution.**
- Modifier le fichier index_template.html (ajouter un commentaire).
- **3^{ème} exécution.**
- Modifier le fichier httpd.conf (ajouter un commentaire).
Arrêter le pare-feu sur la machine centos1 avec la commande suivante :
`ansible centos1 -m firewallld -a 'service=http permanent=yes state=disabled'`
- **4^{ème} exécution.**
- Modifier les deux fichiers (ajouter un commentaire).
Arrêter le pare-feu sur la machine centos1 avec la commande suivante :
`ansible centos1 -m firewallld -a 'service=http permanent=yes state=disabled'`
- **5^{ème} exécution.**
- Supprimer un fichier sur centos2.
- **6^{ème} exécution.**
- **7^{ème} exécution.**

Exercice 6: Les boucles

Créer un répertoire /root/tp/boucles. Positionnez-vous sur ce répertoire.

Créer le playbook boucle.yml pour toutes les machines.

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 gid.

Utiliser un tableau nommé equipe pour la liste des comptes constitué de :
nom utilisateur, uid, nom du groupe utilisateur

Exercice 7: La condition when**Exercice 7a:** Le service cron

Créer un répertoire `/root/tp/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`.

Exercice 7b: Apache

Créer un répertoire `/root/tp/apache_when`. Positionnez-vous sur ce répertoire.
On reprend l'exercice Apache avec la version des templates (**exercice 3**). Copiez les fichiers depuis le répertoire `/root/tp/apache_template` et adaptez les. (nota : il y a une version de `apache_template.yml` au sein du répertoire `/root/tp_sources`).

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

On décide que:

- le service Apache ne doit redémarrer que si le fichier de configuration `httpd.conf` ou si la page web a changé.
- le firewall doit être rechargé si sa configuration a été modifié ou si le fichier de configuration `httpd.conf` a changé.

Nota:

- on n'utilisera pas de handlers.
- pour le service
 - > `skipping` si le service n'est pas redémarré.
 - > `changed` si le service est redémarré.

Exercice 7c: Les filtres

Tester les exemples du support de cours pour les filtres.

Exercice 7d: Installation de netdata

Créer un répertoire `/root/tp/netdata`. Positionnez-vous sur ce répertoire.
Copier le script `netdata_install.sh` de `/root/tp_sources`.

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

Le playbook doit:

- Installer curl sur toutes les machines.
- Installer le dépôt EPEL sur les machines CentOS
(utiliser le module yum et le package epel-release).
- Copier le script `netdata_install.sh` dans le répertoire `/var/tmp`.
- Exécuter le script si la copie a été effectué. Le script doit-être exécuté via la commande suivante: **`bash /var/tmp/netdata_install.sh`** (utiliser un handler).
- Sur les machines CentOS :
 - ouvrir le port 19999 sur le pare-feu,
 - en cas de modification du pare-feu, recharger la configuration.

Exécuter le playbook.

Vérifier par votre navigateur: `http://nom_machine:19999`

Les rôles

Exercice 1: Apache pour CentOS et Débian

Créer un répertoire `/root/tp/lab_roles`. Positionnez-vous sur ce répertoire.
Créer un rôle d'installation d'Apache pour CentOS et Debian.

votre playbook : `apache.yml`
votre rôle: `apache`

pour les CentOS : exploiter le playbook de **l'exercice 5 des Playbooks** concernant le déploiement d'Apache sur CentOS avec Handler.

pour les Débian : les fichiers `debian.yml` et `apache2.conf` se trouvent dans le répertoire `/root/tp_sources`.

Vérifier par: <http://centos1>
<http://centos2>
<http://debian1>

```
# 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
│       │   ├── centos.yml
│       │   ├── debian.yml
│       │   └── main.yml
│       ├── templates
│       │   └── index_template.html
│       ├── tests
│       │   ├── inventory
│       │   └── test.yml
│       └── vars
│           └── main.yml
10 directories, 14 files
```

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 du site_lamp.

Vous obtiendrez un site web LAMP sur chaque poste.

Tester avec : <http://centos1/page.php>
<http://centos2/page.php>
<http://debian1/page.php>

Fonctionnalités avancées

Exercice 1: delegate_to

Créer un répertoire /root/tp/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 sections « hosts ».

La première section concerne que **centos1** pour:

- installer Apache,
- copier le fichier de configuration httpd.conf,
- redémarrer le service apache.

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 centos1 sur lequel a été installé Apache.

Vérifier par: <http://centos1/centos1.html>
 <http://centos1/centos2.html>
 <http://centos1/debian1.html>

Exercice 2: gather_facts

Etape 1

Créer un répertoire /root/tp/gather_facts. Positionnez-vous sur ce répertoire.

Recopier les fichiers de l'exercice précédent dans répertoire courant. Renommer le playbook en gather_facts_no.

Sur la **première section** de l'installation d'Apache intégrer la ligne suivante:

```
gather_facts: no
```

Exécuter le playbook. Contrôler le déroulement de cette exécution.

Vérifier le bon fonctionnement du site.

Etape 2

Sur la **deuxième section** de la création des pages html, intégrer en plus la ligne suivante:

```
gather_facts: no
```

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: Mise à jour du fichier /etc/hosts

Connectez-vous sur chaque machine cliente et notez leur adresse ip ainsi que le nom de la machine (Utilisez les commandes suivantes : hostname et ip address).

Une fois que les adresses IP ont été récupérés mettez à jour le fichier /etc/hosts du serveur.

```
# cat /etc/hosts
...
x.x.x.x centos1
y.y.y.y centos2
z.z.z.z debian1
```

Exercice 2: Les clefs SSH

Créer une paire de clefs RSA.

```
# ssh-keygen -t rsa

# ls $HOME/.ssh
```

Envoyer la clef publique pour le compte root de chaque machine cliente.

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub centos1
# ssh-copy-id -i /root/.ssh/id_rsa.pub centos2
# ssh-copy-id -i /root/.ssh/id_rsa.pub debian1
```

Vérification : Exécuter la commande suivante pour chaque client :

```
# ssh nom_machine_client hostname

# ssh centos1 hostname
# ssh centos2 hostname
# ssh debian1 hostname
```

Exercice 3: Installation Ansible

Installer le dépôt epel, qui correspond au package epel-release.

Installer Ansible.

```
# yum install -y epel-release
# yum install -y ansible
```

Vérifier la version.

```
# ansible --version
```

Installer les packages yamllint et ansible-lint.

```
# yum install -y yamllint
# yum install -y ansible-lint
```

Correction - Configuration et utilisation d'Ansible

Exercice 1: Configuration du fichier d'inventaire

Renommez le fichier /etc/ansible/hosts en /etc/ansible/hosts.origin

```
# mv /etc/ansible/hosts /etc/ansible/hosts.origin
```

Créez un nouveau fichier /etc/ansible/hosts

- avec un groupe nommé **centos_servers** pour les machines CentOS,
- avec un groupe nommé **debian_servers** pour les machines Debian.

```
# cat /etc/ansible/hosts
```

```
[centos_servers]
centos1
centos2
```

```
[debian_servers]
debian1
```

Vérifier le bon fonctionnement avec le module ping.

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

```
debian1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
centos2 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
centos1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Exercice 2: Afficher le contenu d'un fichier d'inventaire

Sur le fichier /root/tp_sources/moninventaire.inv testez les exemples du support de cours:

Tester les exemples du support:

--list, -m debug -a "var=groups".

Afficher les valeurs d'une variable.

Tester la commande ansible-inventory.

Si vous le souhaitez, vous pouvez modifier ce fichier d'inventaire en intégrant d'autres machines ou groupes de machines.

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

Correction - Les commandes et les modules de base Ansible

Exercice 1: Exécution d'une commande

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

```
# ansible centos_servers -m command -a "free -h"
# ansible centos1,centos2 -a "free -h"
centos2 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    487M        165M         21M         4,6M         300M         272M
Swap:    2,0G           0B         2,0G
centos1 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    487M        166M         22M         4,6M         298M         270M
Swap:    2,0G           8K         2,0G
```

Exercice 2: Exécution de plusieurs commandes

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

```
# ansible centos_servers -m shell -a "date; sleep 5; date"
centos1 | CHANGED | rc=0 >>
mer. janv.  2 00:43:28 CET 2019
mer. janv.  2 00:43:33 CET 2019

centos2 | CHANGED | rc=0 >>
mer. janv.  2 00:43:28 CET 2019
mer. janv.  2 00:43:33 CET 2019
```

Exercice 3: Copie de fichiers

Copier le fichier /etc/hosts de votre serveur Ansible vers un fichier /tmp/MACHINES des machines de votre groupe centos_servers.

Le fichier /tmp/MACHINES aura les caractéristiques suivantes:

- permissions: 444
- propriétaire: user1
- groupe propriétaire: user1

```
# ansible centos_servers -m copy \
-a "src=/etc/hosts dest=/tmp/MACHINES mode=444 owner=user1 group=user1"
client2 | SUCCESS => {
  "changed": true,
  "checksum": "0406e2ba79464be3934d5deb2c410636d8491010",
  "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": "0406e2ba79464be3934d5deb2c410636d8491010",
  "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
}
```

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

```
# ansible centos_servers -m command -a "ls -l /tmp/MACHINES"
```


Reexécuter la commande de copie et constater la différence des messages.

```
# ansible centos_servers -m copy \
-a "src=/etc/hosts dest=/tmp/MACHINES mode=444 owner=user1 group=user1"
centos2 | 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
}
centos1 | 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
}
```

Exercice 4: Suppression de fichiers

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

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

```
# ansible centos_servers -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"
}
```

Exercise 5: Installation et suppression de packages

vérifier si une version du package nmap est disponible.

```
# ansible centos_servers -m yum -a "list=nmap"
centos2 | SUCCESS => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": false,
  "results": [
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-16.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "16.el7",
      "repo": "base",
      "version": "6.40",
      "yumstate": "available"
    }
  ]
}
centos1 | SUCCESS => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": false,
  "results": [
    {
      "arch": "x86_64",
      "envra": "2:nmap-6.40-16.el7.x86_64",
      "epoch": "2",
      "name": "nmap",
      "release": "16.el7",
      "repo": "base",
      "version": "6.40",
      "yumstate": "available"
    }
  ]
}
```

vérifier si une version du package xxx est disponible.

```
# ansible centos_servers -m yum -a "list=xxx"
centos1 | SUCCESS => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": false,
  "results": []
}
centos2 | SUCCESS => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": false,
  "results": []
}
```

installer le package nmap-<numero_de_version>.

```
# ansible centos_servers -m yum -a "name=nmap-6.40 state=present"
centos2 | CHANGED => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Loaded plugins: fastestmirror\nLoading mirror speeds from cached hostfile\n *
base: ftp.pasteur.fr\n * epel: ftp.nluug.nl\n * extras: ftp.rezopole.net\n * updates:
ftp.pasteur.fr\nResolving Dependencies\n--> Running transaction check\n--> Package
nmap.x86_64 2:6.40-16.el7 will be installed\n--> Finished Dependency
Resolution\n\nDependencies
Resolved\n\n=====
==\n Package          Arch          Version          Repository
Size\n=====
Installing\n nmap          x86_64          2:6.40-16.el7          base          3.9
M\n\nTransaction
Summary\n=====
\nInstall 1 Package\n\nTotal download size: 3.9 M\nInstalled size: 16 M\nDownloading
packages:\nRunning transaction check\nRunning transaction test\nTransaction test
succeeded\nRunning transaction\n Installing : 2:nmap-6.40-16.el7.x86_64
1/1 \n Verifying : 2:nmap-6.40-16.el7.x86_64 1/1
\n\nInstalled:\n nmap.x86_64 2:6.40-16.el7
\n\nComplete!\n"
  ]
}
centos1 | CHANGED => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Loaded plugins: fastestmirror\nLoading mirror speeds from cached hostfile\n *
base: distrib-coffee.ipel.jussieu.fr\n * epel: ftp.nluug.nl\n * extras: distrib-
coffee.ipel.jussieu.fr\n * updates: distrib-coffee.ipel.jussieu.fr\nResolving
Dependencies\n--> Running transaction check\n--> Package nmap.x86_64 2:6.40-16.el7 will
be installed\n--> Finished Dependency Resolution\n\nDependencies
Resolved\n\n=====
==\n Package          Arch          Version          Repository
Size\n=====
Installing\n nmap          x86_64          2:6.40-16.el7          base          3.9
M\n\nTransaction
Summary\n=====
\nInstall 1 Package\n\nTotal download size: 3.9 M\nInstalled size: 16 M\nDownloading
packages:\nRunning transaction check\nRunning transaction test\nTransaction test
succeeded\nRunning transaction\n Installing : 2:nmap-6.40-16.el7.x86_64
1/1 \n Verifying : 2:nmap-6.40-16.el7.x86_64 1/1
\n\nInstalled:\n nmap.x86_64 2:6.40-16.el7
\n\nComplete!\n"
  ]
}
```

Vérification.

```
# ansible centos_servers -m yum -a "list=nmap"
```

Pour le groupe centos_servers: supprimer le package nmap

```
# ansible centos_servers -m yum -a "name=nmap state=absent"
centos2 | CHANGED => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Modules complémentaires chargés : fastestmirror\nRésolution des dépendances\n-->
    Lancement de la transaction de test\n--> Le paquet nmap.x86_64 2:6.40-16.el7 sera
    effacé\n--> Résolution des dépendances terminée\n\nDépendances
    résolues\n\n=====
    ==\n Package           Architecture      Version           Dépôt
    Taille\n=====
    nSuppression :\n nmap             x86_64           2:6.40-16.el7     @base
    16 M\n\nRésumé de la
    transaction\n=====
    ==\nSupprimer 1 Paquet\n\nTaille d'installation : 16 M\nDownloading packages:\nRunning
    transaction check\nRunning transaction test\nTransaction test succeeded\nRunning
    transaction\n Suppression : 2:nmap-6.40-16.el7.x86_64
    1/1 \n Vérification : 2:nmap-6.40-16.el7.x86_64
    \n\nSupprimé :\n nmap.x86_64 2:6.40-16.el7
    \n\nTerminé !\n"
  ]
}
centos1 | CHANGED => {
  "ansible_facts": {
    "pkg_mgr": "yum"
  },
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Modules complémentaires chargés : fastestmirror\nRésolution des dépendances\n-->
    Lancement de la transaction de test\n--> Le paquet nmap.x86_64 2:6.40-16.el7 sera
    effacé\n--> Résolution des dépendances terminée\n\nDépendances
    résolues\n\n=====
    ==\n Package           Architecture      Version           Dépôt
    Taille\n=====
    nSuppression :\n nmap             x86_64           2:6.40-16.el7     @base
    16 M\n\nRésumé de la
    transaction\n=====
    ==\nSupprimer 1 Paquet\n\nTaille d'installation : 16 M\nDownloading packages:\nRunning
    transaction check\nRunning transaction test\nTransaction test succeeded\nRunning
    transaction\n Suppression : 2:nmap-6.40-16.el7.x86_64
    1/1 \n Vérification : 2:nmap-6.40-16.el7.x86_64
    \n\nSupprimé :\n nmap.x86_64 2:6.40-16.el7
    \n\nTerminé !\n"
  ]
}
}
```

Vérification.

```
# ansible centos_servers -m yum -a "list=nmap"
```

Exercise 6: Création d'un utilisateur

Pour toutes les machines créer l'utilisateur user2 avec les caractéristiques suivantes:

- uid: 1234
- groupe primaire: equipe (gid 1212)
- répertoire de connexion : /home/user2
- pas de mot de passe

```
# ansible all -m group -a "name=equipe gid=1212"
debian1 | CHANGED => {
  "changed": true,
  "gid": 1212,
  "name": "equipe",
  "state": "present",
  "system": false
}
centos1 | CHANGED => {
  "changed": true,
  "gid": 1212,
  "name": "equipe",
  "state": "present",
  "system": false
}
centos2 | CHANGED => {
  "changed": true,
  "gid": 1212,
  "name": "equipe",
  "state": "present",
  "system": false
}
# ansible all -m user -a "name=user2 uid=1234 group=equipe"
debian1 | CHANGED => {
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1212,
  "home": "/home/user2",
  "name": "user2",
  "shell": "",
  "state": "present",
  "system": false,
  "uid": 1234
}
centos1 | CHANGED => {
  "append": false,
  "changed": true,
  "comment": "",
  "group": 1212,
  "home": "/home/user2",
  "move_home": false,
  "name": "user2",
  "shell": "/bin/bash",
  "state": "present",
  "uid": 1234
}
centos2 | CHANGED => {
  "append": false,
  "changed": true,
  "comment": "",
  "group": 1212,
  "home": "/home/user2",
  "move_home": false,
  "name": "user2",
  "shell": "/bin/bash",
  "state": "present",
  "uid": 1234
}
```

Exercise 7: Redémarrage d'un service

Pour le groupe `centos_servers`: redémarrer le service `crond`.

```
# ssh centos1 pgrep -l crond
633 crond
# ansible centos_servers -m service -a "name=crond state=restarted"
centos2 | CHANGED => {
  "changed": true,
  "name": "crond",
  "state": "started",
  "status": {
    "ActiveEnterTimestamp": "mar. 2019-01-01 12:56:33 CET",
    "ActiveEnterTimestampMonotonic": "54941563",
    ...
    "WatchdogTimestampMonotonic": "54941496",
    "WatchdogUSec": "0"
  }
}
centos1 | CHANGED => {
  "changed": true,
  "name": "crond",
  "state": "started",
  "status": {
    "ActiveEnterTimestamp": "mar. 2019-01-01 12:55:52 CET",
    "ActiveEnterTimestampMonotonic": "42248905",
    ...
    "WatchdogTimestampMonotonic": "42248857",
    "WatchdogUSec": "0"
  }
}
# ssh centos1 pgrep -l crond
2820 crond
```

Exercise 8: Ouverture d'un service sur le pare-feu

Pour le groupe `centos_servers`:

ajouter une règle permanente pour laisser passer le service `http` à travers le pare-feu.

Consultez la documentation du module `firewalld` pour connaître les paramètres du module.

```
# ansible centos_servers -m firewalld -a 'service=http permanent=yes state=enabled'
centos2 | CHANGED => {
  "changed": true,
  "msg": "Permanent operation, Changed service http to enabled"
}
centos1 | CHANGED => {
  "changed": true,
  "msg": "Permanent operation, Changed service http to enabled"
}
```

Recharger la configuration du pare-feu pour une application immédiate.

Pour cela utiliser le module `service` pour `'firewalld'`.

```
# ansible centos_servers -m service -a 'name=firewalld state=reloaded'
```

Correction - Les playbooks

Exercise 1: Installer apache

```
# mkdir -p /root/tp/apache_install
# cp /root/tp_sources/httpd.conf /root/tp/apache_install
# cp /root/tp_sources/index_install.html /root/tp/apache_install
# cd /root/tp/apache_install

# cat apache_install.yml
---
- name: "Installation d'Apache"
  hosts: centos_servers
  tasks:
    - name: "Installation du package Apache"
      yum:
        name: "httpd"
        state: "present"

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

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

    - name: "Redemarrage du service Apache"
      service:
        name: "httpd"
        state: "restarted"
        enabled: "true"

    - name: "Regle pare-feu pour http"
      firewallld:
        service: "http"
        permanent: "true"
        state: "enabled"

    - name: "Rechargement du pare-feu"
      service:
        name: "firewalld"
        state: "reloaded"
...

```

Exécution du playbook.

```
# ansible-playbook apache_install.yml
PLAY [Installation d'Apache] *****

TASK [Gathering Facts] *****
ok: [centos2]
ok: [centos1]

TASK [Installation du package Apache] *****
changed: [centos1]
changed: [centos2]

TASK [Copie du fichier httpd.conf] *****
changed: [centos2]
changed: [centos1]

TASK [Installation de la page de garde] *****
changed: [centos1]
changed: [centos2]

TASK [Redemarrage du service Apache] *****
changed: [centos1]
changed: [centos2]

TASK [Regle pare-feu pour http] *****
ok: [centos2]
ok: [centos1]

TASK [Rechargement du pare-feu] *****
changed: [centos1]
changed: [centos2]

PLAY RECAP *****
centos1      : ok=7    changed=5    unreachable=0    failed=0
centos2      : ok=7    changed=5    unreachable=0    failed=0
```


Exercice 2: Les variables

```
# mkdir /root/tp/variables
# cd /root/tp/variables
```

Etape 1

L'arborescence finale.

```
# tree
.
├── group_vars
│   ├── all
│   └── centos_servers
├── host_vars
│   └── centos1
└── variables.yml
```

Le playbook.

```
# cat variables.yml
---
- name: Afficher la valeur d'une variable
  hosts: all
  tasks:
    - name: "Utilisation du module debug pour afficher ma variable"
      debug:
        msg: "La valeur pour compte_bdd de {{inventory_hostname}} est {{compte_bdd}}"
...
```

Création du fichier **all** dans le répertoire **group_vars**.

```
# cat group_vars/all
---
compte_bdd: none
```

Exécution du playbook.

```
# ansible-playbook variables.yml

PLAY [Afficher la valeur d'une variable] *****

TASK [Gathering Facts] *****
ok: [debian1]
ok: [centos2]
ok: [centos1]

TASK [Utilisation du module debug pour afficher ma variable] *****
ok: [centos1] => {
  "msg": "La valeur pour compte_bdd de centos1 est none"
}
ok: [centos2] => {
  "msg": "La valeur pour compte_bdd de centos2 est none"
}
ok: [debian1] => {
  "msg": "La valeur pour compte_bdd de debian1 est none"
}

PLAY RECAP *****
centos1      : ok=2    changed=0    unreachable=0    failed=0
centos2      : ok=2    changed=0    unreachable=0    failed=0
debian1      : ok=2    changed=0    unreachable=0    failed=0
```

Création du fichier **centos_servers** dans le répertoire **group_vars**.

```
# cat group_vars/centos_servers
```

```
---
compte_bdd: operateur
```

Exécution du playbook.

```
# ansible-playbook variables.yml
```

```
PLAY [Afficher la valeur d'une variable] *****

TASK [Gathering Facts] *****
ok: [debian1]
ok: [centos2]
ok: [centos1]

TASK [Utilisation du module debug pour afficher ma variable] *****
ok: [centos1] => {
  "msg": "La valeur pour compte_bdd de centos1 est operateur"
}
ok: [centos2] => {
  "msg": "La valeur pour compte_bdd de centos2 est operateur"
}
ok: [debian1] => {
  "msg": "La valeur pour compte_bdd de debian1 est none"
}

PLAY RECAP *****
centos1           : ok=2    changed=0    unreachable=0    failed=0
centos2           : ok=2    changed=0    unreachable=0    failed=0
debian1           : ok=2    changed=0    unreachable=0    failed=0
```

Création du fichier **centos_servers** dans le répertoire **host_vars**.

```
# cat host_vars/centos1
```

```
---
compte_bdd: administrateur
```

Exécution du playbook.

```
# ansible-playbook variables.yml
```

```
PLAY [Afficher la valeur d'une variable] *****

TASK [Gathering Facts] *****
ok: [debian1]
ok: [centos1]
ok: [centos2]

TASK [Utilisation du module debug pour afficher ma variable] *****
ok: [centos1] => {
  "msg": "La valeur pour compte_bdd de centos1 est administrateur"
}
ok: [centos2] => {
  "msg": "La valeur pour compte_bdd de centos2 est operateur"
}
ok: [debian1] => {
  "msg": "La valeur pour compte_bdd de debian1 est none"
}

PLAY RECAP *****
centos1           : ok=2    changed=0    unreachable=0    failed=0
centos2           : ok=2    changed=0    unreachable=0    failed=0
debian1           : ok=2    changed=0    unreachable=0    failed=0
```

Etape 2

Exécution du playbook.

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

PLAY [Afficher la valeur d'une variable] *****

TASK [Gathering Facts] *****
ok: [debian1]
ok: [centos2]
ok: [centos1]

TASK [Utilisation du module debug pour afficher ma variable] *****
ok: [centos1] => {
  "msg": "La valeur pour compte_bdd de centos1 est TEST"
}
ok: [centos2] => {
  "msg": "La valeur pour compte_bdd de centos2 est TEST"
}
ok: [debian1] => {
  "msg": "La valeur pour compte_bdd de debian1 est TEST"
}

PLAY RECAP *****
centos1      : ok=2    changed=0    unreachable=0    failed=0
centos2      : ok=2    changed=0    unreachable=0    failed=0
debian1      : ok=2    changed=0    unreachable=0    failed=0
```

Exercice 3: Les templates

```
# mkdir /root/tp/apache_template
# cd /root/tp/apache_template
```

Copie des fichiers sources.

```
# cp /root/tp_sources/httpd.conf .
# cp /root/tp_sources/index_template.html .
```

Modification du fichier index_template.html.

```
# cat index_template.html
<html>
  <head><title>Page de garde de {{inventory_hostname}}</title></head>
  <body>
    <hr>
    <center><h1>Machine {{inventory_hostname}}</h1></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_memory_mb.swap.total}} Mo
<br> La variable LANG contient: {{ansible_env["LANG"]}} # ou {{ansible_env.LANG}}
<br>
<br> La liste des adresses IP :<br>
    {% for element in ansible_all_ipv4_addresses %}
      {{element}}<br>
    {% endfor %}
<br>
<br> La liste des interfaces reseaux :<br>
    {% for element in ansible_interfaces %}
      interface : {{element}}<br>
    {% endfor %}
<br><hr>
    </b></center>
  </body>
</html>
```

Copie du playbook de l'exercice1.

```
# cp ../apache/apache_install.yml apache_template.yml
```

Adaptation du playbook pour le template.

```
# cat apache_template.yml
---
- name: Installation d'Apache
  hosts: centos_servers
  tasks:
    - name: Installation du package Apache
      yum:
        name: httpd
        state: present

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

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

    - name: Redemarrage du service Apache
      service:
        name: httpd
        state: restarted
        enabled: true

    - name: Regle pare-feu pour http
      firewallld:
        service: http
        permanent: true
        state: enabled

    - name: Rechargement du pare-feu
      service:
        name: firewallld
        state: reloaded
...

```

Exécution du playbook.

```
# ansible-playbook apache_template.yml

PLAY [Installation d'Apache] *****

TASK [Gathering Facts] *****
ok: [centos2]
ok: [centos1]

TASK [Installation du package Apache] *****
ok: [centos1]
ok: [centos2]

TASK [Copie du fichier httpd.conf] *****
ok: [centos1]
ok: [centos2]

TASK [Installation de la page de garde] *****
changed: [centos2]
changed: [centos1]

TASK [Redemarrage du service Apache] *****
changed: [centos1]
changed: [centos2]

TASK [Regle pare-feu pour http] *****
ok: [centos1]
ok: [centos2]

TASK [Rechargement du pare-feu] *****
changed: [centos2]
changed: [centos1]

PLAY RECAP *****
centos1      : ok=7    changed=3    unreachable=0    failed=0
centos2      : ok=7    changed=3    unreachable=0    failed=0
```

Exercice 4: debug et register

```
# mkdir /root/tp/debug_register
# cd /root/tp/debug_register
```

Le playbook affichage.yml.

```
# cat affichage.yml
---
- name: Test du module debug
  hosts: all
  tasks:
    - name: Exécution de free -h
      command: free -h
      register: result

    - name: Afficher stdout
      debug:
        var: result.stdout

    - name: Afficher stdout_lines
      debug:
        var: result.stdout_lines

    - name: Afficher stdout_lines[1]
      debug:
        var: result.stdout_lines[1]

    - name: Afficher message
      debug:
        msg: Commande exécutée = {{result.cmd[0]}}      Option utilisée =
{{result.cmd[1]}}      Code de retour = {{result.rc}}
...

```

Exécution du playbook.

```
# ansible-playbook affichage.yml

PLAY [Test du module debug] *****

TASK [Gathering Facts] *****
ok: [debian1]
ok: [centos2]
ok: [centos1]

TASK [Exécution de free -h] *****
changed: [debian1]
changed: [centos1]
changed: [centos2]

TASK [Afficher stdout] *****
ok: [centos1] => {
  "result.stdout": "
available\ Mem:           487M          total          used          free          shared  buff/cache
284M\ nSwap:           2,0G          7,8M          2,0G"
}
ok: [centos2] => {
  "result.stdout": "
available\ Mem:           487M          total          used          free          shared  buff/cache
281M\ nSwap:           2,0G          7,0M          2,0G"
}

```

```

}
ok: [debian1] => {
    "result.stdout": "
                total      used      free      shared  buff/cache
available\
Mem:          2,0G      110M      1,4G      5,7M      475M
1,7G\
Swap:          2,0G         0B      2,0G"
}

TASK [Afficher stdout_lines] *****
ok: [centos1] => {
    "result.stdout_lines": [
        "
                total      used      free      shared  buff/cache      available",
        "Mem:          487M      156M      68M      2,3M      262M      284M",
        "Swap:          2,0G      7,8M      2,0G"
    ]
}
ok: [centos2] => {
    "result.stdout_lines": [
        "
                total      used      free      shared  buff/cache      available",
        "Mem:          487M      159M      61M      2,1M      266M      281M",
        "Swap:          2,0G      7,0M      2,0G"
    ]
}
ok: [debian1] => {
    "result.stdout_lines": [
        "
                total      used      free      shared  buff/cache      available",
        "Mem:          2,0G      110M      1,4G      5,7M      475M      1,7G",
        "Swap:          2,0G         0B      2,0G"
    ]
}

TASK [Afficher stdout_lines[1]] *****
ok: [centos1] => {
    "result.stdout_lines[1]": "Mem:          487M      156M      68M      2,3M      262M      284M"
}
ok: [centos2] => {
    "result.stdout_lines[1]": "Mem:          487M      159M      61M      2,1M      266M      281M"
}
ok: [debian1] => {
    "result.stdout_lines[1]": "Mem:          2,0G      110M      1,4G      5,7M      475M      1,7G"
}

TASK [Afficher message] *****
ok: [centos1] => {
    "msg": "Commande exécutée = free          Option utilisée = -h          Code de retour = 0"
}
ok: [centos2] => {
    "msg": "Commande exécutée = free          Option utilisée = -h          Code de retour = 0"
}
ok: [debian1] => {
    "msg": "Commande exécutée = free          Option utilisée = -h          Code de retour = 0"
}

PLAY RECAP *****
centos1          : ok=6    changed=1    unreachable=0    failed=0
centos2          : ok=6    changed=1    unreachable=0    failed=0
debian1         : ok=6    changed=1    unreachable=0    failed=0

```


Exercise 5: Les handlers

```
# mkdir /root/tp/apache_handler
# cd /root/tp/apache_handler
```

Copie des fichiers.

```
# cp ../apache_template/httpd.conf .
# cp ../apache_template/index_template.html .
# cp ../apache_template/apache_template.yml apache_handler.yml
```

Modification du fichier apache_handler.yml.

```
# cat apache_handler.yml
---
- name: Installation d'Apache
  hosts: centos_servers

  handlers:
    - name: Redemarrage du service Apache
      service:
        name: httpd
        state: restarted
        enabled: true

    - name: Rechargement du pare-feu
      service:
        name: firewalld
        state: reloaded

  tasks:
    - name: Installation du package Apache
      yum:
        name: httpd
        state: present

    - name: Copie du fichier httpd.conf
      copy:
        src: httpd.conf
        dest: /etc/httpd/conf/httpd.conf
        notify: Redemarrage du service Apache

    - name: Installation de la page de garde
      template:
        src: index_template.html
        dest: /var/www/html/index.html
        notify: Redemarrage du service Apache

    - name: Regle pare-feu pour http
      firewalld:
        service: http
        permanent: true
        state: enabled
        notify: Rechargement du pare-feu
...

```

Exécution du playbook.

```
# ansible-playbook apache_handler.yml

PLAY [Installation d'Apache] *****

TASK [Gathering Facts] *****
ok: [centos2]
ok: [centos1]

TASK [Installation du package Apache] *****
ok: [centos2]
ok: [centos1]

TASK [Copie du fichier httpd.conf] *****
ok: [centos2]
ok: [centos1]

TASK [Installation de la page de garde] *****
ok: [centos1]
ok: [centos2]

TASK [Regle pare-feu pour http] *****
ok: [centos2]
ok: [centos1]

PLAY RECAP *****
centos1      : ok=5    changed=0    unreachable=0    failed=0
centos2      : ok=5    changed=0    unreachable=0    failed=0
```

4^{ème} exécution.

Modifier les deux fichiers (ajouter un commentaire).

```
# echo '#' >> index_template.html
# echo '#' >> httpd.conf
```

Arretez le pare-feu sur la machine centos1.

```
# ansible centos1 -m firewallld -a 'service=http permanent=yes
state=disabled'
centos1 | CHANGED => {
  "changed": true,
  "msg": "Permanent operation, Changed service http to disabled"
}
```

Ré-exécution du playbook.

```
# ansible-playbook apache_handler.yml

PLAY [Installation d'Apache] *****

TASK [Gathering Facts] *****
ok: [centos1]
ok: [centos2]

TASK [Installation du package Apache] *****
ok: [centos2]
ok: [centos1]

TASK [Copie du fichier httpd.conf] *****
changed: [centos1]
changed: [centos2]

TASK [Installation de la page de garde] *****
changed: [centos1]
changed: [centos2]

TASK [Regle pare-feu pour http] *****
ok: [centos2]
changed: [centos1]

RUNNING HANDLER [Redemarrage du service Apache] *****
changed: [centos1]
changed: [centos2]

RUNNING HANDLER [Rechargement du pare-feu] *****
changed: [centos1]

PLAY RECAP *****
centos1      : ok=7    changed=5    unreachable=0    failed=0
centos2      : ok=6    changed=3    unreachable=0    failed=0
```

Exercice 6: Les boucles

```
# mkdir /root/tp/boucles
# cd /root/tp/boucles
```

Le playbook boucle.yml

```
# cat boucle.yml
---
- name: Création d'utilisateurs
  hosts: centos_servers
  vars:
    grp_name: team
    grp_gid: 1100
    equipe:
      - {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: Création du groupe
      group:
        state: present
        name: "{{grp_name}}"
        gid: "{{grp_gid}}"
    - name: Création des users
      user:
        state: present
        name: "{{item.nom}}"
        uid: "{{item.uid}}"
        group: "{{item.groupe}}"
        home: /home/"{{item.nom}}"
        create_home: true
      loop: "{{equipe}}"
...
```

Exécution du playbook.

```
# ansible-playbook boucle.yml

PLAY [Création d'utilisateurs] *****

TASK [Gathering Facts] *****
ok: [centos2]
ok: [centos1]

TASK [Création du groupe] *****
changed: [centos2]
changed: [centos1]

TASK [Création des users] *****
changed: [centos2] => (item={u'groupe': 1100, u'nom': u'jean', u'uid': 1101})
changed: [centos1] => (item={u'groupe': 1100, u'nom': u'jean', u'uid': 1101})
changed: [centos2] => (item={u'groupe': 1100, u'nom': u'marc', u'uid': 1002})
changed: [centos1] => (item={u'groupe': 1100, u'nom': u'marc', u'uid': 1002})
changed: [centos2] => (item={u'groupe': 1100, u'nom': u'eve', u'uid': 1003})
changed: [centos1] => (item={u'groupe': 1100, u'nom': u'eve', u'uid': 1003})
changed: [centos2] => (item={u'groupe': 1100, u'nom': u'celine', u'uid': 1004})
changed: [centos1] => (item={u'groupe': 1100, u'nom': u'celine', u'uid': 1004})

PLAY RECAP *****
centos1 : ok=3 changed=2 unreachable=0 failed=0
centos2 : ok=3 changed=2 unreachable=0 failed=0
```

Exercice 7: La condition when**Exercice 7a:** Le service cron

```
# mkdir /root/tp/condition_when
# cd /root/tp/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"
```

[Exercise 7b:](#) Apache

```
# mkdir /root/tp/apache_when
# cd /root/tp/apache_when
# cp /root/tp/apache_template/* .
```

```
# cat apache_when.yml
---
- name: Installation de Apache
  hosts: centos_servers
  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
        register: conf_http

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

    - name: Redemarrage du service Apache
      service:
        name: httpd
        state: started
        enabled: true
      when: conf_http.changed or index_http.changed

    - name: Regle pare-feu pour http
      firewallld:
        service: http
        permanent: true
        state: enabled
        register: fire

    - name: Rechargement du pare-feu
      service:
        name: firewallld
        state: reloaded
      when: conf_http.changed or fire.changed
  ...

ou when: conf_http is changed or index_http is changed
ou when: conf_http.changed == true or index_http.changed == true
```

Exercise 7d: Installation de netdata

```
# mkdir /root/tp/netdata
# cd /root/tp/netdata
# cp /root/tp_sources/netdata_install.sh .
```

Le playbook netdata.yml.

```
# cat netdata.yml
---
- name: Installation de netdata
  hosts: all

  handlers:
    - name: Executer le script
      command: bash /var/tmp/netdata_install.sh

    - name: Rechargement du pare-feu
      service:
        name: firewalld
        state: reloaded
  tasks:
    - name: Installation du dépôt EPEL
      yum:
        name: epel-release
        state: present
      when: ansible_distribution|lower == "centos"

    - name: installation de curl
      package:
        name: curl
        state: present

    - name: Transfert du script
      copy:
        src: netdata_install.sh
        dest: /var/tmp
        mode: 0755
      notify: Executer le script

    - name: Ouvrir le port 19999
      firewalld:
        port: 19999/tcp
        permanent: true
        state: enabled
      when: ansible_distribution|lower == "centos"
      notify: Rechargement du pare-feu
...

```

Exécuter le playbook.

```
# ansible-playbook netdata.yml

PLAY [Installation de netdata] *****

TASK [Gathering Facts] *****
ok: [debian1]
ok: [centos2]
ok: [centos1]

TASK [Installation du dépôt EPEL] *****
skipping: [debian1]
changed: [centos1]
changed: [centos2]

TASK [installation de curl] *****
ok: [centos1]
ok: [centos2]
changed: [debian1]

TASK [Transfert du script] *****
changed: [debian1]
changed: [centos2]
changed: [centos1]

TASK [Ouvrir le port 19999] *****
skipping: [debian1]
changed: [centos2]
changed: [centos1]

RUNNING HANDLER [Executer le script] *****
changed: [debian1]
changed: [centos1]
changed: [centos2]

RUNNING HANDLER [Recharger le pare-feu] *****
changed: [centos2]
changed: [centos1]

PLAY RECAP *****
centos1      : ok=7    changed=5    unreachable=0    failed=0
centos2      : ok=7    changed=5    unreachable=0    failed=0
debian1      : ok=4    changed=3    unreachable=0    failed=0
```

<http://centos1:19999>

<http://centos2:19999>

<http://debian1:19999>

Correction - Les rôles

Exercice 1: Apache pour CentOS et Débian

```
# mkdir /root/tp/lab_roles
# cd /root/tp/lab_roles
```

Le playbook apache.yml.

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

Création du rôle apache vierge.

```
# mkdir roles
# cd roles

# ansible-galaxy init apache
- apache was created successfully

# cd apache
# ls
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars
```

Copie des fichiers de configuration.

```
# cp /root/tp_sources/httpd.conf files
# cp /root/tp_sources/apache2.conf files
# ls files
apache2.conf  httpd.conf
```

Copie du fichier template.

```
# cp /root/tp/apache_handler/index_template.html templates/
# ls templates
index_template.html
```

Copie des fichiers.

```
# cp /root/tp_sources/debian.yml tasks
# cp /root/tp/apache_handler/apache_handler.yml tasks/centos.yml
# ls tasks
centos.yml  debian.yml  main.yml
```

Le fichier de taches principal.

```
# cat tasks/main.yml
---
# tasks file for apache

- name: Installer Apache sur les serveurs Centos
  include_tasks: centos.yml
  when: ansible_distribution|lower == 'centos'

- name: Installer Apache sur les serveurs Debian
  include_tasks: debian.yml
  when: ansible_distribution|lower == 'debian'
...
```

Le fichier d'instructions pour debian.

```
# cat tasks/debian.yml
---
- name: Installer les outils de developpement
  apt:
    name: build-essential
    state: present

- name: Installer apache2
  apt:
    name: apache2
    state: present

- name: Copier apache2.conf
  copy:
    src: apache2.conf
    dest: /etc/apache2/apache2.conf
  notify: Redemarrer Apache Debian

- name: Installer la page de garde / Copier le template
  template:
    src: index_template.html
    dest: /var/www/html/index.html
  notify: Redemarrer Apache Debian
...
```

Le fichier d'instructions pour centos.

```
# cat tasks/centos.yml
---
- name: Installer httpd
  yum:
    name: httpd
    state: present

- name: Copier httpd.conf
  copy:
    src: httpd.conf
    dest: /etc/httpd/conf/httpd.conf
  notify: Redemarrer Apache CentOS

- name: Installer la page de garde / Copier le template
  template:
    src: index_template.html
    dest: /var/www/html/index.html
  notify: Redemarrer Apache CentOS

- name: Regle pare-feu pour http
  firewallld:
    service: http
    permanent: true
    state: enabled
  notify: Recharger Pare-feu
...
```

Le fichier des handlers.

```
# cat handlers/main.yml
---
# handlers file for apache
- name: Redemarrer Apache Centos
  service:
    name: httpd
    state: restarted
    enabled: true

- name: Rechargement du pare-feu
  service:
    name: firewalld
    state: reloaded

- name: Redemarrer Apache Debian
  service:
    name: apache2
    state: restarted
    enabled: true
...
```

Exercise 2: LAMP

```
# cd /root/tp/lab_roles

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

Le playbook.

```
# cat site.yml:w
---
- name: Installation de LAMP
  hosts: all
  roles:
    - apache
    - barangerjeanmarc.site_lamp
...
```

Le contenu du rôle.

```
# tree roles/barangerjeanmarc.site_lamp
roles/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
│   ├── centos_vars.yml
│   ├── debian_vars.yml
│   └── main.yml
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

Le fichier de tâches principal.

```
# cat roles/barangerjeanmarc.site_lamp/tasks/main.yml
---
# tasks file for site_lamp
- name: Inclure les variables debian
  include_vars: debian_vars.yml
  when: ansible_distribution|lower == 'debian'
- name: Inclure les variables centos
  include_vars: centos_vars.yml
  when: ansible_distribution|lower == 'centos'
- name: Installer MariaDB
  include_tasks: bdd_install.yml
  notify: Redemarrer Apache
- name: Installer base de données PHP
  include_tasks: bdd_php.yml
  notify: Redemarrer Apache
- name: Injecter la base de donnees
  include_tasks: bdd_manip.yml
  notify: Redemarrer Apache
...
```

Les variables debian.

```
# cat roles/barangerjeanmarc.site_lamp/tasks/debian_vars.yml
---
service_apache: apache2
liste_package_mariadb:
  - mariadb-server
  - mariadb-client
  - python-mysqldb
...
```

Les variables centos.

```
# cat roles/barangerjeanmarc.site_lamp/tasks/centos_vars.yml
---
service_apache: httpd
liste_package_mariadb:
  - mariadb-server
  - mariadb
  - MySQL-python
...
```

Installation de la base de donnée.

```
# cat roles/barangerjeanmarc.site_lamp/tasks/bdd_install.yml
---
- block:
  - name: Installer la clef GPG pour les serveurs Debian
    apt_key:
      url: http://keyserver.ubuntu.com/pks/lookup?op=get&search=0x{{ mariadb.gpg_key_id }}
    state: present

  - name: Ajouter le dépôt MariaDB pour les serveurs Debian
    apt_repository:
      repo: deb http://{{ mariadb.mirror }}/mariadb/repo/
      {{ mariadb.series_version }}/debian {{ debian.version }} main
    state: present
    when: ansible_distribution|lower == 'debian'

- name: Installer MariaDB
  package:
    name: "{{ item }}"
    state: present
  loop: "{{ liste_package_mariadb }}"

- name: Démarrer MariaDB
  service:
    name: mariadb
    state: started
    enabled: true
...
```

Installation de php.

```
# cat roles/barangerjeanmarc.site_lamp/tasks/bdd_php.yml
---
- name: Installation de PHP et des extensions
  package:
    name:
      - php
      - php-pdo
      - php-mysql
      - php-soap
      - php-gd
    state: present
  notify:
    - Redemarrer Apache
...
```

Importation des données.

```
# cat roles/barangerjeanmarc.site_lamp/tasks/bdd_manip.yml
---
- name: Créer la base de données
  mysql_db:
    name: "{{dbname}}"
    state: present

- name: Créer l'utilisateur pour la base de données
  mysql_user: name={{dbuser}} password={{upassword}}
              priv=*.*:ALL host='%' state=present

- name: Copier le fichier annuaire
  copy:
    src: annuaire.sql
    dest: /tmp/annuaire.sql

- name: Importer les données
  mysql_db: name=annuairedb state=import target=/tmp/annuaire.sql

- name: Copier la page PHP
  copy:
    src: page.php
    dest: /var/www/html/page.php
  notify: Redemarrer Apache
...
```

Les handlers du rôle.

```
# cat roles/barangerjeanmarc.site_lamp/handlers/main.yml
---
# handlers file for site_lamp
- name: Redemarrer Apache
  service:
    name: "{{ service_apache }}"
    state: restarted
    enabled: true

- name: Redemarrer MariaDB
  service:
    name: mariadb
    state: restarted
    enabled: true
...
```

Correction - Fonctionnalités avancées

Exercice 1: delegate_to

```
# mkdir /root/tp/delegate_to
# cd /root/tp/delegate_to
# cp -r ../apache_template/* .
# mv apache_template.yml apache_delegate.yml
# cat apache_delegate.yml
---
- name: Installation d'Apache sur centos1
  hosts: centos1
  tasks:
    - name: Installation du package Apache
      yum:
        name: httpd
        state: present

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

    - name: Redemarrage du service Apache
      service:
        name: httpd
        state: restarted
        enabled: true

    - name: Regle pare-feu pour http
      firewallld:
        service: http
        permanent: true
        state: enabled

    - name: Rechargement du pare-feu
      service:
        name: firewallld
        state: reloaded

- name: Création 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
        delegate_to: centos1
```

Exécution du playbook.

```
# ansible-playbook apache_delegate.yml
```

Vérifier par: <http://centos1/centos1.html>
<http://centos1/centos2.html>
<http://centos1/debian1.html>

Exercise 2: gather_facts**Etape 1**

```
# mkdir /root/tp/gather_facts
# cd /root/tp/gather_facts
# cp ../delegate_to/apache_delegate.yml gather_facts_no.yml
```

Le playbook.

```
# head -4 gather_facts_no.yml
---
- name: Installation d'Apache sur centos1
  hosts: centos1
  gather_facts: no
```

Exécution du playbook.

```
# ansible-playbook gather_facts_no.yml
```

Vérifier par: <http://centos1/centos1.html>
<http://centos1/centos2.html>
<http://centos1/debian1.html>

Etape2

Le playbook.

```
# tail -12 gather_facts_no.yml | head -4
- name: Création des pages
  hosts: all
  gather_facts: no
```

Exécution du playbook.

```
# ansible-playbook ../gather_facts_no.yml
```


Fin du support d'exercices

