

Version - **2024.01**

Dernière mise-à-jour : 2024/11/30 10:56

# LDF903 - Rôles, Gabarits, Variables et Facts

## Contenu du Module

- **LDF903 - Rôles, Gabarits, Variables et Facts**
  - Contenu du Module
  - LAB #1 - Dépendances de Rôles
  - LAB #2 - Utilisation des Gabarits
    - 2.1 - Variables
    - 2.2 - Gabarits Conditionnels
    - 2.3 - Boucles
    - 2.4 - Macros
    - 2.5 - Filtres
      - 2.5.1 - Default
      - 2.5.2 - Join
      - 2.5.3 - Map
    - 2.6 - Gabarits Parent - Enfants
      - 2.6.1 - Le Gabarit Parent
      - 2.6.2 - Le Gabarit Enfant
  - LAB #3 - Gestion de la Hiérarchie des Variables
  - LAB #4 - Utilisation des Facts d'Ansible
  - LAB #5 - La Commande ansible-vault
    - 5.1 - Crypter le Fichier
    - 5.2 - Editer le Fichier
    - 5.3 - Décrypter le Fichier
    - 5.4 - Utilisation de Mots de Passe Aléatoires
  - LAB #6 - Ansible par la Pratique

- 6.1 - Instructions
- 6.2 - Corrigés

## LAB #1 - Dépendances de Rôles

Afin de comprendre le fonctionnement des dépendances entre les Rôles vous allez étudier l'exemple de l'installation d'Apache Tomcat. Apache Tomcat est un serveur d'applications Java et par conséquent nécessite à ce que Java soit installé.

Commencez par créer le Rôle **exemple01.java** dans le répertoire **/home/trainee/.ansible/roles/** :

```
trainee@debian11:~$ mkdir /home/trainee/.ansible/roles/exemple01.java/  
trainee@debian11:~$ cd /home/trainee/.ansible/roles/exemple01.java/  
trainee@debian11:~/ansible/roles/exemple01.java$ mkdir defaults tasks templates  
trainee@debian11:~/ansible/roles/exemple01.java$
```

**Important** : Notez que dans ce Rôle nous n'avons besoin que des répertoires **defaults**, **tasks** et **templates**.

Créez le fichier **main.yaml** dans le sous-répertoire **tasks** afin d'installer Java :

```
trainee@debian11:~/ansible/roles/exemple01.java$ vi /home/trainee/.ansible/roles/exemple01.java/tasks/main.yaml  
trainee@debian11:~/ansible/roles/exemple01.java$ cat /home/trainee/.ansible/roles/exemple01.java/tasks/main.yaml  
---  
- name: install jre  
  package: name={{ java_package }} state=present  
  
- name: configure java home  
  template:  
    src: java.sh  
    dest: /etc/profile.d/java.sh
```

```
owner: root
group: root
mode: 0644
```

**Important** : Notez que le paquet à installer n'est pas explicitement déclaré. Le paquet est référencé par le contenu de la variable **java\_package**, elle-même déclarée dans le fichier **main.yaml** du sous-répertoire **defaults** du Rôle. Notez aussi l'utilisation d'un gabarit, appelé **template**, qui fournit le fichier **java.sh** qui doit être copié à l'emplacement **/etc/profile.d/** à partir du sous-répertoire **templates** du Rôle.

Créez donc le fichier **main.yaml** du sous-répertoire **defaults** :

```
trainee@debian11:~/ansible/roles/exemple01.java$ vi
/home/trainee/ansible/roles/exemple01.java/defaults/main.yaml
trainee@debian11:~/ansible/roles/exemple01.java$ cat
/home/trainee/ansible/roles/exemple01.java/defaults/main.yaml
---
java_home: /usr/lib/jvm/java-8-openjdk-amd64/jre
java_package: openjdk-8-jre
```

**Important** : Notez qu'ici sont déclarées deux variables : **java\_home** et **java\_package**.

Dernièrement, créez le fichier vide **java.sh** dans le sous-répertoire **templates** du Rôle **exemple01.java** :

```
trainee@debian11:~/ansible/roles/exemple01.java$ touch
/home/trainee/ansible/roles/exemple01.java/templates/java.sh
```

**Important** : Ce fichier ne serait pas normalement vide. Par contre dans ce LAB, nous nous concentrons sur Ansible et seule la présence du fichier est nécessaire pour le bon fonctionnement du LAB.

Créez maintenant le Rôle tomcat dans le répertoire **/home/trainee/.ansible/roles/** :

```
trainee@debian11:~/ansible/roles/exemple01.java$ mkdir /home/trainee/.ansible/roles/tomcat/  
trainee@debian11:~/ansible/roles/exemple01.java$ cd /home/trainee/.ansible/roles/tomcat/  
trainee@debian11:~/ansible/roles/tomcat$ mkdir meta tasks
```

**Important** : Notez que dans ce Rôle nous n'avons besoin que des répertoires **meta** et **tasks**.

Créez le fichier **main.yaml** dans le sous-répertoire **tasks** afin d'installer Tomcat 8 :

```
trainee@debian11:~/ansible/roles/tomcat$ vi /home/trainee/.ansible/roles/tomcat/tasks/main.yaml  
trainee@debian11:~/ansible/roles/tomcat$ cat /home/trainee/.ansible/roles/tomcat/tasks/main.yaml  
---  
- name: install tomcat  
  package: name=tomcat8 state=present
```

Créez maintenant le fichier **main.yaml** du sous-répertoire **meta** du Rôle **tomcat** :

```
trainee@debian11:~/ansible/roles/tomcat$ vi /home/trainee/.ansible/roles/tomcat/meta/main.yaml  
trainee@debian11:~/ansible/roles/tomcat$ cat /home/trainee/.ansible/roles/tomcat/meta/main.yaml  
---  
dependencies:  
  - exemple01.java
```

**Important** : Ce fichier informe Ansible que le Rôle **tomcat** dépend du Rôle **exemple01.java**.

Ensuite créez le fichier **playbook.yaml** au dessus des deux Rôles précédemment créés :

```
trainee@debian11:~/ansible/roles/tomcat$ vi /home/trainee/ansible/roles/playbook.yaml
trainee@debian11:~/ansible/roles/tomcat$ cat /home/trainee/ansible/roles/playbook.yaml
---
- hosts: all
  become: true
  roles:
    - tomcat
```

**Important** : Notez que dans le Play Book, nous appelons **uniquement** le Rôle **tomcat**.

Copiez le fichier **/home/trainee/inventory** dans le répertoire **/home/trainee/ansible/roles/** :

```
trainee@debian11:~/ansible/roles/tomcat$ cd /home/trainee/ansible/roles/
trainee@debian11:~/ansible/roles$ cp ~/inventory .
```

A l'issu de cette configuration, vous devrez obtenir l'arborescence suivante :

```
trainee@debian11:~/ansible/roles$ tree
.
├── exemple01.java
│   ├── defaults
│   │   └── main.yaml
│   └── tasks
```

```
├── main.yml
├── templates
│   └── java.sh
├── geerlingguy.java
│   ├── defaults
│   │   └── main.yml
│   ├── LICENSE
│   ├── meta
│   │   └── main.yml
│   ├── molecule
│   │   └── default
│   │       ├── converge.yml
│   │       └── molecule.yml
│   ├── README.md
│   ├── tasks
│   │   ├── main.yml
│   │   ├── setup-Debian.yml
│   │   ├── setup-FreeBSD.yml
│   │   └── setup-RedHat.yml
│   ├── templates
│   │   └── java_home.sh.j2
│   └── vars
│       ├── Debian-10.yml
│       ├── Debian-11.yml
│       ├── Debian-12.yml
│       ├── Debian-8.yml
│       ├── Debian-9.yml
│       ├── Fedora.yml
│       ├── FreeBSD.yml
│       ├── RedHat-7.yml
│       ├── RedHat-8.yml
│       ├── RedHat-9.yml
│       ├── Ubuntu-12.yml
│       └── Ubuntu-14.yml
```

```
├── Ubuntu-16.yml
├── Ubuntu-18.yml
├── Ubuntu-20.yml
├── Ubuntu-22.yml
├── inventory
├── playbook.yml
├── tomcat
│   ├── meta
│   │   └── main.yml
│   └── tasks
│       └── main.yml
```

15 directories, 34 files

Exécutez la commande **ansible-playbook** uniquement pour l'hôte **web01** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yml -l web01
```

```
PLAY [all]
```

```
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
```

```
ok: [web01]
```

```
TASK [exemple01.java : install jre]
```

```
*****
*****
```

```
ok: [web01]
```

```
TASK [exemple01.java : configure java home]
```

```
*****
```

```
*****
changed: [web01]

TASK [tomcat : install tomcat]
*****
*****
changed: [web01]

PLAY RECAP
*****
*****
web01          : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : Notez que le Rôle **exemple01.java** est traité **avant** le Rôle **tomcat**.

Vérifiez l'installation de Java et de Tomcat8 dans la machine **Web01** :

```
trainee@debian11:~/ansible/roles$ ssh web01
Debian GNU/Linux 9
Linux web01.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Sep 23 18:10:50 2023 from 10.0.2.46
trainee@web01:~$ systemctl status tomcat8
● tomcat8.service - LSB: Start Tomcat.
```

```

Loaded: loaded (/etc/init.d/tomcat8; generated; vendor preset: enabled)
Active: active (running) since Sat 2023-09-23 18:11:29 CEST; 1min 7s ago
  Docs: man:systemd-sysv-generator(8)
 CGroup: /system.slice/tomcat8.service
         └─25102 /usr/lib/jvm/default-java/bin/java -
Djava.util.logging.config.file=/var/lib/tomcat8/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.he
trainee@web01:~$ exit
déconnexion
Connection to web01 closed.
trainee@debian11:~/ansible/roles$

```

Modifiez maintenant le fichier **/home/trainee/.ansible/roles/tomcat/meta/main.yaml** :

```

trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/tomcat/meta/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/tomcat/meta/main.yaml
---
dependencies:
  - { role: exemple01.java, java_package: tree }

```

**Important** : Notez que cette fois-ci, la valeur de la variable **java\_package** spécifiée dans le fichier **/home/trainee/.ansible/roles/exemple01.java/defaults/main.yaml** est surchargée par la valeur **tree**.

Testez votre configuration :

```

trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l web02

```

```

PLAY [all]

```

```

*****
*****

```

## TASK [Gathering Facts]

```
*****
*****
```

```
ok: [web02]
```

## TASK [exemple01.java : install jre]

```
*****
*****
```

```
changed: [web02]
```

## TASK [exemple01.java : configure java home]

```
*****
*****
```

```
changed: [web02]
```

## TASK [tomcat : install tomcat]

```
*****
*****
```

```
changed: [web02]
```

## PLAY RECAP

```
*****
*****
```

```
web02                : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Vérifiez que la dépendance **tree** a été installée dans Web02 :

```
trainee@debian11:~/ansible/roles$ ssh web02
Debian GNU/Linux 9
Linux web02.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
```

```
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
Last login: Sat Sep 23 18:14:46 2023 from 10.0.2.46
```

```
trainee@web02:~$ which tree
```

```
/usr/bin/tree
```

```
trainee@web02:~$ exit
```

```
déconnexion
```

```
Connection to web02 closed.
```

```
trainee@debian11:~/ansible/roles$
```

## LAB #2 - Utilisation des Gabarits

### 2.1 - Variables

Les **Gabarits** ou *Templates* d'Ansible utilisent une bibliothèque Python qui s'appelle **Jinja2**.

**Important** : La documentation des gabarits se trouvent à cette adresse:  
[https://docs.ansible.com/ansible/latest/modules/template\\_module.html](https://docs.ansible.com/ansible/latest/modules/template_module.html).

Afin de comprendre le fonctionnement des gabarits vous allez étudier l'exemple de l'installation d'HAProxy sur la machine **Web04**.

Commencez par créer le Rôle **haproxy** :

```
trainee@debian11:~/ansible/roles$ mkdir /home/trainee/.ansible/roles/haproxy/  
trainee@debian11:~/ansible/roles$ cd /home/trainee/.ansible/roles/haproxy/  
trainee@debian11:~/ansible/roles/haproxy$ mkdir defaults handlers tasks templates
```

```
trainee@debian11:~/ansible/roles/haproxy$
```

**Important** : Notez que dans ce Rôle nous n'avons besoin que des répertoires **defaults**, **handlers**, **tasks** et **templates**.

Créez maintenant le fichier **/home/trainee/.ansible/roles/haproxy/tasks/main.yaml** :

```
trainee@debian11:~/ansible/roles/haproxy$ vi /home/trainee/.ansible/roles/haproxy/tasks/main.yaml
trainee@debian11:~/ansible/roles/haproxy$ cat /home/trainee/.ansible/roles/haproxy/tasks/main.yaml
---
- name: install
  package: name=haproxy state=present

- name: configure
  template:
    src: haproxy.cfg
    dest: /etc/haproxy/haproxy.cfg
    owner: root
    group: root
    mode: 0644
  notify: reload haproxy

- name: service
  service: name=haproxy state=started enabled=yes
```

**Important** : Notez que l'installation fait appel à un gabarit suivi par un **Handler** qui s'appelle **reload haproxy**.

Créez donc ce Handler dans le fichier **/home/trainee/.ansible/roles/haproxy/handlers/main.yaml** :

```
trainee@debian11:~/ansible/roles/haproxy$ vi /home/trainee/.ansible/roles/haproxy/handlers/main.yaml
trainee@debian11:~/ansible/roles/haproxy$ cat /home/trainee/.ansible/roles/haproxy/handlers/main.yaml
---
- name: reload haproxy
  service: name=haproxy state=reloaded
```

Créez maintenant le fichier **haproxy.cfg** dans le répertoire **/home/trainee/.ansible/roles/haproxy/templates/** :

```
trainee@debian11:~/ansible/roles/haproxy$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
trainee@debian11:~/ansible/roles/haproxy$ cat /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # Default ciphers to use on SSL-enabled listening sockets.
    # For more information, see ciphers(1SSL). This list is from:
    # https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
    # An alternative list with additional directives can be obtained from
    # https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy
    ssl-default-bind-ciphers
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
    ssl-default-bind-options no-sslv3
```

```
defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout  connect 5000
    timeout  client  50000
    timeout  server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend haproxy
    bind {{ haproxy_listen_address }}:{{haproxy_listen_port}}
    mode http
    default_backend dotcms
    stats enable
    stats uri /haproxy?stats
    stats realm HAProxy Statistics
    stats auth admin:admin
    balance roundrobin
    option httpclose
    option forwardfor

backend dotcms
    server web02 10.0.2.55:8080 check
    server web03 10.0.2.56:8080 check
```

**Important** : Notez l'utilisation de deux variables Ansible dans ce fichier - {{

**haproxy\_listen\_address }}** et **{{haproxy\_listen\_port}}**.

Spécifiez la valeur de ces variables dans le fichier **/home/trainee/.ansible/roles/haproxy/defaults/main.yaml** :

```
trainee@debian11:~/ansible/roles/haproxy$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
trainee@debian11:~/ansible/roles/haproxy$ cat /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
---
haproxy_listen_address: 0.0.0.0
haproxy_listen_port: 80
haproxy_log: haproxy.log
```

Dernièrement, modifiez le fichier **~/ansible/roles/playbook.yaml** afin d'appeler le Rôle **haproxy** :

```
trainee@debian11:~/ansible/roles/haproxy$ cd ..
trainee@debian11:~/ansible/roles$ vi playbook.yaml
trainee@debian11:~/ansible/roles$ cat playbook.yaml
---
- hosts: all
  become: true
  roles:
    - haproxy
```

Exécutez la commande **ansible-playbook** sur le groupe **equilibrage** qui contient **web04** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l equilibrage

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

TASK [Gathering Facts]
*****
```

```
*****
ok: [web04]

TASK [haproxy : install]
*****
*****
ok: [web04]

TASK [haproxy : configure]
*****
*****
changed: [web04]

TASK [haproxy : service]
*****
*****
ok: [web04]

RUNNING HANDLER [haproxy : reload haproxy]
*****
*****
changed: [web04]

PLAY RECAP
*****
*****
web04          : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Consultez l'état du service **haproxy** ainsi que le contenu du fichier **/etc/haproxy/haproxy.cfg** dans la machine **Web04** :

```
trainee@debian11:~/ansible/roles$ ssh web04
Debian GNU/Linux 9
Linux web04.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Sat Sep 23 18:24:17 2023 from 10.0.2.46

```
trainee@web04:~$ systemctl status haproxy
```

```
● haproxy.service - HAProxy Load Balancer
```

```
Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since Wed 2023-09-20 15:39:42 CEST; 3 days ago
```

```
Docs: man:haproxy(1)
```

```
file:/usr/share/doc/haproxy/configuration.txt.gz
```

```
Process: 16494 ExecReload=/bin/kill -USR2 $MAINPID (code=exited, status=0/SUCCESS)
```

```
Process: 16492 ExecReload=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRA_OPTS (code=exited, status=0/SUCCESS)
```

```
Main PID: 2970 (haproxy-systemd)
```

```
Tasks: 3 (limit: 4915)
```

```
CGroup: /system.slice/haproxy.service
```

```
├─ 2970 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
```

```
├─ 16500 /usr/sbin/haproxy-master
```

```
└─ 16501 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds -sf 2975
```

```
trainee@web04:~$ cat /etc/haproxy/haproxy.cfg
```

```
global
```

```
log /dev/log local0
```

```
log /dev/log local1 notice
```

```
chroot /var/lib/haproxy
```

```
stats socket /run/haproxy/admin.sock mode 660 level admin
```

```
stats timeout 30s
```

```
user haproxy
```

```
group haproxy
```

```
daemon
```

```
# Default SSL material locations
ca-base /etc/ssl/certs
crt-base /etc/ssl/private

# Default ciphers to use on SSL-enabled listening sockets.
# For more information, see ciphers(1SSL). This list is from:
# https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
# An alternative list with additional directives can be obtained from
# https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy
ssl-default-bind-ciphers
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
ssl-default-bind-options no-sslv3

defaults
    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout  connect 5000
    timeout  client  50000
    timeout  server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend haproxy
    bind 0.0.0.0:80
    mode http
    default_backend dotcms
    stats enable
```

```
stats uri /haproxy?stats
stats realm HAProxy Statistics
stats auth admin:admin
balance roundrobin
option httpclose
option forwardfor

backend dotcms
  server web02 10.0.2.55:8080 check
  server web03 10.0.2.56:8080 check

trainee@web04:~$ exit
déconnexion
Connection to web04 closed.
trainee@debian11:~/ansible/roles$
```

**Important** : Notez que les valeurs des variables spécifiées dans le fichier **/home/trainee/.ansible/roles/haproxy/defaults/main.yaml** ont été injectées à la place des variables **{{ haproxy\_listen\_address }}** et **{{ haproxy\_listen\_port }}**.

## 2.2 - Gabarits Conditionnels

Les gabarits peuvent être configurés d'une manière conditionnelle afin de produire des résultats différents en fonction de la valeur d'une variable. Afin de comprendre le fonctionnement des gabarits conditionnels vous allez modifier l'exemple de l'installation d'HAProxy sur la machine **Web04**.

Editez le fichier **/home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
trainee@debian11:~/ansible/roles$ tail -n 17 /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
frontend haproxy
```

```
bind {{ haproxy_listen_address }}:{{haproxy_listen_port}}
mode http
default_backend dotcms
{% if haproxy_stats %}
  stats enable
  stats uri /haproxy?stats
  stats realm HAProxy Statistics
  stats auth admin:admin
{% endif %}
balance roundrobin
option httpclose
option forwardfor

backend dotcms
  server web02 10.0.2.55:8080 check
  server web03 10.0.2.56:8080 check
```

**Important** : Notez la condition **{% if haproxy\_stats %}** qui ne tiendra compte des quatre lignes jusqu'à la ligne **{% endif %}** que dans le cas où la valeur de la variable **haproxy\_stats** est **True**.

Définissez la variable **haproxy\_stats** dans le fichier **/home/trainee/.ansible/roles/haproxy/defaults/main.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
---
haproxy_listen_address: 0.0.0.0
haproxy_listen_port: 80
haproxy_log: haproxy.log
haproxy_stats: True
```

**Important** : Notez qu'Ansible teste si la variable est définie. Par conséquent la variable peut contenir la valeur **True**, **true** voire toute autre chaîne telle que **toto**.

Exécutez maintenant la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l equilibrage

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

TASK [Gathering Facts]
*****
*****
ok: [web04]

TASK [haproxy : install]
*****
*****
ok: [web04]

TASK [haproxy : configure]
*****
*****
ok: [web04]

TASK [haproxy : service]
*****
*****
ok: [web04]
```

## PLAY RECAP

```
*****
*****
web04          : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Contrôlez le contenu du fichier **/etc/haproxy/haproxy.cfg** dans la machine Web04 :

```
trainee@debian11:~/ansible/roles$ ssh web04
Debian GNU/Linux 9
Linux web04.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar  9 15:29:27 2022 from 10.0.2.45

trainee@web04:~$ tail -n 17 /etc/haproxy/haproxy.cfg
    errorfile 504 /etc/haproxy/errors/504.http

frontend haproxy
    bind 0.0.0.0:80
    mode http
    default_backend dotcms
    stats enable
    stats uri /haproxy?stats
    stats realm HAProxy Statistics
    stats auth admin:admin
    balance roundrobin
    option httpclose
    option forwardfor
```

```
backend dotcms
  server web02 10.0.2.55:8080 check
  server web03 10.0.2.56:8080 check

trainee@web04:~$ exit
déconnexion
Connection to web04 closed.
trainee@debian11:~/ansible/roles$
```

**Important** : Notez que les quatre lignes concernant les statistiques ont été incluses dans le fichier.

Éditez de nouveau le fichier **/home/trainee/.ansible/roles/haproxy/defaults/main.yaml** en modifiant la valeur de la variable **haproxy\_stats** de True à **False** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
---
haproxy_listen_address: 0.0.0.0
haproxy_listen_port: 80
haproxy_log: haproxy.log
haproxy_stats: False
```

**Important** : Notez qu'Ansible teste si la variable n'est pas définie ou définie avec la valeur de **False** ou **false**. La valeur donc **haproxy\_stats**: seule est considérée comme étant fausse.

Exécutez de nouveau la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l equilibrage

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

TASK [Gathering Facts]
*****
*****
ok: [web04]

TASK [haproxy : install]
*****
*****
ok: [web04]

TASK [haproxy : configure]
*****
*****
changed: [web04]

TASK [haproxy : service]
*****
*****
ok: [web04]

RUNNING HANDLER [haproxy : reload haproxy]
*****
*****
changed: [web04]

PLAY RECAP
*****
*****
```

```
web04           : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Contrôlez le contenu du fichier **/etc/haproxy/haproxy.cfg** dans la machine Web04 :

```
trainee@debian11:~/ansible/roles$ ssh web04
Debian GNU/Linux 9
Linux web04.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Sep 23 18:31:28 2023 from 10.0.2.46

trainee@web04:~$ tail -n 17 /etc/haproxy/haproxy.cfg
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend haproxy
    bind 0.0.0.0:80
    mode http
    default_backend dotcms
    balance roundrobin
    option httpclose
    option forwardfor

backend dotcms
    server web02 10.0.2.55:8080 check
```

```
server web03 10.0.2.56:8080 check
```

```
trainee@web04:~$ exit
déconnexion
Connection to web04 closed.
trainee@debian11:~/ansible/roles$
```

**Important** : Notez que les quatre lignes concernant les statistiques n'ont pas été incluses dans le fichier.

L'inverse de la condition **{% if haproxy\_stats %}** peut être obtenue en utilisant le mot **not** :

```
{% if not haproxy_stats %}
```

La condition suivante est remplie quelque soit la valeur de la variable **{% if haproxy\_stats %}** :

```
{% if haproxy_stats is defined %}
```

La condition suivante est identique à la condition **{% if haproxy\_stats %}** :

```
{% if haproxy_stats is defined and haproxy_stats %}
```

## 2.3 - Boucles

Créez la variable **haproxy\_backends** dans le fichier **/home/trainee/.ansible/roles/haproxy/defaults/main.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
---
haproxy_listen_address: 0.0.0.0
```

```
haproxy_listen_port: 80
haproxy_log: haproxy.log
haproxy_stats: True
haproxy_backends:
  - 'server web02 10.0.2.55:8080 check'
  - 'server web03 10.0.2.56:8080 check'
```

**Important** : Notez que la variable **haproxy\_backends** est une liste YAML.

Créez ensuite un boucle dans le fichier **/home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
trainee@debian11:~/ansible/roles$ tail /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
    stats auth admin:admin
{% endif %}
    balance roundrobin
    option httpclose
    option forwardfor

backend dotcms
{% for backend in haproxy_backends %}
    {{ backend }}
{% endfor %}
```

Une autre façon de créer un boucle est d'utiliser un dictionnaire YAML :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
---
haproxy_listen_address: 0.0.0.0
haproxy_listen_port: 80
```

```
haproxy_log: haproxy.log
haproxy_stats: True
haproxy_backends:
  web02:
    ip: 10.0.2.55
  web03:
    ip: 10.0.2.56
```

Pour accéder aux données dans le dictionnaire, il convient d'utiliser la fonction Python **items** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/ansible/roles/haproxy/templates/haproxy.cfg
trainee@debian11:~/ansible/roles$ tail /home/trainee/ansible/roles/haproxy/templates/haproxy.cfg
  stats auth admin:admin
{% endif %}
  balance roundrobin
  option httpclose
  option forwardfor

backend dotcms
{% for key, value in haproxy_backends.items() %}
  server {{ key }} {{ value.ip }}:8080 check
{% endfor %}
```

## 2.4 - Macros

Il est aussi possible d'utiliser un **macro** avec le dictionnaire :

```
trainee@debian11:~/ansible/roles$ vi backend.j2
trainee@debian11:~/ansible/roles$ cat backend.j2
{% macro backend(name, ip, port=8080) -%}
  server {{ name }} {{ ip }}:{{ port }} check
{%- endmacro %}
```

Il convient ensuite d'importer les valeurs dans le fichier **/home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
trainee@debian11:~/ansible/roles$ tail /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
{% endif %}
    balance roundrobin
    option httpclose
    option forwardfor

{% import 'backend.j2' as backend %}
backend dotcms
{% for key, value in haproxy_backends.items() %}
{{backend.backend(key, value.ip)}}
{% endfor %}
```

## 2.5 - Filtres

### 2.5.1 - Default

Le filtre **default** permet de fournir une valeur par défaut pour une variable. Éditez le fichier **/home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg** et modifiez la variable **haproxy\_listen\_port**, la condition **haproxy\_stats** et la section backend dotcms :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
trainee@debian11:~/ansible/roles$ tail -n 17 /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
    bind {{ haproxy_listen_address }}:{{haproxy_listen_port|default('80') }}
    mode http
    default_backend dotcms
{% if haproxy_stats|default(True) %}
    stats enable
    stats uri /haproxy?stats
    stats realm HAProxy Statistics
```

```
    stats auth admin:admin
{% endif %}
    balance roundrobin
    option httpclose
    option forwardfor

backend dotcms
{% for key, value in haproxy_backends.items() %}
    server {{ key }} {{ value.ip }}:8080 check
{% endfor %}
```

Supprimez ensuite les lignes **haproxy\_listen\_port** et **haproxy\_stats** du fichier **/home/trainee/.ansible/roles/haproxy/defaults/main.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
---
haproxy_listen_address: 0.0.0.0
haproxy_log: haproxy.log
haproxy_backends:
  web02:
    ip: 10.0.2.55
  web03:
    ip: 10.0.2.56
```

Testez la configuration avec la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l equilibrage

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

TASK [Gathering Facts]
*****
```

```
*****
ok: [web04]

TASK [haproxy : install]
*****
*****
ok: [web04]

TASK [haproxy : configure]
*****
*****
changed: [web04]

TASK [haproxy : service]
*****
*****
ok: [web04]

RUNNING HANDLER [haproxy : reload haproxy]
*****
*****
changed: [web04]

PLAY RECAP
*****
*****
web04          : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Vérifiez ensuite que les valeurs par défaut ont bien été utilisées :

```
trainee@debian11:~/ansible/roles$ ssh web04
Debian GNU/Linux 9
Linux web04.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
Last login: Wed Mar  9 15:48:45 2022 from 10.0.2.45  
trainee@web04:~$ tail -n 17 /etc/haproxy/haproxy.cfg  
    errorfile 504 /etc/haproxy/errors/504.http
```

```
frontend haproxy  
    bind 0.0.0.0:80  
    mode http  
    default_backend dotcms  
    stats enable  
    stats uri /haproxy?stats  
    stats realm HAProxy Statistics  
    stats auth admin:admin  
    balance roundrobin  
    option httpclose  
    option forwardfor
```

```
backend dotcms  
    server web02 10.0.2.55:8080 check  
    server web03 10.0.2.56:8080 check  
trainee@web04:~$ exit  
déconnexion  
Connection to web04 closed.  
trainee@debian11:~/ansible/roles$
```

## 2.5.2 - Join

Le filtre **Join** :

```
{{ haproxy_backends|join(',') }}
```

permet de prendre une liste YAML :

```
haproxy_backends:  
  - web02  
  - web03
```

et de produire une liste séparée par des virgules :

```
web02,web03
```

### 2.5.3 - Map

Le filtre **Map** :

```
{{ haproxy_backends.values()|map(attribute='ip')|join(',') }}
```

permet de prendre un dictionnaire YAML :

```
haproxy_backends:  
  web02:  
    ip: 10.0.2.55  
  web03:  
    ip: 10.0.2.56
```

et de produire une liste séparée par des virgules :

```
10.0.2.55,10.0.2.56
```

## 2.6 - Gabarits Parent - Enfants

HAProxy ne gère pas uniquement le protocole http. Il peut également gérer d'autres connexions TCP. Pour cette raison, il est intéressant de créer un gabarit générique pour HAProxy et un gabarit enfants par protocole.

### 2.6.1 - Le Gabarit Parent

Modifiez donc le fichier **/home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

{% block globals %}
{% endblock %}

defaults
    log      global
    option dontlognull
    timeout connect 5000
    timeout client  50000
    timeout server  50000

{% block defaults %}
{% endblock %}
```

```
{% block server %}  
{% endblock %}
```

**Important** : Ce gabarit ne contient que des directives générales. Les directives spécifiques au protocole http ont été remplacées par des **blocs** nommés **globals**, **defaults** et **server**.

### 2.6.2 - Le Gabarit Enfant

Créez maintenant le gabarit `/home/trainee/.ansible/roles/haproxy/templates/haproxy.http.cfg`, spécifique au protocole http :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.http.cfg  
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/templates/haproxy.http.cfg  
{% extends 'haproxy.cfg' %}  
{% block globals %}  
    ca-base /etc/ssl/certs  
    crt-base /etc/ssl/private  
    ssl-default-bind-ciphers  
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS  
    ssl-default-bind-options no-ssl3  
{% endblock %}  
{% block defaults %}  
    mode http  
    option httplog  
    errorfile 400 /etc/haproxy/errors/400.http  
    errorfile 403 /etc/haproxy/errors/403.http  
    errorfile 408 /etc/haproxy/errors/408.http  
    errorfile 500 /etc/haproxy/errors/500.http  
    errorfile 502 /etc/haproxy/errors/502.http  
    errorfile 503 /etc/haproxy/errors/503.http
```

```
    errorfile 504 /etc/haproxy/errors/504.http
{% endblock %}
{% block server %}
frontend haproxy
    bind {{ haproxy_listen_address }}:{{haproxy_listen_port|default('80') }}
    mode http
    default_backend dotcms
{% if haproxy_stats|default(True) %}
    stats enable
    stats uri /haproxy?stats
    stats realm HAProxy Statistics
    stats auth admin:admin
{% endif %}
    balance roundrobin
    option httpclose
    option forwardfor

backend dotcms
{% for key, value in haproxy_backends.items() %}
    server {{ key }} {{ value.ip }}:8080 check
{% endfor %}
{% endblock %}
```

**Important** : Notez que les **blocs** nommés **globals**, **defaults** et **server** contiennent les directives qui seront injectées dans le fichier **haproxy.cfg** aux emplacements des trois blocs respectifs.

Modifiez ensuite le fichier **/home/trainee/.ansible/roles/haproxy/tasks/main.yaml** afin d'appeler le gabarit **haproxy.http.cfg** :

```
trainee@debian11:~/.ansible/roles$ vi /home/trainee/.ansible/roles/haproxy/tasks/main.yaml
trainee@debian11:~/.ansible/roles$ cat /home/trainee/.ansible/roles/haproxy/tasks/main.yaml
```

```
---
- name: install
  package: name=haproxy state=present

- name: configure
  template:
    src: haproxy.http.cfg
    dest: /etc/haproxy/haproxy.cfg
    owner: root
    group: root
    mode: 0644
  notify: reload haproxy

- name: service
  service: name=haproxy state=started enabled=yes
```

Exécutez la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l equilibrage
```

```
PLAY [all]
```

```
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
```

```
ok: [web04]
```

```
TASK [haproxy : install]
```

```
*****
*****
```

```
ok: [web04]
```

## TASK [haproxy : configure]

```
*****  
*****  
changed: [web04]
```

## TASK [haproxy : service]

```
*****  
*****  
ok: [web04]
```

## RUNNING HANDLER [haproxy : reload haproxy]

```
*****  
*****  
changed: [web04]
```

## PLAY RECAP

```
*****  
*****  
web04          : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0  
ignored=0
```

Contrôlez maintenant le contenu du fichier `/etc/haproxy/haproxy.cfg` de la machine **Web04** :

```
trainee@debian11:~/ansible/roles$ ssh web04  
Debian GNU/Linux 9  
Linux web04.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
Last login: Wed Mar  9 15:54:56 2022 from 10.0.2.45
```

```
trainee@web04:~$ cat /etc/haproxy/haproxy.cfg
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private
    ssl-default-bind-ciphers
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
    ssl-default-bind-options no-sslv3

defaults
    log          global
    option dontlognull
    timeout connect 5000
    timeout client  50000
    timeout server  50000
    mode          http
    option httplog
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend haproxy
    bind 0.0.0.0:80
```

```
mode http
default_backend dotcms
stats enable
stats uri /haproxy?stats
stats realm HAProxy Statistics
stats auth admin:admin
balance roundrobin
option httpclose
option forwardfor

backend dotcms
server web02 10.0.2.55:8080 check
server web03 10.0.2.56:8080 check
trainee@web04:~$ exit
déconnexion
Connection to web04 closed.
trainee@debian11:~/ansible/roles$
```

**Important** : Notez que les **blocs** nommés **globals**, **defaults** et **server** ont été renseignés.

## LAB #3 - Gestion de la Hiérarchie des Variables

La hiérarchie de la prise en compte des variables par Ansible peut être illustrée par le diagramme suivant :

```
.
├── 1.commandline
├── roles
│   └── 4.playbook.yaml
```

```
├── debug
│   ├── defaults
│   │   └── 6.main.yaml
│   └── tasks
│       └── 7.main.yaml
├── group_vars
│   ├── 3.groupname.yaml
│   └── 5.all.yaml
├── host_vars
│   └── 2.hostname.yaml
```

Pour illustrer cette hiérarchie, créez le Rôle **/home/trainee/.ansible/roles/debug** contenant les sous-répertoires **defaults** et **tasks** :

```
trainee@debian11:~/.ansible/roles$ cd ../../
trainee@debian11:~$ mkdir /home/trainee/.ansible/roles/debug
trainee@debian11:~$ mkdir /home/trainee/.ansible/roles/debug/defaults
trainee@debian11:~$ mkdir /home/trainee/.ansible/roles/debug/tasks
trainee@debian11:~$
```

Créez les fichiers **main.yaml** dans les sous-répertoires **defaults** et **tasks** du Rôle **/home/trainee/.ansible/roles/debug** :

```
trainee@debian11:~$ touch /home/trainee/.ansible/roles/debug/defaults/main.yaml
trainee@debian11:~$ touch /home/trainee/.ansible/roles/debug/tasks/main.yaml
trainee@debian11:~$
```

Vous obtiendrez :

```
trainee@debian11:~$ cd .ansible/roles/
trainee@debian11:~/.ansible/roles$ tree debug
debug
├── defaults
│   └── main.yaml
└── tasks
    └── main.yaml
```

2 directories, 2 files

Créez le répertoire **/home/trainee/.ansible/roles/group\_vars** ainsi que le fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** :

```
trainee@debian11:~/ansible/roles$ mkdir /home/trainee/.ansible/roles/group_vars
trainee@debian11:~/ansible/roles$ touch /home/trainee/.ansible/roles/group_vars/all.yaml
trainee@debian11:~/ansible/roles$
```

Vous obtiendrez :

```
trainee@debian11:~/ansible/roles$ tree group_vars/
group_vars/
├── all.yaml

0 directories, 1 file
```

Éditez le fichier **/home/trainee/.ansible/roles/debug/tasks/main.yaml** en déclarant la valeur par défaut de la variable **endroit** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/debug/tasks/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/debug/tasks/main.yaml
---
- debug:
  msg: "Ce message est issu de {{ endroit|default('roles/debug/tasks/main.yaml') }}"
```

Éditez le fichier **/home/trainee/.ansible/roles/playbook.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/playbook.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/playbook.yaml
---
- hosts: all
  roles:
  - debug
```

Exécutez la commande **ansible-playbook** pour voir la valeur par défaut de la variable **endroit** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml
```

```
PLAY [all]
```

```
*****  
*****
```

```
TASK [Gathering Facts]
```

```
*****  
*****
```

```
ok: [web04]
```

```
ok: [web03]
```

```
ok: [web02]
```

```
ok: [web01]
```

```
TASK [debug : debug]
```

```
*****  
*****
```

```
ok: [web01] => {  
  "msg": "Ce message est issu de roles/debug/tasks/main.yaml"  
}
```

```
ok: [web02] => {  
  "msg": "Ce message est issu de roles/debug/tasks/main.yaml"  
}
```

```
ok: [web03] => {  
  "msg": "Ce message est issu de roles/debug/tasks/main.yaml"  
}
```

```
ok: [web04] => {  
  "msg": "Ce message est issu de roles/debug/tasks/main.yaml"  
}
```

```
PLAY RECAP
```

```
*****
```

```
*****
web01      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Définissez maintenant la valeur de la variable **endroit** dans le fichier **/home/trainee/.ansible/roles/debug/defaults/main.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/debug/defaults/main.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/debug/defaults/main.yaml
---
endroit: 'roles/debug/defaults/main.yaml'
```

Exécutez la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml

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

TASK [Gathering Facts]
*****
*****

ok: [web03]
ok: [web04]
ok: [web02]
ok: [web01]

TASK [debug : debug]
```

```

*****
*****
ok: [web01] => {
  "msg": "Ce message est issu de roles/debug/defaults/main.yaml"
}
ok: [web02] => {
  "msg": "Ce message est issu de roles/debug/defaults/main.yaml"
}
ok: [web03] => {
  "msg": "Ce message est issu de roles/debug/defaults/main.yaml"
}
ok: [web04] => {
  "msg": "Ce message est issu de roles/debug/defaults/main.yaml"
}

PLAY RECAP
*****
*****
web01                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

```

**Important** : La variable fixée dans **defaults/main.yaml** surcharge la variable fixée dans **tasks/main.yaml**.

Définissez la valeur de la variable **endroit** dans le fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/group_vars/all.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/group_vars/all.yaml
---
endroit: 'roles/group_vars/all.yaml'
```

**Important** : La déclaration de la variable peut être faite dans **roles/group\_vars/all** ou dans **roles/group\_vars/all.yaml** ou dans un fichier \*.yaml dans le répertoire **roles/group\_vars/all/**.

Exécutez la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml

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

TASK [Gathering Facts]
*****
*****

ok: [web04]
ok: [web03]
ok: [web01]
ok: [web02]

TASK [debug : debug]
*****
*****

ok: [web01] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
```

```
ok: [web02] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
ok: [web03] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
ok: [web04] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
```

#### PLAY RECAP

```
*****
*****
web01                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : La variable fixée dans **group\_vars/all.yaml** surcharge la variable fixée dans **defaults/main.yaml** qui surcharge la variable fixée dans **tasks/main.yaml**.

Définissez la valeur de la variable **endroit** dans le fichier **/home/trainee/.ansible/roles/playbook.yaml** :

```
trainee@debian11:~/ansible/roles$ vi playbook.yaml
trainee@debian11:~/ansible/roles$ cat playbook.yaml
---
- hosts: all
```

```
roles:
  - { role: debug, endroit: 'playbook.yaml' }
```

Exécutez la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml
```

```
PLAY [all]
```

```
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
```

```
ok: [web02]
```

```
ok: [web03]
```

```
ok: [web01]
```

```
ok: [web04]
```

```
TASK [debug : debug]
```

```
*****
*****
```

```
ok: [web01] => {
  "msg": "Ce message est issu de playbook.yaml"
}
```

```
ok: [web02] => {
  "msg": "Ce message est issu de playbook.yaml"
}
```

```
ok: [web03] => {
  "msg": "Ce message est issu de playbook.yaml"
}
```

```
ok: [web04] => {
  "msg": "Ce message est issu de playbook.yaml"
}
```

## PLAY RECAP

```
*****
*****
web01                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : La variable fixée dans **playbook.yaml** surcharge la variable fixée dans **group\_vars/all.yaml** qui surcharge la variable fixée dans **defaults/main.yaml** qui surcharge la variable fixée dans **tasks/main.yaml**.

Créez maintenant le fichier **/home/trainee/.ansible/roles/group\_vars/group1.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/group_vars/group1.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/group_vars/group1.yaml
---
endroit: 'group_vars/group1.yaml'
```

Créez ensuite le fichier **/home/trainee/.ansible/roles/host\_vars/localhost.yaml** :

```
trainee@debian11:~/ansible/roles$ mkdir /home/trainee/.ansible/roles/host_vars
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/host_vars/localhost.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/host_vars/localhost.yaml
---
# endroit: 'host_vars/localhost.yaml'
```

Modifiez le fichier **/home/trainee/.ansible/roles/playbook.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/playbook.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/playbook.yaml
---
- hosts: all
  roles:
    - debug
```

Modifiez ensuite le fichier **inventory** en ajoutant **localhost** :

```
trainee@debian11:~/ansible/roles$ vi inventory
trainee@debian11:~/ansible/roles$ cat inventory
localhost ansible_connection=local

[basededonnees]
web01

[dotcms]
web02
web03

[equilibrage]
web04

[debian:children]
basededonnees
dotcms
equilibrage

[debian:vars]
ansible_user=trainee
```

Vérifiez que les variables sont lues à partir du fichier **all.yaml** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml
```

```
PLAY [all]
```

```
*****  
*****
```

```
TASK [Gathering Facts]
```

```
*****  
*****
```

```
ok: [web04]
```

```
ok: [web01]
```

```
ok: [web02]
```

```
ok: [web03]
```

```
ok: [localhost]
```

```
TASK [debug : debug]
```

```
*****  
*****
```

```
ok: [localhost] => {  
  "msg": "Ce message est issu de roles/group_vars/all.yaml"}
```

```
ok: [web01] => {  
  "msg": "Ce message est issu de roles/group_vars/all.yaml"}
```

```
ok: [web02] => {  
  "msg": "Ce message est issu de roles/group_vars/all.yaml"}
```

```
ok: [web03] => {  
  "msg": "Ce message est issu de roles/group_vars/all.yaml"}
```

```
ok: [web04] => {  
  "msg": "Ce message est issu de roles/group_vars/all.yaml"}
```

## PLAY RECAP

```
*****
*****
localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web01                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : Notez que la valeur de la variable **endroit** spécifiée dans le fichier **group\_vars/all.yaml** s'applique à tous les groupes **et** à tous les hôtes.

Modifiez ensuite le fichier **inventory** en mettant localhost dans le **group1** :

```
trainee@debian11:~/ansible/roles$ vi inventory
trainee@debian11:~/ansible/roles$ cat inventory
[group1]
localhost ansible_connection=local

[basededonnees]
web01

[dotcms]
web02
web03
```

```
[equilibrage]
web04

[debian:children]
basededonnees
dotcms
equilibrage

[debian:vars]
ansible_user=trainee
```

Exécutez de nouveau **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml

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

TASK [Gathering Facts]
*****
*****

ok: [web03]
ok: [web01]
ok: [web02]
ok: [web04]
ok: [localhost]

TASK [debug : debug]
*****
*****

ok: [localhost] => {
  "msg": "Ce message est issu de group_vars/group1.yaml"
}
```

```
ok: [web01] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
ok: [web02] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
ok: [web03] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
ok: [web04] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
```

#### PLAY RECAP

```
*****
*****
localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web01                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : La variable fixée dans **group\_vars/group1.yaml** surcharge la variable fixée dans **group\_vars/all.yaml**.

Modifiez le fichier **/home/trainee/.ansible/roles/host\_vars/localhost.yaml** en décommentant la ligne 2 :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/host_vars/localhost.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/host_vars/localhost.yaml
---
endroit: 'host_vars/localhost.yaml'
```

Exécutez la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml

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

TASK [Gathering Facts]
*****
*****

ok: [web02]
ok: [web03]
ok: [web01]
ok: [web04]
ok: [localhost]

TASK [debug : debug]
*****
*****

ok: [localhost] => {
  "msg": "Ce message est issu de host_vars/localhost.yaml"
}
ok: [web01] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
ok: [web02] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
```

```
ok: [web03] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
ok: [web04] => {
  "msg": "Ce message est issu de roles/group_vars/all.yaml"
}
```

#### PLAY RECAP

```
*****
*****
localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web01                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04                    : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : La variable fixée dans **host\_vars/localhost.yaml** surcharge la variable fixée dans **group\_vars/group1.yaml** qui surcharge la variable fixée dans **group\_vars/all.yaml**.

Exécutez la commande **ansible-playbook** en définissant la valeur de la variable **endroit** sur la ligne de commande :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -e 'endroit="la ligne de commande"'
```

```
PLAY [all]
```

```
*****
*****
```

TASK [Gathering Facts]

```
*****
*****
```

```
ok: [web04]
ok: [web03]
ok: [web02]
ok: [web01]
ok: [localhost]
```

TASK [debug : debug]

```
*****
*****
```

```
ok: [localhost] => {
  "msg": "Ce message est issu de la ligne de commande"
}
ok: [web01] => {
  "msg": "Ce message est issu de la ligne de commande"
}
ok: [web02] => {
  "msg": "Ce message est issu de la ligne de commande"
}
ok: [web03] => {
  "msg": "Ce message est issu de la ligne de commande"
}
ok: [web04] => {
  "msg": "Ce message est issu de la ligne de commande"
}
```

PLAY RECAP

```
*****
*****
```

```
localhost      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web01          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : La variable fixée dans sur la **ligne de commande** surcharge toutes les autres variables.

## LAB #4 - Utilisation des Facts d'Ansible

Ansible Facts sont :

- des variables collectées automatiquement par le module **setup** d'Ansible lors du contrôle à distance d'un hôte,
- spécifiques à l'hôte concerné.

Il est cependant possible d'exécuter ce module manuellement afin de voir les Facts collectées :

```
trainee@debian11:~/ansible/roles$ ansible all -i web01, -m setup | more
web01 | SUCCESS => {
  "ansible_facts": {
    "ansible_all_ipv4_addresses": [
      "10.0.2.54"
    ],
```

```
"ansible_all_ipv6_addresses": [
  "fe80::d255:17e1:74c1:306e"
],
"ansible_apparmor": {
  "status": "disabled"
},
"ansible_architecture": "x86_64",
"ansible_bios_date": "04/01/2014",
"ansible_bios_vendor": "SeaBIOS",
"ansible_bios_version": "rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org",
"ansible_board_asset_tag": "NA",
"ansible_board_name": "NA",
"ansible_board_serial": "NA",
"ansible_board_vendor": "NA",
"ansible_board_version": "NA",
"ansible_chassis_asset_tag": "NA",
"ansible_chassis_serial": "NA",
"ansible_chassis_vendor": "QEMU",
"ansible_chassis_version": "pc-i440fx-5.2",
"ansible_cmdline": {
  "BOOT_IMAGE": "/boot/vmlinuz-4.9.0-8-amd64",
  "quiet": true,
  "ro": true,
  "root": "UUID=b29b93ed-8de4-4980-a407-15bba3ad09ba"
},
"ansible_date_time": {
  "date": "2022-03-09",
  "day": "09",
  "epoch": "1646840799",
  "hour": "16",
  "iso8601": "2022-03-09T15:46:39Z",
  "iso8601_basic": "20220309T164639891872",
  "iso8601_basic_short": "20220309T164639",
  "iso8601_micro": "2022-03-09T15:46:39.891872Z",
```

```
"minute": "46",
"month": "03",
"second": "39",
"time": "16:46:39",
"tz": "CET",
"tz_dst": "CEST",
"tz_offset": "+0100",
"weekday": "mercredi",
"weekday_number": "3",
"weeknumber": "10",
"year": "2022"
},
"ansible_default_ipv4": {
  "address": "10.0.2.54",
  "alias": "ens18",
  "broadcast": "10.0.2.255",
  "gateway": "10.0.2.1",
```

--More--

**Important** : Notez que le nom de chaque **Fact** commence par **ansible\_**. Évitez donc de créer des variables commençant par cette valeur !

Parmi les Facts, les plus utiles sont :

- "ansible\_hostname": "web01",
- "ansible\_fqdn": "web01.i2tch.loc",
- "ansible\_architecture": "x86\_64",
- "ansible\_distribution": "Debian",
- "ansible\_distribution\_major\_version": "9",
- "ansible\_distribution\_version": "9.7",
- "ansible\_os\_family": "Debian",

- "ansible\_memtotal\_mb": 240,
- "ansible\_processor\_cores": 1,
- "ansible\_virtualization\_role": "guest".

Ainsi que dictionnaire **ansible\_default\_ipv4** :

```
"ansible_default_ipv4": {
  "address": "10.0.2.54",
  "alias": "ens18",
  "broadcast": "10.0.2.255",
  "gateway": "10.0.2.1",
  "interface": "ens18",
  "macaddress": "de:b9:d4:17:cd:da",
  "mtu": 1500,
  "netmask": "255.255.255.0",
  "network": "10.0.2.0",
  "type": "ether"
},
```

Modifiez le fichier **inventory** en mettant les deux machines TargetA et TargetB dans le groupe **linux** :

```
trainee@debian11:~/.ansible/roles$ vi inventory
trainee@debian11:~/.ansible/roles$ cat inventory
[linux]
targeta
targetb

[group1]
localhost ansible_connection=local

[basededonnees]
web01

[dotcms]
```

```
web02
web03

[equilibrage]
web04

[debian:children]
basededonnees
dotcms
equilibrage

[debian:vars]
ansible_user=trainee
```

Modifiez le fichier **playbook.yaml** afin d'utiliser le Rôle **exemple01.java** en tant qu'utilisateur privilégié :

```
trainee@debian11:~/ansible/roles$ vi playbook.yaml
trainee@debian11:~/ansible/roles$ cat playbook.yaml
---
- hosts: all
  become: true
  roles:
    - exemple01.java
```

Utilisez la commande `scp` pour copier le fichier `authorized_keys` de la machine virtuelle `debian11` vers la machine virtuelle `targetb` :

```
trainee@debian11:~/ansible/roles$ cd ~
trainee@debian11:~$ scp .ssh/authorized_keys trainee@10.0.2.53:/home/trainee/.ssh/authorized_keys
trainee@10.0.2.53's password: trainee
authorized_keys
100% 888    8.9KB/s   00:00
trainee@debian11:~$ cd -
/home/trainee/.ansible/roles
```

```
trainee@debian11:~/ansible/roles$
```

Grâce au Fact **ansible\_os\_family** il est possible d'appliquer le fichier **/home/trainee/ansible/roles/exemple01.java/tasks/main.yaml** aux deux distributions. Notez l'utilisation de la clause **when** :

```
trainee@debian11:~/ansible/roles$ vi exemple01.java/tasks/main.yaml
trainee@debian11:~/ansible/roles$ cat exemple01.java/tasks/main.yaml
---
- name: install jre (Debian)
  package: name=openjdk-8-jre state=present
  when: ansible_os_family == 'Debian'

- name: install jre (CentOS)
  package: name=java-1.8.0-openjdk state=present
  when: ansible_os_family == 'RedHat'
```

Modifiez le fichier **/etc/hosts** de la machine **debian11** :

```
trainee@debian11:~/ansible/roles$ su -
Password: fenestros

root@debian11:~# vi /etc/hosts

root@debian11:~# cat /etc/hosts
127.0.0.1      localhost
10.0.2.46     debian11.ittraining.loc debian11
10.0.2.54     web01.i2tch.loc web01
10.0.2.55     web02.i2tch.loc web02
10.0.2.56     web03.i2tch.loc web03
10.0.2.57     web04.i2tch.loc web04
10.0.2.52     targeta.i2tch.loc targeta
10.0.2.53     targetb.i2tch.loc targetb

# The following lines are desirable for IPv6 capable hosts
```

```
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

root@debian11:~# exit
logout

trainee@debian11:~/ansible/roles$
```

Connectez-vous en ssh à targeta et testez la configuration de sudo :

```
trainee@debian11:~/ansible/roles$ ssh targeta
Debian GNU/Linux 9
Linux targeta.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 20 16:16:07 2023 from 10.0.2.46
trainee@targeta:~$
```

Modifiez le contenu du fichier **/etc/apt/sources.list** :

```
trainee@targeta:~$ sudo vi /etc/apt/sources.list

trainee@targeta:~$ cat /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 9.6.0 _Stretch_ - Official amd64 NETINST 20181110-11:34]/ stretch main
#deb cdrom:[Debian GNU/Linux 9.6.0 _Stretch_ - Official amd64 NETINST 20181110-11:34]/ stretch main
```

```
deb http://archive.debian.org/debian/ stretch main
deb-src http://archive.debian.org/debian/ stretch main

deb http://archive.debian.org/debian-security stretch/updates main
deb-src http://archive.debian.org/debian-security stretch/updates main

trainee@targeta:~$ sudo apt update
Ign:1 http://archive.debian.org/debian stretch InRelease
Réception de:2 http://archive.debian.org/debian-security stretch/updates InRelease [59,1 kB]
Réception de:3 http://archive.debian.org/debian stretch Release [118 kB]
Réception de:4 http://archive.debian.org/debian stretch Release.gpg [3 177 B]
Réception de:5 http://archive.debian.org/debian-security stretch/updates/main Sources [413 kB]
Réception de:6 http://archive.debian.org/debian-security stretch/updates/main amd64 Packages [782 kB]
Réception de:7 http://archive.debian.org/debian-security stretch/updates/main Translation-en [368 kB]
Réception de:8 http://archive.debian.org/debian stretch/main Sources [6 736 kB]
Réception de:9 http://archive.debian.org/debian stretch/main amd64 Packages [7 080 kB]
Réception de:10 http://archive.debian.org/debian stretch/main Translation-fr [1 894 kB]
Réception de:11 http://archive.debian.org/debian stretch/main Translation-en [5 377 kB]
22,8 Mo réceptionnés en 27s (843 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
353 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Supprimez le paquet **openjdk-8-jre** :

```
trainee@targeta:~$ sudo apt-get -y remove openjdk-8-jre
...
trainee@targeta:~$ exit
déconnexion
Connection to targeta closed.
```

Connectez-vous en ssh à targetb et testez la configuration de sudo :

```
trainee@debian11:~/ansible/roles$ ssh targetb
The authenticity of host 'targetb (10.0.2.53)' can't be established.
ECDSA key fingerprint is SHA256:Rg0sp/XI7JHNq+oIfHKw+jkHdtTnBIh+Dd7kVmHRxtU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'targetb,10.0.2.53' (ECDSA) to the list of known hosts.
trainee@targetb's password:
Last login: Sun Mar 21 08:53:18 2021 from 10.0.2.10
```

```
[trainee@targetb ~]$ yum clean all
Loaded plugins: fastestmirror, langpacks
Reposdata is over 2 weeks old. Install yum-cron? Or run: yum makecache fast
Cleaning repos: base extras updates
Other repos take up 119 M of disk space (use --verbose for details)
```

```
[trainee@targetb ~]$ sudo yum -y makecache
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: centos.mirror.ate.info
 * extras: mirror.plusserver.com
 * updates: mirror.plusserver.com
base
| 3.6 kB  00:00:00
extras
| 2.9 kB  00:00:00
updates
| 2.9 kB  00:00:00
(1/6): extras/7/x86_64/other_db
| 147 kB  00:00:00
(2/6): base/7/x86_64/filelists_db
| 7.2 MB  00:00:00
(3/6): base/7/x86_64/other_db
| 2.6 MB  00:00:00
(4/6): extras/7/x86_64/filelists_db
| 277 kB  00:00:00
```

```
(5/6): updates/7/x86_64/other_db
| 1.0 MB  00:00:00
(6/6): updates/7/x86_64/filelists_db
| 7.8 MB  00:00:00
Metadata Cache Created
```

Supprimez le paquet **java-1.8.0-openjdk** :

```
[trainee@targetb ~]$ sudo yum -y remove java-1.8.0-openjdk
...
[trainee@targetb ~]$ exit
logout
Connection to targetb closed.
trainee@debian11:~/.ansible/roles$
```

Exécutez **ansible-playbook** et constatez le résultat :

```
trainee@debian11:~/.ansible/roles$ ansible-playbook -i inventory playbook.yaml -l linux
```

```
PLAY [all]
```

```
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
```

```
ok: [targeta]
```

```
ok: [targetb]
```

```
TASK [exemple01.java : install jre (Debian)]
```

```
*****
*****
```

```
skipping: [targetb]
```

```
changed: [targeta]
```

```

TASK [exemple01.java : install jre (CentOS)]
*****
*****
skipping: [targeta]
changed: [targetb]

PLAY RECAP
*****
*****
targeta                : ok=2    changed=1    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0
targetb                : ok=2    changed=1    unreachable=0    failed=0    skipped=1    rescued=0
ignored=0

```

Les conditions peuvent être combinées grâce à **and** et **or**. En voici un exemple :

```

tasks:
  - name: "Extinction des systèmes CentOS 6 et Debian 7"
    command: /sbin/shutdown -t now
    when: (ansible_distribution == "CentOS" and ansible_distribution_major_version == "6") or
          (ansible_distribution == "Debian" and ansible_distribution_major_version == "7")

```

## LAB #5 - La Commande ansible-vault

La commande **ansible-vault** permet de créer et d'utiliser des fichiers cryptés, par exemple, ceux qui contiennent des mots de passe.

Pour illustrer son utilisation, créez le Rôle **vault** :

```

trainee@debian11:~/ansible/roles$ mkdir vault

```

Éditez le fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** :

```
trainee@debian11:~/ansible/roles$ vi /home/trainee/.ansible/roles/group_vars/all.yaml
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/group_vars/all.yaml
---
secret: gardezmoisecret
```

Modifiez le fichier **/home/trainee/.ansible/roles/playbook.yaml** :

```
trainee@debian11:~/ansible/roles$ vi playbook.yaml
trainee@debian11:~/ansible/roles$ cat playbook.yaml
---
- hosts: all
  tasks:
  - debug:
    msg: "Le secret est {{ secret }}"
```

Exécutez la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l localhost

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

TASK [Gathering Facts]
*****
*****

ok: [localhost]

TASK [debug]
*****
*****

ok: [localhost] => {
  "msg": "Le secret est gardezmoisecret"
}
```

**PLAY RECAP**

```
*****  
*****  
localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0  
ignored=0
```

**Important** : L'exécution démontre que la variable **secret** est bien lue.

## 5.1 - Crypter le Fichier

Cryptez maintenant le fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** en utilisant la commande **ansible-vault** :

```
trainee@debian11:~/ansible/roles$ ansible-vault encrypt group_vars/all.yaml  
New Vault password: fenestros  
Confirm New Vault password: fenestros  
Encryption successful  
trainee@debian11:~/ansible/roles$
```

**Important** : Notez que le mot de passe **fenestros** ne sera pas en clair.

**Important** : Notez que le mot de passe est pour la commande **ansible-vault** et non pas uniquement pour l'action de cryptage en cours.

Constatez maintenant le contenu du fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** :

```
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/group_vars/all.yaml
$ANSIBLE_VAULT;1.1;AES256
33343664333931323331346338346439613439633633396262393733663463363438663163383761
6566653230323032396434356132653262313962633265630a366436666261363933343663623131
63373230663530313864373236316465333464323131643933363664303332336261353732623064
3538303864633035320a376235333637656534376638613661303765373165383936653336646562
31376535333861616165346433306230366231333139323062366432333033386366
```

Exécutez de nouveau la commande **ansible-playbook** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l localhost
```

```
PLAY [all]
```

```
*****
*****
```

```
ERROR! Attempting to decrypt but no vault secrets found
```

**Important** : Notez l'erreur due au fait que le fichier a été crypté.

Pour indiquer à la commande **ansible-playbook** que le fichier qui doit être lu soit crypté, utilisez l'option **-ask-vault-pass** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l localhost
```

```
PLAY [all]
```

```
*****
*****
```

```
ERROR! Attempting to decrypt but no vault secrets found
```

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l localhost --ask-vault-pass
Vault password: fenestros
```

```
PLAY [all]
```

```
*****
*****
TASK [Gathering Facts]
*****
*****
ok: [localhost]

TASK [debug]
*****
*****
ok: [localhost] => {
  "msg": "Le secret est gardezmoisecret"
}

PLAY RECAP
*****
*****
localhost           : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

**Important** : Notez que le mot de passe **fenestros** ne sera pas en clair.

## 5.2 - Editer le Fichier

Pour éditer le fichier, utilisez la sous-commande **edit** de la commande **ansible-vault** :

```
trainee@debian11:~/ansible/roles$ ansible-vault edit group_vars/all.yaml
Vault password: fenestros
```

**Important** : Notez que le mot de passe **fenestros** ne sera pas en clair.

Le fichier est chargé en mémoire pour édition :

```
---  
secret: gardezlesecret  
~  
~  
~  
~  
~  
~  
~  
~  
~  
-- INSERT --
```

Modifiez le secret et sauvegardez le fichier :

```
---  
secret: gardezlesecret  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
:X
```

Constatez que le contenu du fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** est toujours crypté :

```

trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/group_vars/all.yaml
$ANSIBLE_VAULT;1.1;AES256
36633861376466313363373336643832646663666334633931663262666361306236386232316461
3331633863386666626230663038653136653830303465660a353337316634363465663962336365
61396430313965653666333738376663336139613566636437353935653232306264336235333538
6439366539653436310a666361313237623137666634663439343132393532623962353163303364
37626635363832333934363930383265313932663230356234343235643935353535

```

Exécutez de nouveau la commande **ansible-playbook** avec l'option **-ask-vault-pass** :

```

trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l localhost --ask-vault-pass
Vault password: fenestros

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

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

TASK [debug]
*****
*****
ok: [localhost] => {
  "msg": "Le secret est gardezlesecret"
}

PLAY RECAP
*****
*****
localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0

```

```
ignored=0
```

**Important** : Notez que le mot de passe **fenestros** ne sera pas en clair.

### 5.3 - Décrypter le Fichier

Il est possible de décrypter le fichier en utilisant la sous-commande **decrypt** de la commande **ansible-vault** :

```
trainee@debian11:~/ansible/roles$ ansible-vault decrypt group_vars/all.yaml
Vault password: fenestros
Decryption successful
```

**Important** : Notez que le mot de passe **fenestros** ne sera pas en clair.

Constatez que le contenu du fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** n'est plus crypté :

```
trainee@debian11:~/ansible/roles$ cat /home/trainee/.ansible/roles/group_vars/all.yaml
---
secret: gardezlesecret
```

### 5.4 - Utilisation de Mots de Passe Aléatoires

Installez le paquet **pwgen** :

```
trainee@debian11:~/ansible/roles$ which pwgen
```

```
trainee@debian11:~/ansible/roles$ su -
Password: fenestros
root@debian11:~# apt install pwgen
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libopengl0 linux-headers-5.10.0-15-amd64 linux-headers-5.10.0-15-common
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  pwgen
0 upgraded, 1 newly installed, 0 to remove and 22 not upgraded.
Need to get 19.6 kB of archives.
After this operation, 52.2 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 pwgen amd64 2.08-2 [19.6 kB]
Fetched 19.6 kB in 0s (749 kB/s)
Selecting previously unselected package pwgen.
(Reading database ... 206562 files and directories currently installed.)
Preparing to unpack .../pwgen_2.08-2_amd64.deb ...
Unpacking pwgen (2.08-2) ...
Setting up pwgen (2.08-2) ...
Processing triggers for man-db (2.9.4-2) ...
root@debian11:~# exit
logout

trainee@debian11:~/ansible/roles$
```

Utilisez maintenant la commande **pwgen** pour générer un mot de passe aléatoire :

```
trainee@debian11:~/ansible/roles$ pwgen 16 1
di3Be1AiPayeehai
```

Utilisez pwgen pour créer un mot de passe aléatoire contenu dans un fichier :

```
trainee@debian11:~/ansible/roles$ pwgen 16 1 > vault-password

trainee@debian11:~/ansible/roles$ ls
backend.j2  debug  exemple01.java  geerlingguy.java  group_vars  haproxy  host_vars  inventory  playbook.yaml
tomcat  vault  vault-password

trainee@debian11:~/ansible/roles$ cat vault-password
oo3aeph5PaiVeiBu
```

Utilisez maintenant le mot de passe aléatoire contenu dans le fichier **vault-password** pour crypter le fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** :

```
trainee@debian11:~/ansible/roles$ ansible-vault encrypt group_vars/all.yaml --vault-password-file vault-password
Encryption successful
```

Constatez le contenu du fichier **/home/trainee/.ansible/roles/group\_vars/all.yaml** :

```
trainee@debian11:~/ansible/roles$ cat group_vars/all.yaml
$ANSIBLE_VAULT;1.1;AES256
61326364653334653935643735623863353362333536396638356362376631633732306332316661
6662333436336237346433363637393666623165643438620a343436366136386637343238303063
38383866386631363834623461323334313030623136333334646335626165363263373966373663
3137333330653635620a396133346264656335633530353966383930613731626639393631623066
31323331383563373830306331653166326563333135326631363461313666313864
```

Exécutez de nouveau la commande **ansible-playbook** avec l'option **-vault-password-file** :

```
trainee@debian11:~/ansible/roles$ ansible-playbook -i inventory playbook.yaml -l localhost --vault-password-file
vault-password

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

## TASK [Gathering Facts]

```
*****  
*****
```

```
ok: [localhost]
```

## TASK [debug]

```
*****  
*****
```

```
ok: [localhost] => {  
  "msg": "Le secret est gardezlesecret"  
}
```

## PLAY RECAP

```
*****  
*****
```

```
localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0  
ignored=0
```

## LAB #6 - Ansible par la Pratique

Connectez-vous à la machine virtuelle **CentOS7\_10.0.2.51\_SSH**.

### 6.1 - Instructions

Il vous est demandé d'automatiser avec Ansible :

- l'installation du serveur vsftpd,
- la mise à jour éventuelle de firewalld,
- la création d'une règle dans firewalld pour le trafic vers le serveur vsftpd,
- la création sécurisée de deux comptes ftp :
  - user : mike

- mot de passe : toto
- user : bob
  - mot de passe : tata

Installez donc Ansible à partir des dépôts :

```
[trainee@centos7 ~]$ su -  
Mot de passe : fenestros  
Dernière connexion : jeudi 5 septembre 2019 à 18:28:51 CEST sur pts/0  
[root@centos7 ~]# yum install epel-release  
[root@centos7 ~]# yum install ansible
```

Vérifiez ensuite la présence des exécutable :

```
[root@centos7 ~]# which ansible  
/bin/ansible  
[root@centos7 ~]# which ansible-playbook  
/bin/ansible-playbook
```

Afin de vous aider, on vous fournit deux fichiers :

```
[root@centos7 ~]# cat vsftpd.yml  
---  
- name: ftp server install  
  hosts: localhost  
  gather_facts: no  
  become: yes  
  vars_files:  
    - users.yml  
  tasks:  
    - name: latest vsftpd version  
      yum:  
        name: vsftpd  
        state: latest
```

```
- name: latest firewalld version
  name: firewalld
  state: latest
- name: vsftpd
  service:
    name vsftpd
    enabled: true
    state: started
- name: firewalld
  service:
    name: firewalld
    enabled: true
    state: started
- name: firewalld allows ftp
  firewalld:
    permanent: yes
    immediate: yes
    state: enabled
- name: Create_FTP_users
  user:
    password: "{{ item.password | password_hash(sha512) }}"
    with_items: "{{ ftp_users }}"
```

```
[root@centos7 ~]# ansible-vault view secrets.yml
Vault password: fenestros
ftp_users:
- { username: mike, password: toto }
- { username: bob, password: tata }
```

Lors de l'exécution de la commande `ansible-playback`, vous obtenez le résultat suivant :

```
[root@centos7 ~]# ansible-playbook vsftpd.yml
[WARNING]: Could not match supplied host pattern, ignoring: all
```

```
[WARNING]: provided hosts list is empty, only localhost is available
```

```
ERROR! Syntax Error while loading YAML.
```

```
The error appears to have been in '/root/vsftpd.yml': line 14, column 13, but may be elsewhere in the file depending on the exact syntax problem.
```

```
The offending line appears to be:
```

```
- name: latest firewalld version
  name: firewalld
  ^ here
```

```
exception type: <class 'yaml.scanner.ScannerError'>
exception: mapping values are not allowed in this context
in "<unicode string>", line 14, column 13
```

A vous de jouer !

---

Copyright © 2024 Hugh Norris.