

Version - **2023.01**

Dernière mise-à-jour : 2020/01/30 03:44

LCF803 - Rôles, Gabarits, Variables et Facts

Contenu du Module

- **LCF803 - 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 #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@centos8 ~]$ mkdir /home/trainee/.ansible/roles/exemple01.java/  
[trainee@centos8 ~]$ cd /home/trainee/.ansible/roles/exemple01.java/  
[trainee@centos8 exemple01.java]$ mkdir defaults tasks templates  
[trainee@centos8 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@centos8 exemple01.java]$ vi /home/trainee/.ansible/roles/exemple01.java/tasks/main.yaml  
[trainee@centos8 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@centos8 exemple01.java]$ vi /home/trainee/.ansible/roles/exemple01.java/defaults/main.yaml
[trainee@centos8 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@centos8 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@centos8 exemple01.java]$ mkdir /home/trainee/.ansible/roles/tomcat/
```

```
[trainee@centos8 exemple01.java]$ cd /home/trainee/.ansible/roles/tomcat/  
[trainee@centos8 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@centos8 tomcat]$ vi /home/trainee/.ansible/roles/tomcat/tasks/main.yaml  
[trainee@centos8 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@centos8 tomcat]$ vi /home/trainee/.ansible/roles/tomcat/meta/main.yaml  
[trainee@centos8 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@centos8 tomcat]$ vi /home/trainee/.ansible/roles/playbook.yaml  
[trainee@centos8 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@ansible:~/ansible/roles/tomcat$ cd /home/trainee/.ansible/roles/
trainee@ansible:~/ansible/roles$ cp ~/inventory .
```

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

```
[trainee@centos8 roles]$ tree
```

```
.
├── exemple01.java
│   ├── defaults
│   │   └── main.yaml
│   ├── tasks
│   │   └── main.yaml
│   └── 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-8.yml
│   ├── Debian-9.yml
│   ├── Fedora.yml
│   ├── FreeBSD.yml
│   ├── RedHat-7.yml
│   ├── RedHat-8.yml
│   ├── Ubuntu-12.yml
│   ├── Ubuntu-14.yml
│   ├── Ubuntu-16.yml
│   ├── Ubuntu-18.yml
│   └── Ubuntu-20.yml
├── inventory
├── playbook.yaml
├── tomcat
│   ├── meta
│   │   └── main.yaml
│   └── tasks
│       └── main.yaml
```

15 directories, 31 files

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

```
[trainee@centos8 roles]$ ansible-playbook -i inventory playbook.yaml -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@centos8 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: Wed Mar  9 13:49:26 2022 from 10.0.2.45
trainee@web01:~$ systemctl status tomcat8
● tomcat8.service - LSB: Start Tomcat.
   Loaded: loaded (/etc/init.d/tomcat8; generated; vendor preset: enabled)
   Active: active (running) since Wed 2022-03-09 13:50:11 CET; 1min 13s ago
     Docs: man:systemd-sysv-generator(8)
    CGroup: /system.slice/tomcat8.service
            └─11904 /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.headle
trainee@web01:~$ exit
déconnexion
Connection to web01 closed.
[trainee@centos8 roles]$
```

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

```
[trainee@centos8 roles]$ vi /home/trainee/.ansible/roles/tomcat/meta/main.yaml
[trainee@centos8 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 sur-chargée par la valeur **tree**.

Testez votre configuration :

```
[trainee@centos8 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@centos8 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: Wed Mar  9 13:53:57 2022 from 10.0.2.45  
trainee@web02:~$ which tree  
/usr/bin/tree  
trainee@web02:~$ exit  
déconnexion  
Connection to web02 closed.  
[trainee@centos8 roles]$ ssh web03  
Debian GNU/Linux 9  
Linux web03.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 12:12:51 2022 from 10.0.2.45
```

```
trainee@web03:~$ which tree
```

```
trainee@web03:~$ exit
```

```
déconnexion
```

```
Connection to web03 closed.
```

```
[trainee@centos8 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.

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@centos8 roles]$ mkdir /home/trainee/.ansible/roles/haproxy/
[trainee@centos8 roles]$ cd /home/trainee/.ansible/roles/haproxy/
[trainee@centos8 haproxy]$ mkdir defaults handlers tasks templates
```

```
[trainee@centos8 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@centos8 haproxy]$ vi /home/trainee/.ansible/roles/haproxy/tasks/main.yaml
[trainee@centos8 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@centos8 haproxy]$ vi /home/trainee/.ansible/roles/haproxy/handlers/main.yaml
[trainee@centos8 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@centos8 haproxy]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
[trainee@centos8 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-ssl3

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@centos8 haproxy]$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
[trainee@centos8 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@centos8 haproxy]$ cd ..
[trainee@centos8 roles]$ vi playbook.yaml
[trainee@centos8 roles]$ cat playbook.yaml
---
- hosts: all
  become: true
  roles:
    - haproxy
```

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

```
[trainee@centos8 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@centos8 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 15:23:48 2022 from 10.0.2.45
```

```
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 2022-03-09 12:03:47 CET; 3h 20min ago
```

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

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

```
Process: 9089 ExecReload=/bin/kill -USR2 $MAINPID (code=exited, status=0/SUCCESS)
```

```
Process: 9087 ExecReload=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRA_OPTS (code=exited, status=0/SUCCESS)
```

```
Main PID: 5394 (haproxy-systemd)
```

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

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

```
├─5394 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
```

```
├─9091 /usr/sbin/haproxy-master
```

```
└─9093 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds -sf 5400
```

```
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-ssl3
```

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@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
[trainee@centos8 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@centos8 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@centos8 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@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
[trainee@centos8 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@centos8 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@centos8 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 15:32:20 2022 from 10.0.2.45  
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@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
[trainee@centos8 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@centos8 roles]$ vi backend.j2
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/defaults/main.yaml
[trainee@centos8 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@centos8 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@centos8 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@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.cfg
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/templates/haproxy.http.cfg
[trainee@centos8 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-sslv3
{% 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@centos8 roles]$ vi /home/trainee/.ansible/roles/haproxy/tasks/main.yaml
[trainee@centos8 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@centos8 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@centos8 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-ssl3
```

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

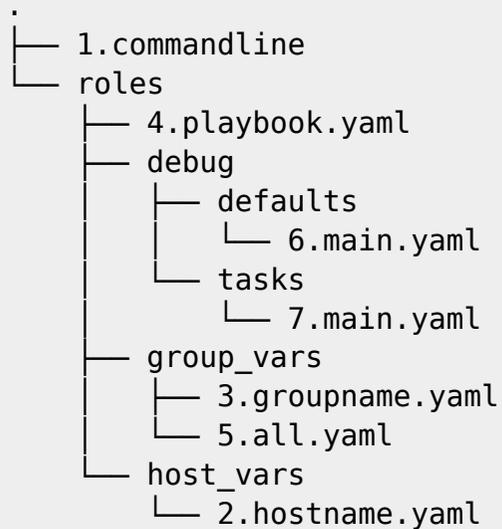
```
server web02 10.0.2.55:8080 check
server web03 10.0.2.56:8080 check
trainee@web04:~$ exit
déconnexion
Connection to web04 closed.
[trainee@centos8 roles]$
```



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 :



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

```
[trainee@centos8 roles]$ cd ../../
[trainee@centos8 ~]$ mkdir /home/trainee/.ansible/roles/debug
[trainee@centos8 ~]$ mkdir /home/trainee/.ansible/roles/debug/defaults
[trainee@centos8 ~]$ mkdir /home/trainee/.ansible/roles/debug/tasks
[trainee@centos8 ~]$
```

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

```
[trainee@centos8 ~]$ touch /home/trainee/.ansible/roles/debug/defaults/main.yaml
[trainee@centos8 ~]$ touch /home/trainee/.ansible/roles/debug/tasks/main.yaml
[trainee@centos8 ~]$
```

Vous obtiendrez :

```
[trainee@centos8 ~]$ cd .ansible/roles/
[trainee@centos8 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@centos8 roles]$ mkdir /home/trainee/.ansible/roles/group_vars
[trainee@centos8 roles]$ touch /home/trainee/.ansible/roles/group_vars/all.yaml
[trainee@centos8 roles]$
```

Vous obtiendrez :

```
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/debug/tasks/main.yaml
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/playbook.yaml
[trainee@centos8 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@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/debug/defaults/main.yaml
[trainee@centos8 roles]$ cat /home/trainee/.ansible/roles/debug/defaults/main.yaml
---
endroit: 'roles/debug/defaults/main.yaml'
```

Exécutez la commande **ansible-playbook** :

```
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/group_vars/all.yaml
[trainee@centos8 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@centos8 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@centos8 roles]$ vi playbook.yaml
[trainee@centos8 roles]$ cat playbook.yaml
---
- hosts: all
  roles:
    - { role: debug, endroit: 'playbook.yaml' }

```

Exécutez la commande **ansible-playbook** :

```

[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/group_vars/group1.yaml
[trainee@centos8 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@centos8 roles]$ mkdir /home/trainee/.ansible/roles/host_vars
[trainee@centos8 roles]$ vi /home/trainee/.ansible/roles/host_vars/localhost.yaml
[trainee@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/playbook.yaml
[trainee@centos8 roles]$ cat /home/trainee/.ansible/roles/playbook.yaml
---
- hosts: all
  roles:
    - debug
```

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

```
[trainee@centos8 roles]$ vi inventory
[trainee@centos8 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@centos8 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@centos8 roles]$ vi inventory
[trainee@centos8 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@centos8 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@centos8 roles]$ vi /home/trainee/.ansible/roles/host_vars/localhost.yaml
[trainee@centos8 roles]$ cat /home/trainee/.ansible/roles/host_vars/localhost.yaml
---
endroit: 'host_vars/localhost.yaml'

```

Exécutez la commande **ansible-playbook** :

```

[trainee@centos8 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@centos8 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@centos8 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@centos8 roles]$ vi inventory
[trainee@centos8 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@centos8 roles]$ vi playbook.yaml
[trainee@centos8 roles]$ cat playbook.yaml
---
```

```
- hosts: all
  become: true
  roles:
    - exemple01.java
```

Connectez-vous à la machine TargetA et supprimez le paquet **openjdk-8-jre** :

```
[trainee@centos8 roles]$ ssh 10.0.2.52
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 Mar  9 11:29:25 2022 from 10.0.2.45
trainee@targeta:~$ sudo apt-get -y remove openjdk-8-jre
...
trainee@targeta:~$ exit
déconnexion
Connection to 10.0.2.52 closed.
[trainee@centos8 roles]$
```

Utilisez la commande scp pour copier le fichier `authorized_keys` de la machine virtuelle centos8 vers la machine virtuelle targetb :

```
[trainee@centos8 roles]$ cd ~
[trainee@centos8 ~]$ scp .ssh/authorized_keys trainee@10.0.2.53:/home/trainee/.ssh/authorized_keys
trainee@10.0.2.53's password:
authorized_keys
100% 888    8.9KB/s   00:00
[trainee@centos8 ~]$ cd -
/home/trainee/.ansible/roles
```

```
[trainee@centos8 roles]$
```

Connectez-vous à la machine TargetB et supprimez le paquet **java-1.8.0-openjdk** :

```
[trainee@centos8 roles]$ ssh 10.0.2.53
Last login: Wed Mar  9 16:51:41 2022 from 10.0.2.45
[trainee@targetb ~]$ sudo yum -y remove java-1.8.0-openjdk
...
[trainee@targetb ~]$ exit
logout
Connection to 10.0.2.53 closed.
[trainee@centos8 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@centos8 roles]$ vi exemple01.java/tasks/main.yaml
[trainee@centos8 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 **centos8** :

```
[trainee@centos8 roles]$ su -
Password: fenestros
[root@centos8 ~]# vi /etc/hosts
[root@centos8 ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
::1      localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.2.45 centos8.ittraining.loc centos8
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
[root@centos8 ~]# exit
logout
[trainee@centos8 roles]$
```

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

```
[trainee@centos8 roles]$ ssh targeta
The authenticity of host 'targeta (10.0.2.52)' can't be established.
ECDSA key fingerprint is SHA256:sEfHBv9azmK60cjqF/aJgUc9jg56slNaZQdAUcvB0vE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'targeta' (ECDSA) to the list of known hosts.
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 Mar  9 16:52:16 2022 from 10.0.2.45

trainee@targeta:~$ sudo apt update
Réception de:1 http://security.debian.org/debian-security stretch/updates InRelease [53,0 kB]
Ign:2 http://ftp.fr.debian.org/debian stretch InRelease
Réception de:3 http://ftp.fr.debian.org/debian stretch-updates InRelease [93,6 kB]
```

```
Réception de:4 http://ftp.fr.debian.org/debian stretch Release [118 kB]
Réception de:5 http://ftp.fr.debian.org/debian stretch Release.gpg [3 177 B]
Réception de:6 http://security.debian.org/debian-security stretch/updates/main Sources [396 kB]
Réception de:7 http://security.debian.org/debian-security stretch/updates/main amd64 Packages [759 kB]
Réception de:8 http://security.debian.org/debian-security stretch/updates/main Translation-en [354 kB]
Réception de:9 http://ftp.fr.debian.org/debian stretch/main Sources [6 736 kB]
Réception de:10 http://ftp.fr.debian.org/debian stretch/main amd64 Packages [7 080 kB]
Réception de:11 http://ftp.fr.debian.org/debian stretch/main Translation-en [5 377 kB]
21,0 Mo réceptionnés en 7s (2 753 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
333 packages can be upgraded. Run 'apt list --upgradable' to see them.

trainee@targeta:~$ exit
déconnexion
Connection to targeta closed.
```

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

```
[trainee@centos8 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' (ECDSA) to the list of known hosts.
Last login: Wed Mar  9 16:56:33 2022 from 10.0.2.45

[trainee@targetb ~]$ yum clean all
Loaded plugins: fastestmirror, langpacks
Cleaning repos: base extras updates
Other repos take up 190 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

[trainee@targetb ~]$ exit
logout
Connection to targetb closed.
[trainee@centos8 roles]$
```

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

```
[trainee@centos8 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@centos8 roles]$ mkdir vault
```

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

```
[trainee@centos8 roles]$ vi /home/trainee/.ansible/roles/group_vars/all.yaml
[trainee@centos8 roles]$ cat /home/trainee/.ansible/roles/group_vars/all.yaml
---
secret: gardezmoisecret
```

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

```
[trainee@centos8 roles]$ vi playbook.yaml
[trainee@centos8 roles]$ cat playbook.yaml
---
- hosts: all
  tasks:
  - debug:
    msg: "Le secret est {{ secret }}"
```

Exécutez la commande **ansible-playbook** :

```
[trainee@centos8 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@centos8 roles]$ ansible-vault encrypt group_vars/all.yaml  
New Vault password: fenestros
```

```
Confirm New Vault password: fenestros
Encryption successful
[trainee@centos8 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@centos8 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@centos8 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@centos8 roles]$ ansible-playbook -i inventory playbook.yaml -l localhost

PLAY [all]
*****
*****
ERROR! Attempting to decrypt but no vault secrets found
[trainee@centos8 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@centos8 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 --
```



```
*****
*****
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@centos8 roles]$ ansible-vault decrypt group_vars/all.yml
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@centos8 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@centos8 roles]$ su -
Password: fenestros
[root@centos8 ~]#

[root@centos8 ~]# dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
Last metadata expiration check: 1:30:37 ago on Wed 09 Mar 2022 10:10:01 EST.
epel-release-latest-8.noarch.rpm
47 kB/s | 22 kB    00:00
Dependencies resolved.
=====
=====
Package                               Architecture                               Version
Repository                               Size
=====
=====
Installing:
  epel-release                           noarch                                       8-14.el8
@commandline                             22 k
=====
Transaction Summary
=====
=====
Install 1 Package
```

```
Total size: 22 k
Installed size: 32 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
1/1
  Installing     : epel-release-8-14.el8.noarch
1/1
  Running scriptlet: epel-release-8-14.el8.noarch
1/1
  Verifying      : epel-release-8-14.el8.noarch
1/1
Installed products updated.

Installed:
  epel-release-8-14.el8.noarch

Complete!

[root@centos8 ~]# dnf install pwgen
Extra Packages for Enterprise Linux 8 - x86_64
1.1 MB/s | 11 MB    00:09
Extra Packages for Enterprise Linux Modular 8 - x86_64
530 kB/s | 979 kB  00:01
Last metadata expiration check: 0:00:01 ago on Wed 09 Mar 2022 11:41:14 EST.
Dependencies resolved.
```

```
=====
Package
```

```
Architecture
```

```
Version
```

```
Repository                               Size
=====
Installing:
  pwgen                                x86_64                2.08-3.el8
  epel                                  31 k
Transaction Summary
=====
Install 1 Package

Total download size: 31 k
Installed size: 46 k
Is this ok [y/N]: y

[root@centos8 ~]# exit
logout
[trainee@centos8 roles]$
```

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

```
[trainee@centos8 roles]$ pwgen 16 1
di3Be1AiPayeehai
```

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

```
[trainee@centos8 roles]$ pwgen 16 1 > vault-password

[trainee@centos8 roles]$ ls
backend.j2  debug  exemple01.java  geerlingguy.java  group_vars  haproxy  host_vars  inventory  playbook.yaml
tomcat  vault  vault-password

[trainee@centos8 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@centos8 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@centos8 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@centos8 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
```