Sommaire

[1 Les Hosts 3](#_Toc526678867)

[1.1 Lister tous les hosts 3](#_Toc526678868)

[1.2 Gestion des hosts 3](#_Toc526678869)

[2 Les Modules 4](#_Toc526678870)

[2.1 Ping module 4](#_Toc526678871)

[2.2 Command module 4](#_Toc526678872)

[2.3 Package Managing (apt, yum etc…) module 4](#_Toc526678873)

[2.4 Service module 4](#_Toc526678874)

[2.5 Les boucles 5](#_Toc526678875)

[2.6 authorized\_key et user module 5](#_Toc526678876)

[2.7 Les Handlers et l'apache2\_module 5](#_Toc526678877)

[2.8 Copy module 6](#_Toc526678878)

[2.9 Pip module 6](#_Toc526678879)

[2.10 File module 6](#_Toc526678880)

[2.11 Register et Uri (Interacts with webservices) module 6](#_Toc526678881)

[2.12 Fail et Debug (vérification avant de poursuivre une task) module 7](#_Toc526678882)

[2.13 Template module 8](#_Toc526678883)

[2.14 Lineinfile module 8](#_Toc526678884)

[2.15 mysql\_db and mysql\_user modules 8](#_Toc526678885)

[2.16 when + shell modules 9](#_Toc526678886)

[2.17 wait\_for (condition/vérification avant de continuer une task) 9](#_Toc526678887)

[2.18 Include playbooks 10](#_Toc526678888)

[2.19 Projet 10](#_Toc526678889)

[3 Projet d'utilisation de playbooks 10](#_Toc526678890)

[3.1 Environnement 10](#_Toc526678891)

[3.2 Source 10](#_Toc526678892)

[4 Les rôles 11](#_Toc526678893)

[4.1 Structure en image 11](#_Toc526678894)

[4.2 Handlers 11](#_Toc526678895)

[4.3 Tasks 11](#_Toc526678896)

[4.4 Templates 12](#_Toc526678897)

[4.5 meta 12](#_Toc526678898)

[4.6 defaults 13](#_Toc526678899)

[4.7 vars 13](#_Toc526678900)

[4.8 files 13](#_Toc526678901)

[4.9 Playbook exécuteur de rôles 13](#_Toc526678902)

[4.10 Source 13](#_Toc526678903)

[5 Les variables 13](#_Toc526678904)

[5.1 Lister et afficher toutes les variables d'ansible d'un host 13](#_Toc526678905)

[5.2 Utiliser une variable dans un playbook 14](#_Toc526678906)

[5.3 Création de variable (defaults) 14](#_Toc526678907)

[5.4 group\_vars (variable globale pour tous les rôles) 14](#_Toc526678908)

[5.5 Vault chiffrer les données 15](#_Toc526678909)

[6 ansible galaxy (repo de playbooks) 16](#_Toc526678910)

[7 Debug 16](#_Toc526678911)

[7.1 Step (éxécuter tâche par tâche) 16](#_Toc526678912)

[8 Optimisation: 16](#_Toc526678913)

[8.1 Gather\_fact 16](#_Toc526678914)

[8.2 update\_cache dans les packages management 17](#_Toc526678915)

[8.3 –limit pour limiter les hosts de test 17](#_Toc526678916)

[8.4 Les tags 17](#_Toc526678917)

[8.5 Changed\_when 18](#_Toc526678918)

[9 Astuces 18](#_Toc526678919)

# Les Hosts

## Lister tous les hosts

[root@CM ~]# ansible --list-hosts all

## Gestion des hosts

On peut gérer les hosts depuis la configuration sur la configuration par défaut d'ansible

[root@CM ~]# nano /etc/ansible/hosts

[app]

node1 node2

[db]

db

[all\_srv:children]

app

db

On peut utiliser notre propre fichier de configuration de hosts, comme ça on a une config déjà prête pour être lancer depuis n'importe quelle machine de contrôle, pour cela il faut créer un fichier **ansible.cfg et un fichier d'host**



Le fichier dev va contenir la config des hosts

Et notre **ansible.cfg** contiendra :

[defaults]

inventory = ./dev

# Les Modules

## Ping module

[root@CM ~]# ansible all -m ping

## Command module

***Lien :*** [*https://docs.ansible.com/ansible/2.5/modules/command\_module.html#command-module*](https://docs.ansible.com/ansible/2.5/modules/command_module.html#command-module)

---

- hosts: all

tasks:

- name: get server hostname

command: hostname

## Package Managing (apt, yum etc…) module

***Lien :***[*https://docs.ansible.com/ansible/latest/modules/yum\_module.html*](https://docs.ansible.com/ansible/latest/modules/yum_module.html)

---

- hosts: database

become: true

tasks:

- name: install mysql-server

yum: name=mysql-server state=latest update\_cache=yes

* Become : utiliser la commande sudo
* update\_cache : Forcez yum à vérifier si le cache (yum update) est obsolète et à télécharger à nouveau si nécessaire. N'a d'effet que si l'état est present ou latest.

***Conseil :*** *À ne faire qu'au début dans la 1ere task d'installation*

* state :
  + present ou installed : garantira simplement qu'un paquet souhaité est installé.
  + Latest : mettra à jour le paquet spécifié s'il ne s'agit pas de la dernière version disponible.
  + absent et removed enlèvera le paquet spécifié.

## Service module

***Lien :***[*https://docs.ansible.com/ansible/latest/modules/service\_module.html*](https://docs.ansible.com/ansible/latest/modules/service_module.html)

---

- hosts: webserver

become: true

tasks:

- name: ensure apache2 started

service: name=apache2 state=started enabled=yes

* enabled : Si le service doit démarrer au démarrage

## Les boucles

---

- hosts: webserver

become: true

tasks:

- name: install web components

apt: name={{item}} state=present update\_cache=yes

with\_items:

- apache2

- libapache2-mod-wsgi

- python-pip

- python-virtualenv

## authorized\_key et user module

---

- hosts: app

tasks:

- name: Add the user 'test' with password 'test'

user:

name: test

password: test

- name: Add our local public to the remote user authorized\_keys

authorized\_key:

user: test

state: present

key: "{{ lookup('file', '~/.ssh/id\_rsa.pub') }}"

## Les Handlers et l'apache2\_module

***Lien :***[*https://docs.ansible.com/ansible/2.5/modules/apache2\_module\_module.html*](https://docs.ansible.com/ansible/2.5/modules/apache2_module_module.html)

Les handlers sont des events qui s'exécuterons lors d'un changement dans une task

---

- hosts: webserver

become: true

tasks:

- name: ensure mod\_wsgi enabled

apache2\_module: state=present name=wsgi

notify: restart apache2

handlers:

- name: restart apache2

service: name=apache2 state=restarted

## Copy module

***Lien :*** [*https://docs.ansible.com/ansible/2.5/modules/copy\_module.html?highlight=copy*](https://docs.ansible.com/ansible/2.5/modules/copy_module.html?highlight=copy)

Copie des fichiers sur le serveur distant

- name: example copying file with owner and permissions

copy:

src: /srv/myfiles/foo.conf

dest: /etc/foo.conf

owner: foo

group: foo

mode: 0644

## Pip module

***Lien :*** [*https://docs.ansible.com/ansible/2.5/modules/pip\_module.html?highlight=pip*](https://docs.ansible.com/ansible/2.5/modules/pip_module.html?highlight=pip)

- name: setup python virtualenv

pip: requirements=/var/www/demo/requirements.txt virtualenv=/var/www/demo/.venv

notify: restart apache2

## File module

***Lien :*** [*https://docs.ansible.com/ansible/2.5/modules/file\_module.html#file-module*](https://docs.ansible.com/ansible/2.5/modules/file_module.html#file-module)

Gestion et manipulation des fichiers et des répertoires

- name: de-activate default apache site

file: path=/etc/apache2/sites-enabled/000-default.conf state=absent

notify: restart apache2

- name: activate demo apache site

file: src=/etc/apache2/sites-available/demo.conf dest=/etc/apache2/sites-enabled/demo.conf state=link

notify: restart apache2

## Register et Uri (Interacts with webservices) module

***Lien :*** [*https://docs.ansible.com/ansible/2.5/modules/uri\_module.html*](https://docs.ansible.com/ansible/2.5/modules/uri_module.html)

Interagit avec les services Web HTTP et HTTPS et prend en charge les mécanismes d'authentification Digest, Basic et WSSE HTTP.

**register :** stocker le résultat dans une variable

---

- hosts: app

tasks:

- name: Check that you can connect (GET) to a page and it returns a status 200

uri:

url: http://ip.42.pl/raw

return\_content: yes

register: webpage

*# get only the content*

- name: debug content

debug: var=webpage.content

## Fail et Debug (vérification avant de poursuivre une task) module

***Lien :*** [*https://docs.ansible.com/ansible/2.5/modules/fail\_module.html*](https://docs.ansible.com/ansible/2.5/modules/fail_module.html)

J'ai une préférence pour fail au lieu de failed\_when car avec le module fail on peut afficher un message lors d'une erreur

**Les conditions sont les mêmes que sur python donc bien respecter le type (int, str etc …) de comparaison !**

---

*### test from the loadbalancer console*

- hosts: loadbalancer

tasks:

- name: verify backend index response

uri: url=http://{{item}} return\_content=yes

*## groups.webserver est une varialbe global qui retourne le nom de domaine de chaque hôte*

with\_items: groups.webserver

register: response

*## chechking http status of all webserver*

- name: Cheking webpage status

fail: msg="index failed to return content"

when: 200 != {{item.item}}

with\_items: "{{response.status}}"

- name: debug content

debug: msg="If you see this message then it means everything is alright ..."

## Template module

***Lien :***[*https://docs.ansible.com/ansible/latest/modules/template\_module.html*](https://docs.ansible.com/ansible/latest/modules/template_module.html)

Les templates sont de simples fichiers texte que nous utilisons dans Ansible et qui contiennent tous vos paramètres de configuration. La plupart du temps, nous les utilisons pour remplacer des fichiers de configuration ou placer des documents sur le serveur.

---

- hosts: loadbalancer

become: true

tasks:

- name: configure nginx site

template: src=templates/nginx.conf.j2 dest=/etc/nginx/sites-available/demo mode=0644

notify: restart nginx

handlers:

- name: restart nginx

service: name=nginx state=restarted



## Lineinfile module

***Lien :*** [*https://docs.ansible.com/ansible/2.5/modules/lineinfile\_module.html?highlight=lineinfile*](https://docs.ansible.com/ansible/2.5/modules/lineinfile_module.html?highlight=lineinfile)

Permet de préserver le contenu du fichier tel qu'il est et de ne changer qu'une ligne

---

- hosts: database

become: true

tasks:

- name: ensure we can connect to mysql from external

lineinfile: dest=/etc/mysql/my.cnf regexp=^bind-address line="bind-address = 0.0.0.0"

notify: restart mysql

handlers:

- name: restart mysql

service: name=mysql state=restarted

## mysql\_db and mysql\_user modules

***Lien (mysql\_db) :*** [*https://docs.ansible.com/ansible/2.5/modules/mysql\_db\_module.html?highlight=mysql\_db*](https://docs.ansible.com/ansible/2.5/modules/mysql_db_module.html?highlight=mysql_db)

***Lien (mysql\_user) :***

[*https://docs.ansible.com/ansible/2.5/modules/mysql\_user\_module.html?highlight=mysql\_user*](https://docs.ansible.com/ansible/2.5/modules/mysql_user_module.html?highlight=mysql_user)

- name: create demo database

mysql\_db: name=demo state=present

- name: create demo user

mysql\_user: name=demo password=demo priv=demo.\*:ALL host='%' state=present

## when + shell modules

---

- hosts: app

vars:

web\_site\_name: monsite

tasks:

- name: get active sites

*# ls -1 permet d'afficher le resultat ligne par ligne*

shell: ls -1 /etc/nginx/sites-enabled

register: active

- name: de-activate sites

file: path=/etc/nginx/sites-enabled/{{ item }} state=absent

with\_items: active.stdout\_lines

*# on ne garde que notre site*

when: item not in {{ web\_site\_name }}

notify: restart nginx

## wait\_for (condition/vérification avant de continuer une task)

***Lien :*** <https://docs.ansible.com/ansible/2.5/modules/wait_for_module.html>

**Utilisation :**

* Vous pouvez attendre un laps de temps défini.
* Vérifier qu'un port est ouvert ou fermé est utile lorsque les services ne sont pas immédiatement disponibles après le retour de leurs scripts d'initialisation
* C'est également utile lorsque vous démarrez des invités avec le module virt et qu'il est nécessaire de faire une pause jusqu'à ce qu'ils soient prêts.
* Attendre qu'une expression rationnelle corresponde à une chaîne devant figurer dans un fichier.
* Attendre qu'un fichier soit disponible ou absent du système de fichiers.

*# Restart mysql*

- hosts: database

become: true

tasks:

- service: name=mysql state=restarted

- wait\_for: port=3306 state=started

## Include playbooks

Inclure des playbooks dans un autre fichier

---

- include: playbook1.yml

- include: playbook2.yml

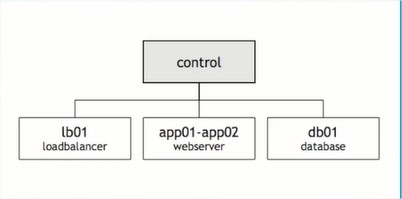
- include: playbook3.yml

## Projet



# Projet d'utilisation de playbooks

## Environnement

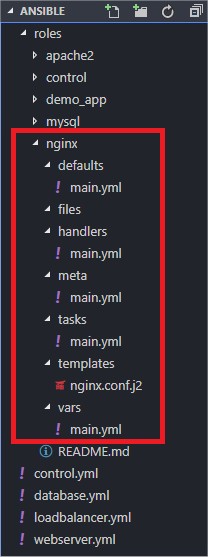


## Source



# Les rôles

## Structure en image



## Handlers

Contient les handlers (event) pouvant être utilisés par ce rôle ou même en dehors de ce rôle.

---

- name: restart nginx

service: name=nginx state=restarted

## Tasks

Contient la liste principale des tâches à exécuter par le rôle.

---

- name: install tools

apt: name={{item}} state=present update\_cache=yes

with\_items:

- python-httplib2

- name: install nginx

apt: name=nginx state=present update\_cache=yes

- name: configure nginx site

template: src=nginx.conf.j2 dest=/etc/nginx/sites-available/demo mode=0644

notify: restart nginx

- name: de-activate default nginx site

file: path=/etc/nginx/sites-enabled/default state=absent

notify: restart nginx

- name: activate demo nginx site

file: src=/etc/nginx/sites-available/demo dest=/etc/nginx/sites-enabled/demo state=link

notify: restart nginx

- name: ensure nginx started

service: name=nginx state=started enabled=yes

## Templates

Contient des modèles pouvant être déployés étant destinés à être traités par le module templates

Exemple du fichier qui permet de configurer le loadbalancer ***" nginx.conf.j2"***

upstream demo {

{% for server in groups.webserver %}

server {{ server }};

{% endfor %}

}

server {

listen 80;

location / {

proxy\_pass http://demo;

}

}

## meta

Définit des métadonnées pour ce rôle



## defaults

Les variables par défaut qui seront à disposition du rôle.

## vars

De la même manière que defaults, il s’agit ici de variables qui seront à disposition du rôle, cependant, celles-ci ont en général vocation à être modifiées par l’utilisateur et elles prennent le dessus sur celles de defaults si elles sont renseignées

## files

Contient des fichiers pouvant être déployés étant destinés à être traités par le module copy

## Playbook exécuteur de rôles

***loadbalancer.yml***

---

- hosts: loadbalancer

become: true

roles:

- nginx

## Source



# Les variables

## Lister et afficher toutes les variables d'ansible d'un host

[root@CM ~]# ansible host -m setup

***Résultat dans un fichier :***



## Utiliser une variable dans un playbook

---

- hosts: all

tasks:

- debug: msg="Adresse ip :{{ ansible\_default\_ipv4.address }}"

## Création de variable (defaults)

***Roles/tasls/main.yml***

---

- name: create database

mysql\_db: name={{ db\_name }} state=present

- name: create user

mysql\_user: name={{ db\_user\_name }} password={{ db\_user\_pass }} priv={{ db\_name }}.\*:ALL

host='{{ db\_user\_host }}' state=present

***Roles/defaults/main.yml***

---

db\_name: myapp

db\_user\_name: dbuser

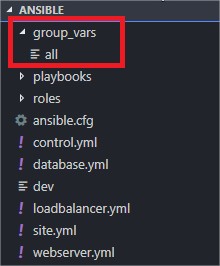
db\_user\_pass: dbpass

db\_user\_host: localhost

## group\_vars (variable globale pour tous les rôles)

Si on utilise une variable sur différents rôles alors le mieux est d'utiliser des variables globales en créant un dossier à la racine du projet group\_vars

Dans cet exemple le nom du fichier s'apelle **all**, on peut le renommer par un group d'host définit au préalable pour qu'elle ne soit utilisée que par ce groupe d'host



---

db\_name: demo

db\_user: demo

db\_pass: demo

## Vault chiffrer les données

Il est possible de chiffrer un fichier (en principe un fichier qui contient des données confidentielles comme un mot de passe) avec vault

Sur l'exemple sur les [group\_vars](#group_vars) on va chiffrer le mot de passe, pour cela on va changer un peu la structure de group\_vars



**Comment créer le fichier vault ?**

Créer un fichier (le nommé comme on veut ici vault) avec le contenu suivant :

---

vault\_db\_pass: demo

et lancer la commande suivante pour commencer le chiffrage du fichier contenant la variable de mot de passe

[root@CM ansible]# ansible-vault encrypt vault

Une fois que le fichier est chiffré, on peut donc utiliser ainsi variable dans notre fichier vars

---

db\_name: demo

db\_user: demo

db\_pass: "{{ vault\_db\_pass }}"

**Astuce pour editer le fichier :**

[root@CM ansible]# ansible-vault edit vault

si on tente d'éxécuter notre playbook on aura l'erreur suivante:

[root@CM ansible]# ansible-playbook site.yml

**ERROR: A vault password must be specified to decrypt path/le\_fichier\_en\_question**

On doit donc rentrer le mot de passe qu'on a saisie pour chiffrer le fichier selon 2 solutions :

**Solution 1**

Avec solution si on a créé plein de fichier vault il va nous le demander à chaque fois

[root@CM ansible]# ansible-playbook --ask-vault-pass site.yml

**Solution 2**

On va créer un fichier **AVEC NOTRE MDP VAULT HORS DU PROJET COMME CA SI JAMAIS ON FAIT UN COMMIT ALORS ON NE COMMITERA JAMAIS LE FICHER AVEC LE MDP**

[root@CM ansible]# echo "mon\_mdp" > dossier\_hors\_projet/.mdp\_vault.txt

[root@CM ansible]# chmod 0600

Et on va spécifier dans notre fichier **ansible.cfg** ou il trouvera le mot de passe :

[defaults]

inventory = ./dev

vault\_password\_file = ~/.vault\_pass.txt

# ansible galaxy (repo de playbooks)

<https://galaxy.ansible.com/>

# Debug

## Step (éxécuter tâche par tâche)

On va nous demander pour chaque tâche si on veut l'exécuter ou pas

[root@CM ~]# ansible-playbook test.yml --step

Perform task: TASK: be sure ntp is installed (N)o/(y)es/(c)ontinue: n

# Optimisation:

## Gather\_fact

Ce que j'entends par gather\_fact ce sont les variables qu'ansible récupère pour chaque playbook lancé, en gros ce qu'on récupère avec la commande (vu [ici](#list_var)):

[root@CM ~]# ansible host -m setup

Ansible le fait systématiquement pour chaque playbook lancer on peut ainsi pour gagner + de performance annuler la récupération de ces variables si jamais on ne les utilise pas en mettant **gather\_fact à false**

**Test de performance**

Fichier test.yml gathe\_fact = false:

---

- hosts: app

gather\_facts: false

tasks:

- debug: msg="With gather\_facts = false"

- debug: var=[ansible\_distribution,ansible\_hostname]

[root@CM ~]# time ansible-playbook test.yml

real 0m0.531s

user 0m0.465s

sys 0m0.064s

**Fichier test.yml gathe\_fact = true:**

---

- hosts: app

gather\_facts: true

tasks:

- debug: msg="With gather\_facts = false"

- debug: var=[ansible\_distribution,ansible\_hostname]

[root@CM ~]# time ansible-playbook test.yml

real 0m1.437s

user 0m0.993s

sys 0m0.114s

## update\_cache dans les packages management

utiliser update\_cache dans les packages management au début d'installation d'un package seulement

## –limit pour limiter les hosts de test

Si jamais notre playbook est testé sur plusieurs hosts on peut limiter le test pour 1 host avec l'option –limit

[root@CM ~]# time ansible-playbook test.yml --limit node1

## Les tags

Les tags peuvent nous permettent d'exécuter ou d'ignorer des tasks bien précise (on peut par exemple ignorer les tags de packages d'installation/vérification):

---

- hosts: app

gather\_facts: false

tasks:

- name: be sure ntp is installed

yum:

name: ntp

state: installed

tags: ntp

**lister les tags :**

[root@CM ~]# ansible-playbook test.yml --list-tags

**Exécuter les tasks avec un tag en particulier:**

[root@CM ~]# ansible-playbook test.yml –tags "ntp"

**Ignorer les tasks avec un tag en particulier:**

[root@CM ~]# ansible-playbook test.yml –skip-tags "ntp"

## Changed\_when

On peut être amener à faire des vérifications pour tester par exemple une commande ou autre le problème avec ça c'est qu'on aura on output et l'état de notre task sera toujours en **changed au lieu d'OK** juste à cause de l'ouput, on peut modifier ce comportement avec le module **changed\_when** qui va nous permettre de contrôler le changement d'une task

---

- hosts: app

gather\_facts: false

tasks:

- name: verify nginx service

command: service nginx status

changed\_when: false

dans ce cas on va tester juste si le service nginx existe (on s'en fou de l'ouput) et si la commande ne fail pas alors l'état sera **OK**

C:\Users\Hatim\Documents\My Screen Captures\Films et TV_2.jpg

# Astuces

Exécuter un playbook selon une distribution Linux

---

- hosts: all

vars:

- OS:

- CentOS

- Debian

tasks:

- name: are you centos ?

when: ansible\_distribution == OS[0]

debug:

msg: "Hello i am Centos and i am executing your task"

- name: are you centos ?

when: ansible\_distribution == OS[1]

debug:

msg: "Hello i am debian and i am executing your task"

## Jinja2-Homepage

<http://jinja.pocoo.org/>

## YAML-Tutorial

<http://learnxinyminutes.com/docs/yaml/>