



Support pédagogique Ansible - 4



Le programme de la formation

- Les Rôles
- Cas pratique avec l'installation de MediaWiki
- Les opérations séquentielles
- Ansible et le cloud

Les Rôles

Les rôles répondent à une problématique d'organisation et de réutilisation des fichiers de configuration.

```
├─ defaults
│   └─ main.yml
├─ files
├─ handlers
│   └─ main.yml
├─ meta
│   └─ main.yml
├─ README.md
├─ tasks
│   └─ main.yml
├─ templates
├─ tests
│   └─ inventory
│   └─ test.yml
└─ vars
    └─ main.yml
```

- Les rôles se présentent sous forme d'arborescence
- Permettent de réutiliser du code dans plusieurs `playbooks` de façon modulaire
- Et d'organiser le code dans les `playbooks`

Structure d'un rôle

- **files** : tous les fichiers à copier avec le module copy
- **templates** : fichiers de template Jinja
- **tasks** : liste des instructions à exécuter (dans le fichier main.yml, obligatoire)
- **handlers** : même chose pour les instructions handlers (fichier main.yml, obligatoire)
- **vars** : fichier contenant des déclarations de variables (fichier main.yml, obligatoire)
- **defaults** : valeurs par défaut (fichier main.yml, obligatoire)
- **meta** : dépendances du rôle et informations sur le rôle (fichier main.yml, obligatoire)

Pour générer une arborescence il faudrait toujours utiliser `ansible-galaxy init apache-galaxy` et supprimer ce qui n'est pas utile.

Reprenons l'installation de Apache mais cette fois avec un rôle

- Pour commencer, création du rôle

```
$ ansible-galaxy init apache-galaxy
```

- Configuration des actions pour l'installation de Apache dans le fichier `apache-galaxy/tasks/main.yml`

```
---
# tasks file for apache-galaxy

- name: "apache installation"
  yum:
    name: "httpd"
    state: "present"
- name: "apache service activation"
  service:
    name: "httpd"
    state: "started"
    enabled: yes
```

- Configuration du `Handler` qui va relancer `Apache` dans `apache-galaxy/handlers/main.yml`

```
---  
# handlers file for apache-galaxy  
  
- name: "apache restart"  
  service:  
    name: "httpd"  
    state: "restarted"
```

- Le `Playbook` `install-apache.yml` qui va lancer le rôle correspondant et exécuter en premier `tasks/main.yml`

```
- name: "Apache installation"  
  hosts: centos  
  roles:  
    - role: "apache-galaxy"
```

- Lancement du Playbook

```
ansible-playbook -i inventaire install-apache.yml -b -K
```

- Résultat sans surprise

```
PLAY [Apache installation] *****

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

TASK [apache-galaxy : apache installation] *****
ok: [centos]

TASK [apache-galaxy : apache service activation]*****
ok: [centos]

PLAY RECAP *****
centos                : ok=3    changed=0    unreachable=0    failed=0
```

- Ajoutons maintenant un nouveau rôle pour configurer Apache

```
$ ansible-galaxy init apache-config
```

- Définition des actions pour la configuration de Apache dans le fichier `roles/apache-config/tasks/main.yml`

```
---
# tasks file for apache-config
#
- name: "Apache configuration"
  template:
    src: "inventaire.conf.j2"
    # apache_conf_d pointe vers la conf apache
    dest: "/etc/httpd/conf.d/inventaire.conf"
    owner: "apache"
    group: "apache"
  # En cas de changement, redémarrage d'apache
  notify: [ "apache restart" ]
```


- Définition du `template` dans le fichier `roles/apache-config/templates/inventaire.conf.j2` qui va être déposé sur le serveur `Apache`

```
Alias /inventaire /var/www/html/inventory

# Donne des droits d'accès à tout le monde
# une ligne en plus
<Directory /var/www/html/inventory/>
    Order Allow,Deny
    Allow from All
</Directory>
```

- Modification du fichier `install-apache.yml` pour intégrer le nouveau rôle

```
- name: "Apache installation"
  hosts: centos
  roles:
    - role: "apache-galaxy"
    - role: "apache-config"
```

- Relance le `playbook`

```
$ ansible-playbook -i inventaire install-apache.yml -b -K
```

- Surprise deux taches ont changées, Apache configuration et le restart de Apache

```
PLAY [Apache installation] *****

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

TASK [apache-galaxy : apache installation] *****
ok: [centos]

TASK [apache-galaxy : apache service activation]
ok: [centos]

TASK [apache-config : Apache configuration] *****
changed: [centos]

RUNNING HANDLER [apache-galaxy : apache restart]
changed: [centos]

PLAY RECAP *****
centos                : ok=5    changed=2    unreachable=0    failed=0
```

Installation de MediaWiki

- Installation de `Apache` et `PHP` dans le rôle `apache`
- Installation de `MariaDB` dans un rôle `mariadb`
- Installation de `MediaWiki` avec trois rôles : `common`, `configuration` et `mariadb`

Installation de Apache et PHP

Pour l'installation de Apache reprenons le playbook et le rôle Apache.

- Le fichier inventaire

```
[apache]  
apache1
```

```
[mysql]  
mysql1
```

- Le playbook install-apache.yml

```
- name: "Installation apache"  
  hosts: apache  
  roles:  
    - role: "apache"
```

- Ajout l'installation de `PHP7` , en ajoutant une condition sur une variable booléenne dans le `roles/apache/tasks/main.yml`

```
# tasks file for apache-galaxy

- name: "apache installation"
  yum:
    name: "httpd"
    state: "present"

- name: "apache service activation"
  service:
    name: "httpd"
    state: "started"
    enabled: yes

- name: "install php7 packages"
  include: "php7-install.yml"
  when: php_install|default(False)|bool
```

- La configuration pour l'installation de `PHP7` dans le fichier `roles/apache/tasks/php7-install.yml`

```
- name: "epel activation"
  yum:
    name: "epel-release"
    state: present
- name: "remi repo activation"
  yum:
    name: "https://rpms.remirepo.net/enterprise/remi-release-7.rpm"
    state: present
- name: "install php70 packages"
  yum:
    name: "php,php-mysql,php-xml,php-mbstring,php-mcrypt,php-gd,php-intl"
    state: latest
    enablerepo: "remi-php70"
  changed_when: yes
  # We need to restart apache
  notify: [ "apache restart" ]
```

- Lancement du `playbook` une première fois, sans la variable qui est par défaut à `false`

```
ansible-playbook -i inventaire install-apache.yml -b -K
```

- Par défaut `PHP` n'est pas installé

```
TASK [apache : install php packages] *****  
skipping: [apache1]
```

- Lancement une deuxième fois avec la définition de la variable sur la ligne de commande

```
$ ansible-playbook -i inventaire install-apache.yml -b -K -e php_install=yes
```


- Changement de l'état de la tâche qui s'exécute et installation de **PHP7**

```
[...]
```

```
TASK [apache : remi repo activation] *****  
changed: [apache1]
```

```
TASK [apache : install php70 packages] *****  
changed: [apache1]
```

```
RUNNING HANDLER [apache : apache restart] *****  
changed: [apache1]
```

```
PLAY RECAP *****  
apache1                : ok=7    changed=3    unreachable=0    failed=0
```

- Sur le serveur **Apache1**

```
$ php --version  
PHP 7.0.30 (cli) (built: Apr 24 2018 21:28:23) ( NTS )
```

- La variable peut également être définie dans le fichier `inventaire`

```
[apache]
apache1

[apache:vars]
php_install=yes

[mysql]
mysql1
```

- ou encore dans le rôle
 - `vars/main.yml`
 - `defaults/main.yml`
- ou dans le `playbook`

```
- name: "Installation apache"
  hosts: apache
  roles:
    - role: "apache"
      php_install: yes
```

Installation de MariaDB

- Création du `roles/mariadb/tasks/main.yml`

```
- name: "mariadb-server installation"
  yum:
    name: "mariadb-server,MySQL-python"
    state: "installed"

- name: "start mariadb service"
  service:
    name: "mariadb"
    state: "started"
    enabled: yes
```

- Lancement du `playbook`

```
$ ansible-playbook -i inventaire install-mariadb.yml -b -K
```

```
PLAY [Install MariaDB] *****

TASK [mariadb : mariadb-server installation] *****
changed: [mysql1]

TASK [mariadb : start mariadb service] *****
changed: [mysql1]

PLAY RECAP *****
mysql1                : ok=2    changed=2    unreachable=0    failed=0
```

Installation de MediaWiki

Définition des variables pour créer la nouvelle base de données

- Les variables vont être partagées avec plusieurs `roles`, le plus simple est de les mettre dans un `role` commun et de faire appel à la notion de dépendance.
- `roles/mediawiki/common/defaults/main.yml`

```
# Default schema
mediawiki_db_name: "mediawiki"

# Default user for mediawiki
mediawiki_db_user: "mediawiki"
mediawiki_db_password: "mediawiki"
```

- Les dépendances se font dans le répertoire `meta` :

`roles/mediawiki/mariadb/meta/main.yml`

```
dependencies:
  - role: "mediawiki/common"
```

- Installation de la base pour MediaWiki et définition des privilèges dans

`roles/mediawiki/mariadb/tasks/main.yml`

```
- name: "mediawiki database"
  mysql_db:
    name: "{{mediawiki_db_name}}"
    state: present

- name: "mediawiki user+privileges"
  mysql_user:
    name:      "{{mediawiki_db_user}}"
    password:  "{{mediawiki_db_password}}"
    priv:      "{{mediawiki_db_name}}.*:ALL"
    host:      "{{item}}"
    state:     present
  with_items: "{{groups.apache}}"
```

- Le fichier de configuration `install-mediawiki.yml`

```
- name: "MediaWiki db configuration"
  hosts: mysql
  gather_facts: no
  roles:
    - role: "mediawiki/mariadb"
```

- Résultat de l'exécution qui montre l'application du `playbook`

```
$ ansible-playbook -i inventaire install-mediawiki.yml -b -K
```

```
PLAY [MediaWiki db configuration] *****
TASK [mediawiki/mariadb : mediawiki database] *****
changed: [mysql1]

TASK [mediawiki/mariadb : mediawiki user+privileges] *****
changed: [mysql1]

PLAY RECAP *****
mysql1                : ok=2    changed=2    unreachable=0    failed=0
```

Configuration de MediaWiki

- Ajout des variables nécessaires à la configuration dans le fichier

`roles/mediawiki/common/defaults/main.yml`

[...]

default mediawiki administrator

`mediawiki_admin_user: "admin"`

`mediawiki_admin_password: "@adm1n"`

`mediawiki_name: "mediawiki"`

`mediawiki_title: "ELS"`

`mediawiki_directory: "/var/www/html/{{mediawiki_name}}"`

`mediawiki_maintenance_directory: "{{mediawiki_directory}}/maintenance"`

default database host: first machine from group mysql

`mediawiki_db_host: "{{groups.mysql.0}}"`

`mediawiki_archive_url: "https://releases.wikimedia.org/mediawiki/1.31/"`

`mediawiki-1.31.0.tar.gz"`

- Ajoute la dépendance avec le rôle common dans le fichier
`roles/mediawiki/configuration/meta/main.yml`

```
dependencies:  
  - role: "mediawiki/common"
```

- Le fichier de configuration

```
roles/mediawiki/configuration/tasks/main.yml
```

```
- name: "mediawiki directory"
  file:
    path: "{{mediawiki_directory}}"
    owner: "apache"
    group: "apache"
    state: directory

- name: "uncompress mediawiki archive"
  unarchive:
    src: "{{mediawiki_archive_url}}"
    dest: "{{mediawiki_directory}}"
    owner: "apache"
    group: "apache"
    # unpack an archive which already exists on the target
    remote_src: yes
    # remove mediawiki-1.xx.x/ from path
    extra_opts: --transform=s/mediawiki-[0-9\\.]*\\///
```

- Suite du fichier `roles/mediawiki/configuration/tasks/main.yml`

```
- name: "mediawiki configuration"
  become: yes
  become_user: "apache"
  args:
    creates: "{{mediawiki_directory}}/LocalSettings.php"
    chdir: "{{mediawiki_maintenance_directory}}"
  command:
    php install.php --scriptpath /{{mediawiki_name}}
    --dbname mediawiki --lang fr
    --dbuser {{mediawiki_db_user}}
    --dbpass {{mediawiki_db_password}}
    --pass {{mediawiki_admin_password}}
    --dbserver {{mediawiki_db_host}}
    {{mediawiki_title}} {{mediawiki_admin_user}}
```

- Le `playbook` : `install-mediawiki.yml` qui créer la base de données et configure Mediawiki

```
- name: "MediaWiki db configuration"
  hosts: mysql
  gather_facts: no
  tags: [ "mariadb", "mysql" ]
  roles:
    - role: "mediawiki/mariadb"

- name: "MediaWiki apache configuration"
  hosts: apache
  tags: "apache"
  gather_facts: no
  roles:
    - role: "mediawiki/configuration"
```

- Exécution du `playbook` limitée au tag `apache`

```
$ ansible-playbook -i inventaire install-mediawiki.yml -b -K --tags apache
```

```
PLAY [MediaWiki db configuration] *****

PLAY [MediaWiki apache configuration] *****

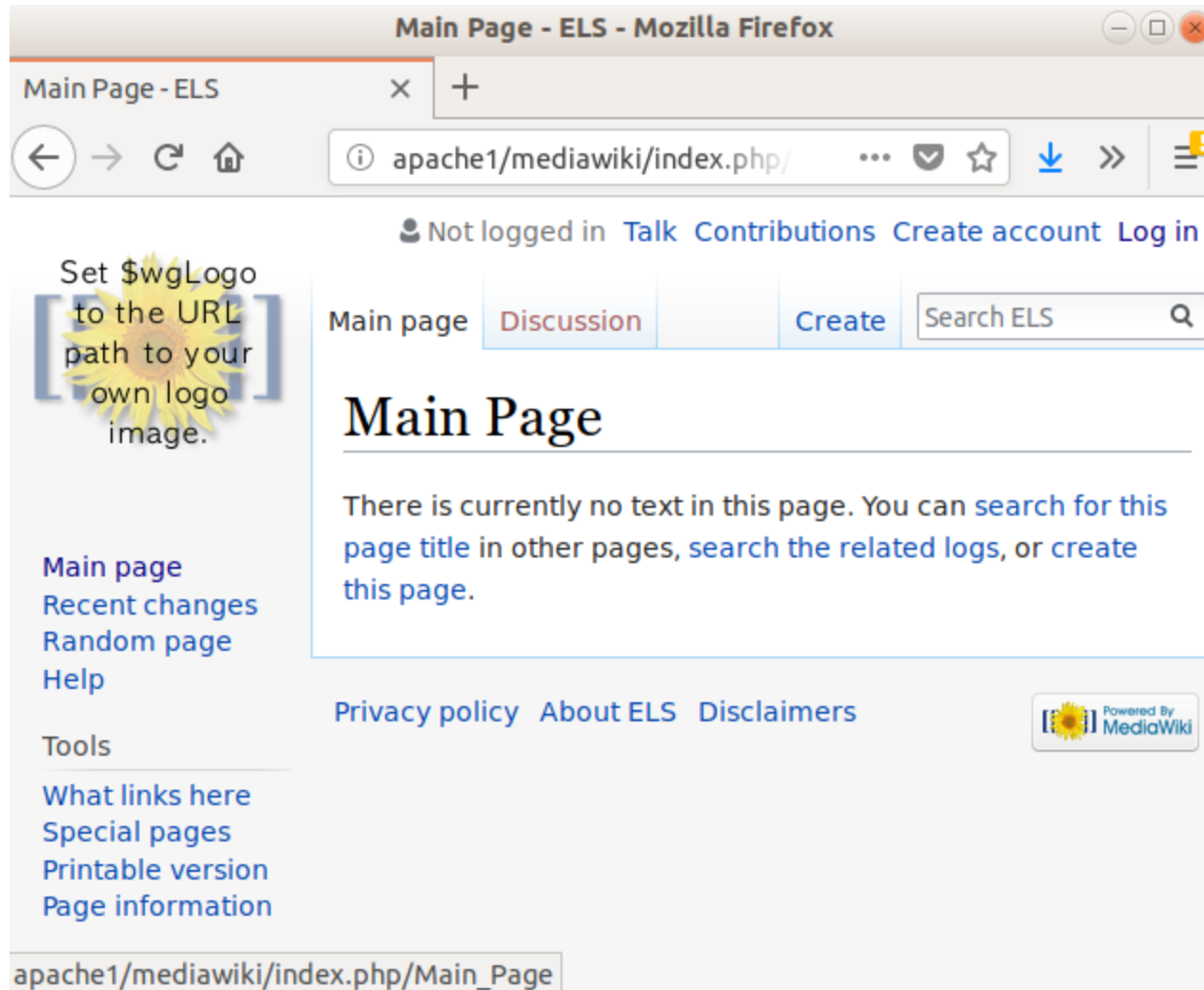
TASK [mediawiki/configuration : mediawiki directory] *****
ok: [apache1]

TASK [mediawiki/configuration : uncompress mediawiki archive] *****
ok: [apache1]

TASK [mediawiki/configuration : mediawiki configuration] *****
changed: [apache1]

PLAY RECAP *****
apache1                : ok=3    changed=1    unreachable=0    failed=0
```

- Installation faite



Scalabilité

Installation d'un nouveau serveur Apache sur la base des configurations existantes

- Ajout d'un serveur `apache2` dans `inventaire`

```
[apache]
apache[1:2]

[apache:vars]
php_install=yes

[mysql]
mysql
```

- Réutilisation du `playbook` `install-apache.yml` en limitant l'exécution au serveur `apache2`

```
$ ansible-playbook -i inventaire install-apache.yml -b -K -l apache2
```

```

PLAY [Installation apache] *****

TASK [Gathering Facts] *****
ok: [apache2]

TASK [apache : apache installation] *****
ok: [apache2]

TASK [apache : apache service activation]
ok: [apache2]

TASK [apache : epel activation] *****
changed: [apache2]

TASK [apache : remi repo activation] *****
changed: [apache2]

TASK [apache : install php70 packages] *****
changed: [apache2]

RUNNING HANDLER [apache : apache restart] *****
changed: [apache2]

PLAY RECAP *****
apache2                : ok=7    changed=4    unreachable=0    failed=0

```


- Relance le `playbook install-mediawiki.yml` en limitant l'exécution aux serveurs `apache2` et `mysql1` pour notamment ajouter les accès à la base pour le nouveau serveur (ajout des privilèges dans MariaDB).

```
$ ansible-playbook -i inventaire install-mediawiki.yml -b -K -l apache2,mysql1
```

- Le résultat avec application des privilèges pour `apache2` uniquement

```
PLAY [MediaWiki db configuration] *****

TASK [mediawiki/mariadb : mediawiki database] *****
ok: [mysql1]

TASK [mediawiki/mariadb : mediawiki user+privileges] *****
ok: [mysql1] => (item=apache1)
changed: [mysql1] => (item=apache2)

PLAY [MediaWiki apache configuration] *****

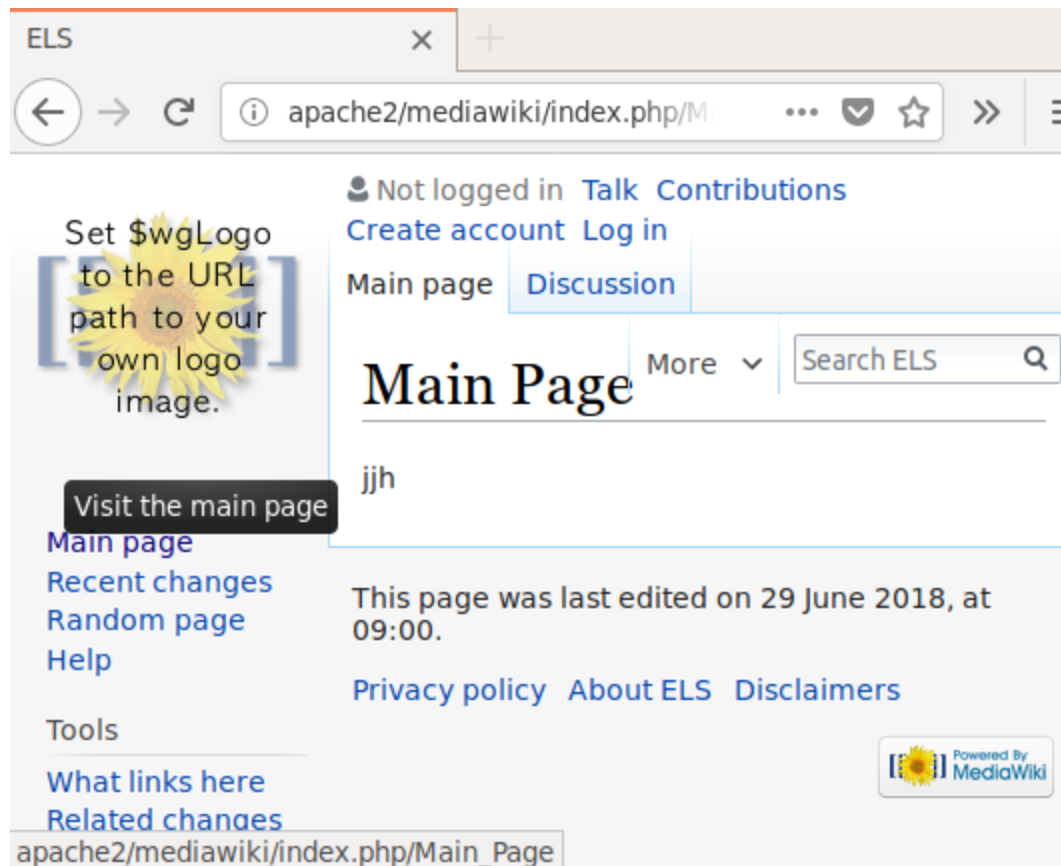
TASK [mediawiki/configuration : mediawiki directory] *****
ok: [apache2]

TASK [mediawiki/configuration : uncompress mediawiki archive] *****
ok: [apache2]

TASK [mediawiki/configuration : mediawiki configuration] *****
changed: [apache2]

[...]
```

- Le deuxième serveur Apache est opérationnel



Lancement d'une opération de manière séquentielle

Il est parfois prudent de ne pas lancer deux actions en même temps sur plusieurs serveurs (par exemple, modification du schéma de la base données)

Pour cela, les options suivantes peuvent être utilisées : `run_once` ,
`delegate_to` et `with_items`

Les actions seront ainsi exécutées une seule fois et exécutées les unes à la suite des autres.

- Appliqué au `playbook`

`roles/mediawiki/configuration/tasks/main.yml`

```
[...]
- name: "mediawiki configuration"
  become: yes
  become_user: "apache"
  args:
    creates: "{{mediawiki_directory}}/LocalSettings.php"
    chdir: "{{mediawiki_maintenance_directory}}"
  command:
    php install.php --scriptpath /{{mediawiki_name}}
    --dbname mediawiki --lang fr
    --dbuser {{mediawiki_db_user}}
    --dbpass {{mediawiki_db_password}}
    --pass {{mediawiki_admin_password}}
    --dbserver {{mediawiki_db_host}}
    {{mediawiki_title}} {{mediawiki_admin_user}}
# Execute une seule fois
  run_once: yes
# Passe la main au serveur suivant
  delegate_to: "{{item}}"
  with_items: "{{groups.apache}}"
```

Ansible et le Cloud

Exemple avec les services de Amazon - AWS



- Création d'une instance EC2
- Connexion SSH à une instance
- Suppression d'une instance EC2

Il y a 98 modules AWS dans Ansible : [Modules Amazon](#)

Il faut avant tout configurer les accès à l'API `Amazon`, pour cela, il faut créer une clés dans la console `Amazon`

- Une fois la clé créée, exporter les variables `AWS_ACCESS_KEY_ID` et `AWS_SECRET_ACCESS_KEY` dans l'environnement courant.

```
$ export AWS_ACCESS_KEY_ID=XXXXXXXXXXXXXXXXX  
$ export AWS_SECRET_ACCESS_KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

- Création d'une instance EC2 tagué `instance1`

```
- name: "Create ec2 instance"
hosts: localhost
gather_facts: no
tasks:
# NAT
  - name: "Configure SSH access"
    ec2_group:
      name: "test"
      description: "SSH access to test server"
      region: us-east-2
      rules:
        - proto: tcp
          from_port: 22
          to_port: 22
          cidr_ip: 0.0.0.0/0
# Charge la clé SSH
  - name: "Upload my public key"
    ec2_key:
      name: "mykey"
      region: "us-east-2"
      key_material: "{{ item }}"
      with_file: "~/.ssh/id_rsa_ansible.pub"
```



```
# Crée l'instance
```

```
- name: "Provision a test instance"
  ec2:
    key_name: "mykey"
    group: "test"
    instance_type: "t2.micro"
    image: "ami-cfdafaaa"
    wait: true
    region: "us-east-2"
    instance_tags:
      type: "instance1"
    count_tag:
      type: "instance1"
    exact_count: 1
```

`wait: true` on attend que l'instance soit créé avant de rendre la main.

`count_tag` et `exact_count` vont rendre *idempotent* la creation de l'instance. Si une instance "instance1" existe alors elle ne sera pas créée.

- Lancement de l'installation de l'instance

```
$ ansible-playbook aws-creation.yml
```

```
PLAY [Create ec2 instance] *****
```

```
TASK [Configure SSH access] *****
```

```
ok: [localhost]
```

```
TASK [Upload my public key] *****
```

```
ok: [localhost] => (item=ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAA[...])
```

```
TASK [Provision a test instance] *****
```

```
changed: [localhost]
```

```
PLAY RECAP ***
```

```
localhost                : ok=3    changed=1    unreachable=0    failed=0
```

Vue de la console AWS

The screenshot displays the AWS Management Console interface. At the top, the browser address bar shows the URL `https://us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2`. The console header includes the AWS logo, navigation tabs for 'Services' and 'Groupes de ressources', and user information for 'alex+aws' in the 'Ohio' region. A tooltip 'Page d'accueil de la console' points to the 'Lancer une instance' button. Below the header, a search bar is labeled 'Filtrer par balises et attributs ou rechercher par mot clé'. A table lists the EC2 instances:

	Name	ID d'instance	Type d'instance	Zone de dispon	État de l'instai	Contrôles des	Statut des alar	DNS publi
<input type="checkbox"/>		i-0ed30e845f8abb...	t2.micro	us-east-2c	● running	✓ 2/2 contrôl...	Aucun	ec2-18-21

Accès réseau aux instances

Inventaire dynamique pour communiquer avec toutes les instances

<https://github.com/ansible/ansible/tree/devel/contrib/inventory>

En utilisant les deux scripts `ec2.py` et `ec2.ini`, pour générer un inventaire dynamique et l'utiliser pour communiquer avec les instances.

- Lancement du script `ec2.py`

```
$ ./ec2.py
```

- La sortie d'inventaire générée par le script

```
{
  "_meta": {
    "hostvars": {
      "18.218.209.216": {
        "ansible_host": "18.218.209.216",
        "ec2__in_monitoring_element": false,
        [...]
      },
      [...]
    },
    "ami_cfdafaaa": [
      "18.218.35.171",
      "18.218.209.216"
    ],
    "ec2": [
      "18.218.35.171",
      "18.218.209.216"
    ],
    [...]
  }
}
```

- Test de `ping` sur l'ensemble des instances en utilisant l'inventaire dynamique.

```
$ ansible -i ./ec2.py -m ping all
```

```
18.219.193.54 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
18.188.65.52 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
18.191.152.34 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```

- Connexion **ssh** à une des instances

```
$ ssh ec2-user@18.191.152.34
```

```
Last login: Fri Jun 29 13:35:09 2018 from 89.2.106.133
```

```
[ec2-user@ip-172-31-42-94 ~]$ df -h
```

Sys. de fichiers	Taille	Utilisé	Dispo	Uti%	Monté sur
/dev/xvda2	10G	878M	9,2G	9%	/
devtmpfs	476M	0	476M	0%	/dev
tmpfs	496M	0	496M	0%	/dev/shm
tmpfs	496M	13M	483M	3%	/run
tmpfs	496M	0	496M	0%	/sys/fs/cgroup
tmpfs	100M	0	100M	0%	/run/user/1000

Suppression d'une instance

- le fichier de configuration `aws-delete.yml` , suppression des instances taguées en `instance1`

```
- name: "Remove ec2 instance"
  hosts: localhost
  gather_facts: no
  tasks:
    - name: "Remove a test instance"
      ec2:
        region: "us-east-2"
        instance_type: "t2.micro"
        image: "ami-cfdafaaa"
        wait: true
        instance_tags:
          type: "instance1"
        count_tag:
          type: "instance1"
        exact_count: 0
        register: ec2
```


Pour aller plus loin...

