

Ansistrano

Table des matières

1. Introduction	2
2. Module 1 : Prise en main de la plateforme	4
2.1. Exercice 1.1 : Déploiement de la plateforme	4
3. Module 2 : Déployer le serveur Web	5
3.1. Exercice 2.1 : Utiliser le rôle <code>geerlingguy.apache</code> pour configurer le serveur	5
4. Module 3 : Déployer le logiciel	7
4.1. Exercice 3.1 : Installer le rôle <code>ansistrano-deploy</code>	7
4.2. Exercice 3.2 : Utiliser le rôle <code>ansistrano-deploy</code> pour déployer le logiciel	7
4.3. Exercice 3.3 : Visualiser le résultat dans un navigateur	7
4.4. Exercice 3.4 : Vérification sur le serveur	8
5. Module 4 : Ansistrano	10
5.1. Exercice 4.1 : Limiter le nombre de releases	10
5.2. Exercice 4.2 : Utiliser des <code>shared_paths</code> et <code>shared_files</code>	11
5.3. Exercice 4.3 : Utiliser un sous répertoire du dépôt pour le déploiement	13
6. Module 5 : La gestion des branches ou des tags git	17
6.1. Exercice 5.1 : Déployer une branche	17
6.2. Exercice 5.2 : Déployer un tag	18
7. Module 6 : Actions entre les étapes de déploiement	20
7.1. Exercice 6.1 : Envoyer un mail en début de MEP	20
7.2. Exercice 6.2 : Redémarrer apache en fin de MEP	22

Objectifs

- Mettre en oeuvre Ansistrano ;
- Configurer Ansistrano ;
- Utiliser des dossiers et fichiers partagés entre versions déployées ;
- Déployer différentes versions d'un site depuis git ;
- Réagir entre les étapes de déploiement.

Chapitre 1. Introduction

Ansistrano est un rôle Ansible pour déployer facilement des applications PHP, Python, etc. Il se base sur le fonctionnement de [Capistrano](#)

Ansistrano nécessite pour fonctionner :

- Ansible sur la machine de déploiement,
- **rsync** ou **git** sur la machine cliente.

Il peut télécharger le code source depuis **rsync**, **git**, **scp**, **http**, **S3**, ...



Dans le cadre de notre exemple de déploiement, nous allons utiliser le protocole **git**.

Ansistrano déploie les applications en suivant ces 5 étapes :

- **Setup** : création de la structure de répertoire pour accueillir les releases
- **Update Code** : téléchargement de la nouvelle release sur les cibles
- **Symlink Shared** et **Symlink** : après avoir déployé la nouvelle release, le lien symbolique **current** est modifié pour pointer vers cette nouvelle release
- **Clean Up** : pour faire un peu de nettoyage (suppression des anciennes versions)

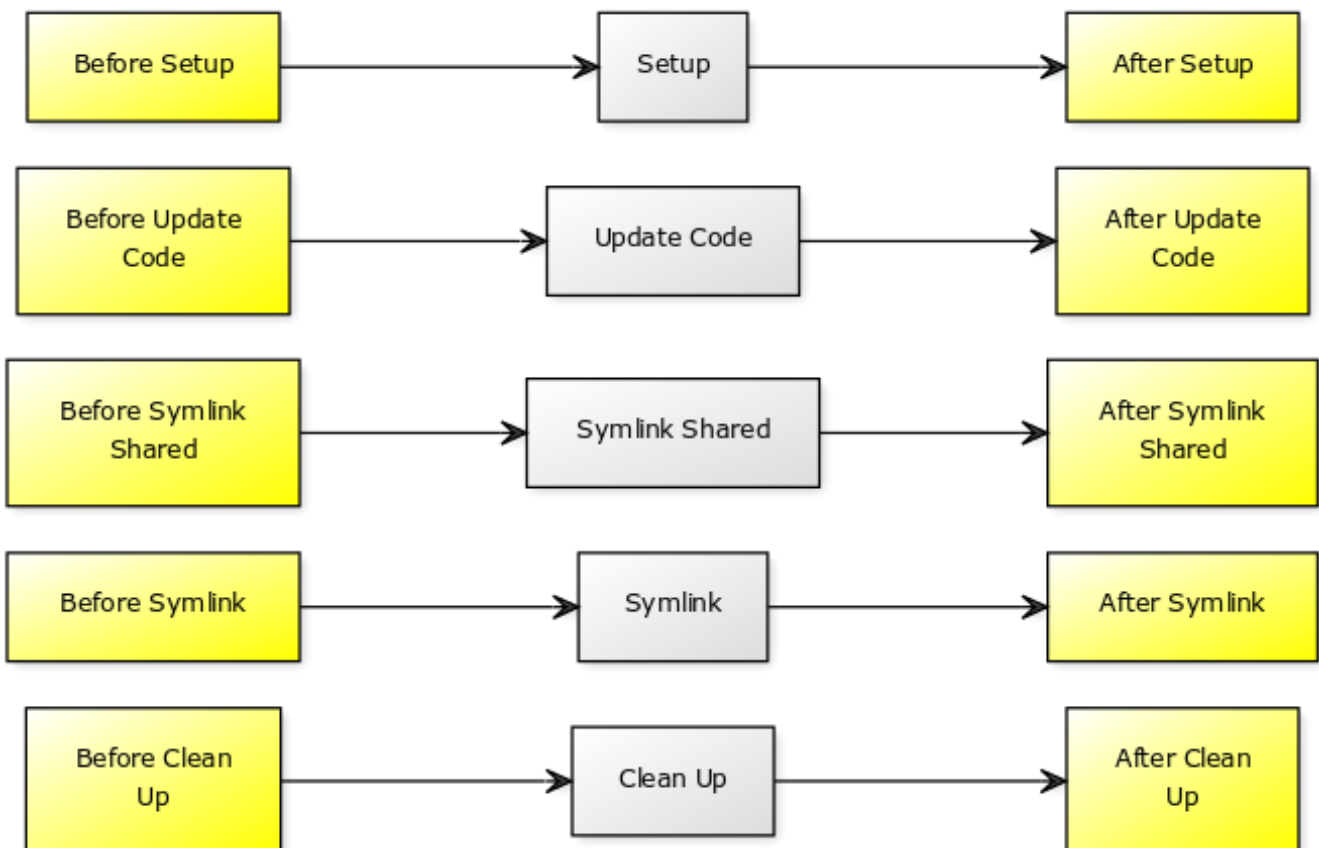


Figure 1. Etapes d'un déploiement

Le squelette d'un déploiement avec ansistrano ressemble à :

Squelette d'un déploiement avec ansistrano

```
/var/www/site/  
├── current -> ./releases/20180821070043Z  
├── releases  
│   ├── 20180821070043Z  
│   │   ├── css -> ../../shared/css/  
│   │   ├── img -> ../../shared/img/  
│   │   └── REVISION  
├── repo  
└── shared  
    ├── css/  
    └── img/
```

Vous retrouverez toute la documentation ansistrano sur son [dépôt Github](#).

Chapitre 2. Module 1 : Prise en main de la plateforme

2.1. Exercice 1.1 : Déploiement de la plateforme

Vous allez travailler sur 2 VM :

- La VM de gestion :
 - Vous devrez installer ansible et déployer le rôle **ansistrano-deploy**
- La VM cliente :
 - Cette VM n'a aucun logiciel spécifique.
 - Vous devrez installer Apache et déployer le site du client

Chapitre 3. Module 2 : Déployer le serveur Web

3.1. Exercice 2.1 : Utiliser le rôle `geerlingguy.apache` pour configurer le serveur

Pour gérer notre serveur web, nous allons utiliser le rôle `geerlingguy.apache` qu'il faut installer sur le serveur :

```
[ansible@srv] $ ansible-galaxy install geerlingguy.apache
```

Une fois le rôle installé, nous allons pouvoir créer la première partie de notre playbook, qui va :

- Installer Apache,
- Créer un dossier cible pour notre `vhost`,
- Créer un `vhost` par défaut,
- Démarrer ou redémarrer Apache.

Considérations techniques :

- Nous allons déployer notre site dans le dossier `/var/www/site/`.
- Comme nous le verrons plus loin, `ansistrano` va créer un lien symbolique `current` vers le dossier de la release en cours.
- Le code source à déployer contient un dossier `html` sur lequel le `vhost` devra pointer. Sa `DirectoryIndex` est `index.htm`.
- Le déploiement se faisant par `git`, le paquet sera installé.



La cible de notre `vhost` sera donc : `/var/www/site/current/html`.

Playbook de configuration du serveur playbook-config-serveur.yml

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    apache_global_vhost_settings: |
      DirectoryIndex index.php index.htm
    apache_vhosts:
      - servername: "website"
        documentroot: "{{ dest }}current/html"

  tasks:

    - name: create directory for website
      file:
        path: /var/www/site/
        state: directory
        mode: 0755

    - name: install git
      package:
        name: git
        state: latest

  roles:
    - { role: geerlingguy.apache }
```

Le playbook peut être appliqué au serveur :

```
[ansiblesrv] $ ansible-playbook -i hosts playbook-config-serveur.yml
```

Notez l'exécution des tâches suivantes :

```
TASK [geerlingguy.apache : Ensure Apache is installed on RHEL.] *****
TASK [geerlingguy.apache : Configure Apache.] *****
TASK [geerlingguy.apache : Add apache vhosts configuration.] *****
TASK [geerlingguy.apache : Ensure Apache has selected state and enabled on boot.] ***
RUNNING HANDLER [geerlingguy.apache : restart apache] *****
```

Le rôle **geerlingguy.apache** nous facilite grandement la tâche en s'occupant de l'installation et la configuration d'Apache.

Chapitre 4. Module 3 : Déployer le logiciel

Notre serveur étant maintenant configuré, nous allons pouvoir déployer l'application.

Pour cela, nous allons utiliser le rôle `ansistrano.deploy` dans un second playbook dédié au déploiement applicatif (pour plus de lisibilité).

4.1. Exercice 3.1 : Installer le rôle `ansistrano-deploy`

```
[ansiblesrv] $ ansible-galaxy install ansistrano.deploy
```

4.2. Exercice 3.2 : Utiliser le rôle `ansistrano-deploy` pour déployer le logiciel

Les sources du logiciel se trouvent dans le dépôt :

- sur framagit <https://framagit.org/alemorvan/demo-ansible.git> accessible depuis internet.

Nous allons créer un playbook `playbook-deploy.yml` pour gérer notre déploiement :

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"

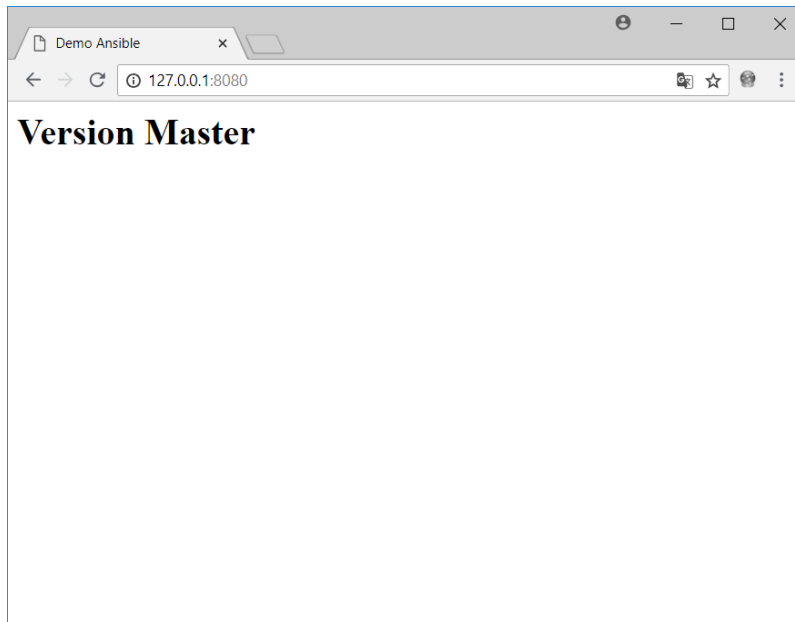
  roles:
    - { role: ansistrano.deploy }
```

```
[ansiblesrv] $ ansible-playbook -i hosts playbook-deploy.yml
```

4.3. Exercice 3.3 : Visualiser le résultat dans un navigateur

Une redirection du port `http` a été mis en place pour accéder depuis l'extérieur à votre serveur.

- Ouvrir un navigateur web vers l'url <http://@IPPUBLIQUE:PORTREDIR/> :



Et voir apparaître le déploiement fonctionnel de la branche **master** de notre dépôt.

4.4. Exercice 3.4 : Vérification sur le serveur

- Se connecter en ssh sur le serveur client :

```
[ansiblesrv] $ ssh ansible@192.168.10.11
```

- Faire un **tree** sur le repertoire **/var/www/site/** :

```
[ansiblecli] $ tree /var/www/site/
/var/www/site/
├── current -> ./releases/20180821070043Z
├── releases
│   ├── 20180821070043Z
│   │   ├── html
│   │   │   └── index.htm
│   │   └── REVISION
│   └── repo
│       ├── html
│       │   └── index.htm
│       └── shared
```

Notez :

- le lien symbolique **current** vers la release **./releases/20180821070043Z**
- la présence d'un dossier **shared**
- la présence du dépôt git dans **./repo/**

- Depuis le serveur ansible, relancer **3** fois le déploiement, puis vérifier sur le client.

```
[ansiblecli] $ tree /var/www/site/
/var/www/site/
├── current -> ./releases/20180821112348Z
├── releases
│   ├── 20180821070043Z
│   │   ├── html
│   │   │   └── index.htm
│   │   └── REVISION
│   ├── 20180821112047Z
│   │   ├── html
│   │   │   └── index.htm
│   │   └── REVISION
│   └── 20180821112100Z
│       ├── html
│       │   └── index.htm
│       └── REVISION
│   └── 20180821112348Z
│       ├── html
│       │   └── index.htm
│       └── REVISION
├── repo
│   └── html
│       └── index.htm
└── shared
```

Notez :

- **ansistrano** a conservé les 4 dernières releases,
- le lien **current** pointe maintenant vers la dernière release exécutée

Chapitre 5. Module 4 : Ansistrano

5.1. Exercice 4.1 : Limiter le nombre de releases

La variable `ansistrano_keep_releases` permet de spécifier le nombre de releases à conserver.

- En utilisant la variable `ansistrano_keep_releases`, ne conserver que 3 releases du projet. Vérifier.

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"
    ansistrano_keep_releases: 3

  roles:
    - { role: ansistrano.deploy }
```

```
---
[ansiblesrv] $ ansible-playbook -i hosts playbook-deploy.yml
```

Sur le client :

```
[ansiblecli] $ tree /var/www/site/
/var/www/site/
├── current -> ./releases/20180821113223Z
├── releases
│   ├── 20180821112100Z
│   │   ├── html
│   │   │   └── index.htm
│   │   └── REVISION
│   ├── 20180821112348Z
│   │   ├── html
│   │   │   └── index.htm
│   │   └── REVISION
│   └── 20180821113223Z
│       ├── html
│       │   └── index.htm
│       └── REVISION
├── repo
│   └── html
│       └── index.htm
└── shared
```

5.2. Exercice 4.2 : Utiliser des `shared_paths` et `shared_files`

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"
    ansistrano_keep_releases: 3
    ansistrano_shared_paths:
      - "img"
      - "css"
    ansistrano_shared_files:
      - "logs"

  roles:
    - { role: ansistrano.deploy }
```

Sur le client, créer le fichier `logs` dans le répertoire `shared` :

```
sudo touch /var/www/site/shared/logs
```

Puis lancer l'exécution du job :

```
TASK [ansistrano.deploy : ANSISTRANO | Ensure shared paths targets are absent]
*****
ok: [192.168.10.11] => (item=img)
ok: [192.168.10.11] => (item=css)
ok: [192.168.10.11] => (item=logs/log)

TASK [ansistrano.deploy : ANSISTRANO | Create softlinks for shared paths and files]
*****
changed: [192.168.10.11] => (item=img)
changed: [192.168.10.11] => (item=css)
changed: [192.168.10.11] => (item=logs)
```

Sur le client :

```
[ansiblecli] $ tree -F /var/www/site/
/var/www/site/
├── current -> ./releases/20180821120131Z/
├── releases
│   ├── 20180821112348Z/
│   │   ├── html/
│   │   │   └── index.htm
│   │   └── REVISION
│   └── 20180821113223Z/
│       ├── html/
│       │   └── index.htm
│       └── REVISION
│   └── 20180821120131Z/
│       ├── css -> ../../shared/css/
│       ├── html
│       │   └── index.htm
│       ├── img -> ../../shared/img/
│       ├── logs -> ../../shared/logs
│       └── REVISION
├── repo/
│   └── html
│       └── index.htm
└── shared/
    ├── css/
    ├── img/
    └── logs
```

Notez que la dernière release contient :

- Un répertoire **css**, un répertoire **img**, et un fichier **logs**
- Des liens symboliques :
 - du dossier `/var/www/site/releases/css/` vers le dossier `../../shared/css/`.
 - du dossier `/var/www/site/releases/img/` vers le dossier `../../shared/img/`.
 - du fichier `/var/www/site/releases/logs/` vers le fichier `../../shared/logs`.

Dès lors, les fichiers contenus dans ces 2 dossiers et le fichier **logs** sont toujours accessibles via les chemins suivants :

- `/var/www/site/current/css/`,
- `/var/www/site/current/img/`,
- `/var/www/site/current/logs`,

mais surtout ils seront conservés d'une release à l'autre.

5.3. Exercice 4.3 : Utiliser un sous répertoire du dépôt pour le déploiement

Dans notre cas, le dépôt contient un dossier **html**, qui contient les fichiers du site.

- Pour éviter ce niveau supplémentaire de répertoire, utiliser la variable `ansistrano_git_repo_tree` en précisant le path du sous répertoire à utiliser. Ne pas oublier de modifier la configuration d'apache pour prendre en compte ce changement !

Playbook de configuration du serveur playbook-config-serveur.yml

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    apache_global_vhost_settings: |
      DirectoryIndex index.php index.htm
    apache_vhosts:
      - servername: "website"
        documentroot: "{{ dest }}current/" ①

  tasks:

    - name: create directory for website
      file:
        path: /var/www/site/
        state: directory
        mode: 0755

    - name: install git
      package:
        name: git
        state: latest

  roles:
    - { role: geerlingguy.apache }
```

① [Modifier cette ligne](#)

Playbook de déploiement playbook-deploy.yml

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"
    ansistrano_keep_releases: 3
    ansistrano_shared_paths:
      - "img"
      - "css"
    ansistrano_shared_files:
      - "log"
    ansistrano_git_repo_tree: 'html' ①

  roles:
    - { role: ansistrano.deploy }
```

① Modifier cette ligne

- Vérifier sur le client :

```
[ansiblecli] $ tree -F /var/www/site/
/var/www/site/
├── current -> ./releases/20180821120131Z/
├── releases
│   ├── 20180821113223Z/
│   │   ├── html/
│   │   │   └── index.htm
│   │   └── REVISION
│   ├── 20180821120131Z/
│   │   ├── css -> ../../shared/css/
│   │   ├── html
│   │   │   └── index.htm
│   │   ├── img -> ../../shared/img/
│   │   ├── logs -> ../../shared/logs
│   │   └── REVISION
│   └── 20180821130124Z/
│       ├── css -> ../../shared/css/
│       ├── img -> ../../shared/img/
│       ├── index.htm ①
│       ├── logs -> ../../shared/logs
│       └── REVISION
├── repo/
│   └── html
│       └── index.htm
└── shared/
    ├── css/
    ├── img/
    └── logs
```

① Notez l'absence du dossier **html**

Chapitre 6. Module 5 : La gestion des branches ou des tags git

La variable `ansistrano_git_branch` permet de préciser une `branch` ou un `tag` à déployer.

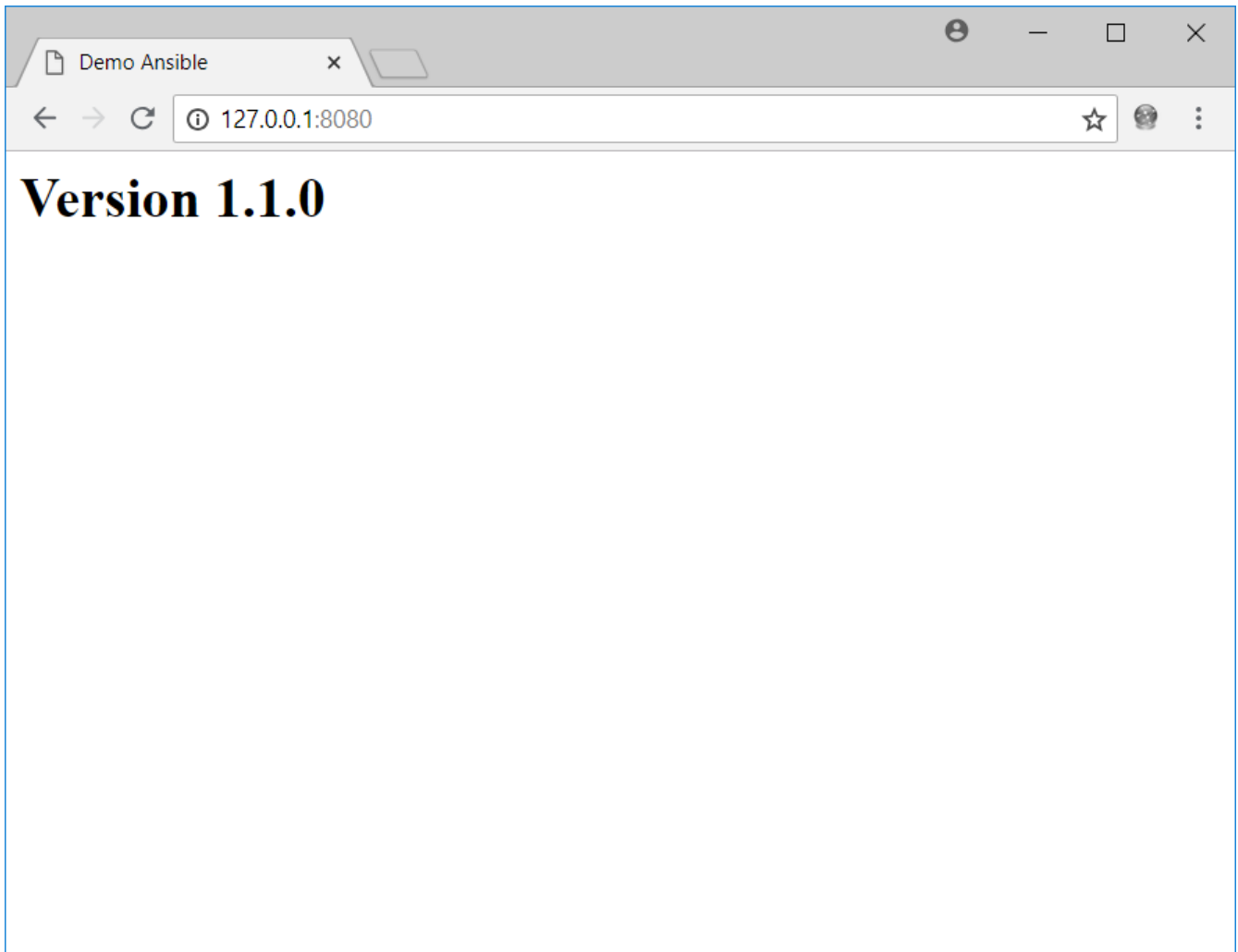
6.1. Exercice 5.1 : Déployer une branche

- Déployer la branche `releases/v1.1.0` :

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"
    ansistrano_keep_releases: 3
    ansistrano_shared_paths:
      - "img"
      - "css"
    ansistrano_shared_files:
      - "log"
    ansistrano_git_repo_tree: 'html'
    ansistrano_git_branch: 'releases/v1.1.0'

  roles:
    - { role: ansistrano.deploy }
```

Vous pouvez vous amuser, durant le déploiement, à rafraichir votre navigateur, pour voir en 'live' le changement s'effectuer.



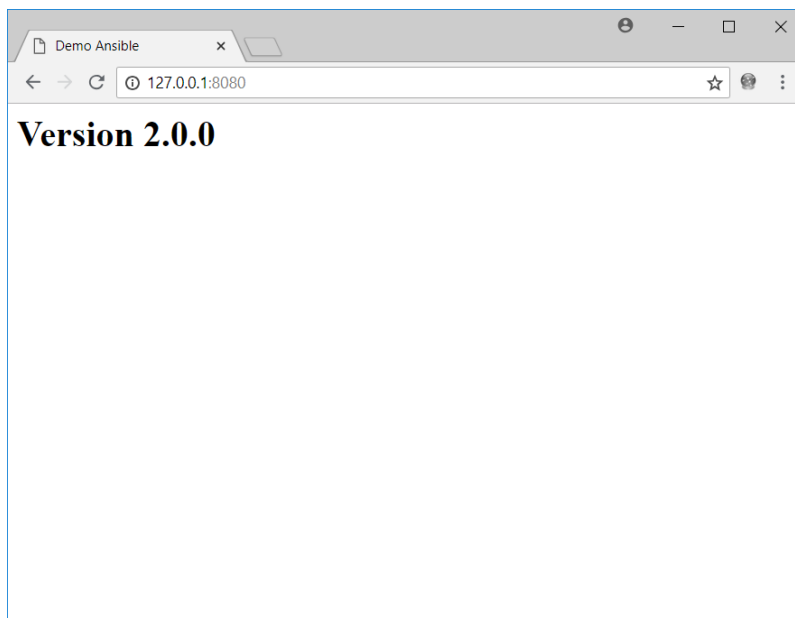
6.2. Exercice 5.2 : Déployer un tag

- Déployer le tag **v2.0.0** :

```
---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"
    ansistrano_keep_releases: 3
    ansistrano_shared_paths:
      - "img"
      - "css"
    ansistrano_shared_files:
      - "log"
    ansistrano_git_repo_tree: 'html'
    ansistrano_git_branch: 'v2.0.0'

  roles:
    - { role: ansistrano.deploy }
```

Vous pouvez vous amuser, durant le déploiement, à rafraichir votre navigateur, pour voir en 'live' le changement s'effectuer.

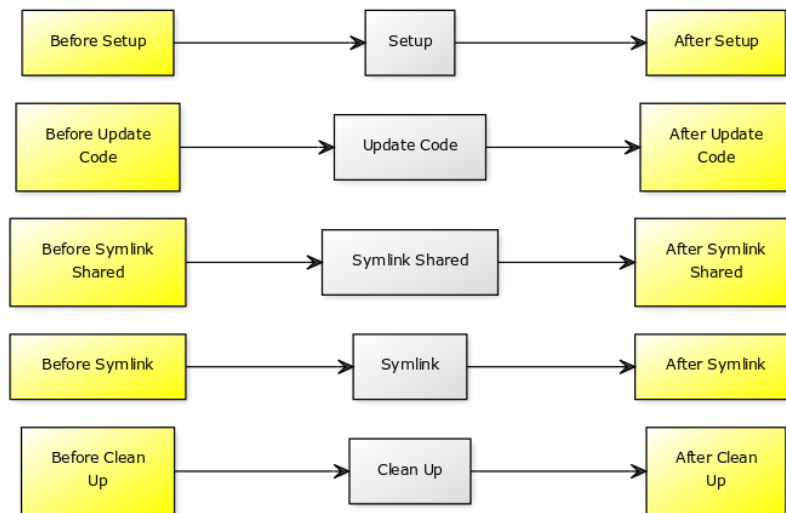


Chapitre 7. Module 6 : Actions entre les étapes de déploiement

Un déploiement avec Ansistrano respecte les étapes suivantes :

- **Setup**
- **Update Code**
- **Symlink Shared**
- **Symlink**
- **Clean Up**

Il est possible d'intervenir avant et après chacune de ses étapes.



Un playbook peut être inclus par les variables prévues à cet effet :

- `ansistrano_before_<task>_tasks_file`
- ou `ansistrano_after_<task>_tasks_file`

7.1. Exercice 6.1 : Envoyer un mail en début de MEP

```

---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"
    ansistrano_keep_releases: 3
    ansistrano_shared_paths:
      - "img"
      - "css"
    ansistrano_shared_files:
      - "logs"
    ansistrano_git_repo_tree: 'html'
    ansistrano_git_branch: 'v2.0.0'
    ansistrano_before_setup_tasks_file: "{{ playbook_dir }}/deploy/before-setup-
tasks.yml"

  roles:
    - { role: ansistrano.deploy }

```

Le fichier deploy/before-setup-tasks.yml

```

---
- name: Envoyer un mail
  mail:
    subject: Debut de MEP sur {{ ansible_hostname }}.
    delegate_to: localhost

```

```

TASK [ansistrano.deploy : include]
*****
included: /home/ansible/deploy/before-setup-tasks.yml for 192.168.10.11

TASK [ansistrano.deploy : Envoyer un mail]
*****
ok: [192.168.10.11 -> localhost]

```

```

[root] # mailx
Heirloom Mail version 12.5 7/5/10. Type ? for help.
"/var/spool/mail/root": 1 message 1 new
>N 1 root@localhost.local Tue Aug 21 14:41 28/946 "Debut de MEP sur localhost."

```

7.2. Exercice 6.2 : Redémarrer apache en fin de MEP

```

---
- hosts: ansiblecli
  become: yes
  vars:
    dest: "/var/www/site/"
    ansistrano_deploy_via: "git"
    ansistrano_git_repo: https://framagit.org/alemorvan/demo-ansible.git
    ansistrano_deploy_to: "{{ dest }}"
    ansistrano_keep_releases: 3
    ansistrano_shared_paths:
      - "img"
      - "css"
    ansistrano_shared_files:
      - "logs"
    ansistrano_git_repo_tree: 'html'
    ansistrano_git_branch: 'v2.0.0'
    ansistrano_before_setup_tasks_file: "{{ playbook_dir }}/deploy/before-setup-
tasks.yml"
    ansistrano_after_symlink_tasks_file: "{{ playbook_dir }}/deploy/after-symlink-
tasks.yml"

  roles:
    - { role: ansistrano.deploy }

```

Le fichier `deploy/after-symlink-tasks.yml`

```

---
- name: restart apache
  service:
    name: httpd
    state: restarted

```

```
TASK [ansistrano.deploy : include]
```

```
*****
included: /home/ansible/deploy/after-symlink-tasks.yml for 192.168.10.11
```

```
TASK [ansistrano.deploy : restart apache]
```

```
*****
changed: [192.168.10.11]
```