Mini-projet Final

Repository

https://github.com/infradev4/roleWordpress.git

https://github.com/infradev4/roleDocker.git

Créez un rôle permettant de déployer Wordpress

- Vous avez reçu la demande d'une autre équipe qui souhaiterait utiliser un de vos playbook de déploiement de WordPress, mais sous forme de rôle car sous cette forme ils pourront mieux variabiliser et adapter à leur situation
- Leur objectif est que votre rôle possède un playbook tests afin de leur permettre de tester rapidement votre rôle et ainsi l'intégrer à leur process de déploiement, exactement comme le rôle wordpress
- Utilisez le rôle docker précédemment crée afin de créer un nouveau rôle qui permettra de déployer l'application wordpress à l'aide de deux containers docker (mysql + wordpress)
- Déployez vos différents container à l'aide du module docker_container, ces containers devront appartenir à un meme réseau « wordpress » que vous devez au préalable créer à partir du module ansible « docker-network »
- Variabilisez un maximum de paramètres de ce role afin qu'il puisse etre utilisé par différentes entreprises fonction du besoin, on devra pouvoir personnaliser lors du déploiement :
- Le nom du réseau dans lequel sera crée les container
- Le nom des container
- Le port sur lequel consommer l'application
- Le nom du volume dans lequel sera sauvegardé le BDD mysql (mettre en place la persistence)
- A la fin de votre travail, poussez votre rôle sur github et sur la galaxy ansible (wordpress_role) et envoyez nous votre rapport détaillé de mise en œuvre en PDF via l'intranet (Partage de documents)

Structure de mon projet

/home/ubuntu/roles

```
docker/
— defaults
   └─ main.yml
  - files
  - handlers
   └─ main.yml
  — meta
   └─ main.yml
  — tasks
   └─ main.yml
  templates
  - tests
    ├─ inventory
    └─ test.yml
  – vars
    └─ main.yml
wordpress/
  - defaults
    └─ main.yml
  - meta
   └─ main.yml
  - README.md
  - tasks
   └─ main.yml
  - tests
    ├─ inventory
    └─ test.yml
```

Variabilisation de mon role wordpress

vi wordpress/defaults/main.yml

```
# Docker Create a network
network_one: wordpress

# Docker wordpress settings
wordpress_container_name: wordpress
wordpress_exposed_port: 80
wordpress_container_data: wordpressData

# Docker mysql settings
mariadb_container_name: mysql
```

```
mysql_exposed_port: 3306
mysql_root_password: root123!
mysql_database: db_wordpress
mysql_user: wordpress
mysql_password: wordpress123!
mysql_container_data: mysqlData
```

Création du fichier de tâches (tasks)

```
wordpress/
├── tasks
│ └── main.yml
```

vi wordpress/tasks/main.yml

```
# tasks file for wordpress_role
- name: Create a network
 docker network:
   name: "{{ network_one }}"
- name: start mariadb containers
 docker_container:
   name: "{{ mariadb_container_name }}"
    image: "{{ mariadb_container_name }}"
   state: started
   env:
     MYSQL_ROOT_PASSWORD: "{{ mysql_root_password }}"
      MYSQL_DATABASE: "{{ mysql_database }}"
     MYSQL_USER: "{{ mysql_user }}"
     MYSQL_PASSWORD: "{{ mysql_password }}"
      - "{{ mysql_exposed_port }}:3306"
   volumes:
      - "{{ mysql_container_data }}:/var/lib/mysql"
- name: start wordpress containers
 docker_container:
   name: "{{ wordpress_container_name }}"
    image: "{{ wordpress_container_name }}"
   state: started
   env:
     WORDPRESS_DB_HOST: mysql_host
      WORDPRESS_DB_USER: "{{ mysql_user }}"
      WORDPRESS_DB_PASSWORD: "{{ mysql_password }}"
     WORDPRESS_DB_NAME: "{{ mysql_database }}"
   ports:
      - "{{ wordpress_exposed_port }}:80"
```

```
links:
    - "{{ mariadb_container_name }}:mysql_host"
    volumes:
    - "{{ wordpress_container_data }}:/var/www/html"

- name: Add a container to a network, leaving existing containers connected docker_network:
    name: "{{ network_one }}"
    connected:
    - "{{ mariadb_container_name }}"
    - "{{ wordpress_container_name }}"
    appends: yes
```

Playbook tests

```
wordpress/
|-- tests
| -- inventory
| -- test.yml
```

vi wordpress/tests/inventory

```
localhost
```

vi wordpress/tests/test.yml

```
---
- hosts: localhost
remote_user: root
become: true
vars:
   ansible_connection: local
roles:
   - wordpress
```

ansible-playbook pour le role Docker & Wordpress

```
roles/
|-- docker
|-- wordpress
```

```
├── hosts.yaml
├── docker.yaml
```

vi docker.yaml

```
---
- name:
hosts: prod
become: true
roles:
- docker
- wordpress
```

vi hosts.yaml

```
all:
 children:
   ansible:
      hosts:
        localhost:
          ansible_connection: local
          ansible_user: ubuntu
          ansible_password: ubuntu
          hostname: AnsibleMaster
    prod:
      hosts:
        worker01:
          ansible host: 172.31.14.210
          ansible_user: ubuntu
          ansible_password: ubuntu
          ansible_ssh_common_args: '-o StrictHostKeyChecking=no'
          hostname: AnsibleWorker01
        worker02:
          ansible_host: 172.31.0.185
          ansible_user: ubuntu
          ansible_password: ubuntu
          ansible_ssh_common_args: '-o StrictHostKeyChecking=no'
          hostname: AnsibleWorker02
```

Création dy fichier galaxy

```
roles/
|-- docker
|-- wordpress
```

```
├── hosts.yaml
├── docker.yaml
├── galaxy.yaml
```

vi galaxy.yaml

Création dy fichier main.yml de meta

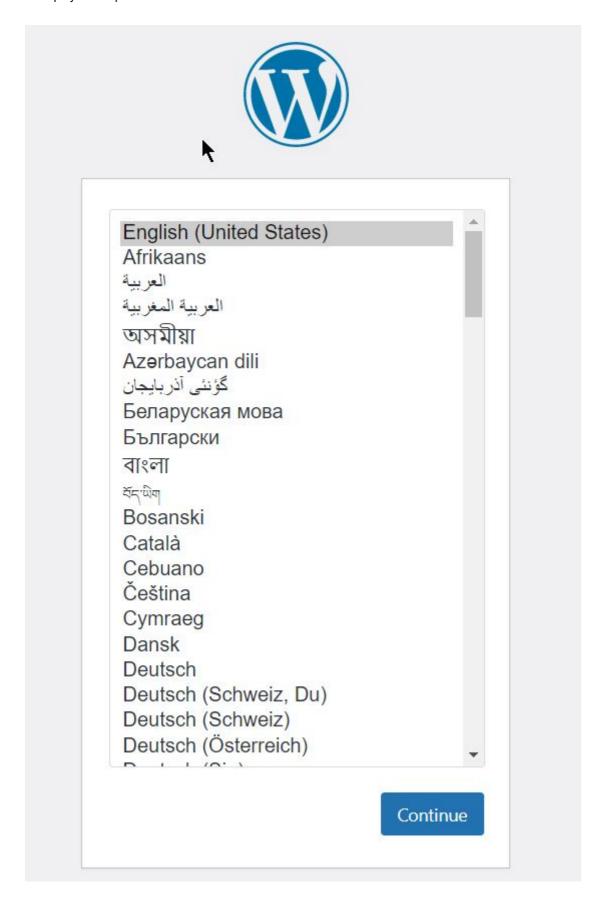
vi main.yml

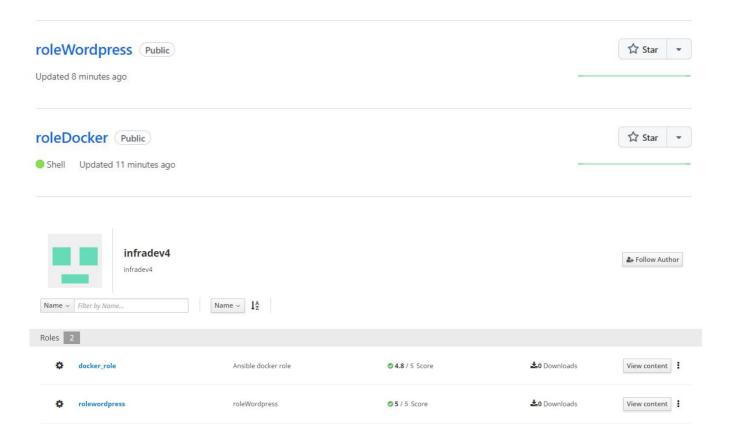
```
galaxy_info:
  author: Oussama ZAID
  description: Setup simple wordpress site using docker container
  license: COPYING
  min_ansible_version: core 2.12.1
```

ubuntu@AnsibleMaster:~/roles\$ ansible-playbook -i hosts.yaml docker.yaml

```
********************
TASK [Gathering Facts]
******************************
*************
ok: [worker02]
ok: [worker01]
TASK [docker : push script file]
*******************************
***********
ok: [worker02]
ok: [worker01]
TASK [docker : download pip script]
                     ***************
***********
ok: [worker01]
ok: [worker02]
TASK [docker : install python-pip]
*******************************
**********
skipping: [worker01]
skipping: [worker02]
TASK [docker : Install docker python]
***********************************
**********
skipping: [worker01]
skipping: [worker02]
TASK [docker : run the script]
***********************************
************
skipping: [worker01]
skipping: [worker02]
TASK [wordpress : Create a network]
**********
                    *****************
ok: [worker02]
ok: [worker01]
TASK [wordpress : start mariadb containers]
***********************************
*********
ok: [worker02]
ok: [worker01]
TASK [wordpress : start wordpress containers]
***********************************
*********
changed: [worker02]
```

changed: [worker01] TASK [wordpress : Add a container to a network, leaving existing containers changed: [worker02] changed: [worker01] PLAY RECAP ************************ ***************** worker01 : ok=7 changed=2 unreachable=0 failed=0 skipped=3 rescued=0 ignored=0 changed=<mark>2</mark> unreachable=<mark>0</mark> failed=0 worker02 : ok=7 skipped=<mark>3</mark> rescued=0 ignored=0





S