Docker / Ansible

Infrastructure

1. Analyze the `docker-compose.yml` file.

```yaml
version: "3.9"
services:

  alpine01:
    container_name: alpine01
    hostname: alpine01
    build: ./build
    ports:
    - "8081:80"
    volumes:
      - ./alpine01:/opt
    networks:
      - alpine-network

  alpine02:
    container_name: alpine02
    hostname: alpine02
    build: ./build
    ports:
    - "8082:80"
    volumes:
      - ./alpine02:/opt
    networks:
      - alpine-network

networks:
  alpine-network:
```

2. Use `docker-compose up -d`

3. Execute followgin commands to start up the `ansible` container:

```
docker run --rm -it --name ansible --workdir /root -h ansible -v $PWD/playbooks:/opt/playbooks -v $PWD/workspace:/opt/workspace --network docker-compose_alpine-network woahbase/alpine-ansible:x86_64 bash
PS1="[\h]\w $ "
ssh-keygen -t rsa -b 4096 -f /root/.ssh/id_rsa -q -P ""
cp .ssh/id_rsa.pub /opt/workspace/authorized_keys
```

⚠ **DO NOT EXIT** this container.

4. In another terminal windows will copy the `authorized_keys` into `alpine01` and `alpine02`

   ```
   docker cp workspace/authorized_keys alpine01:/root/.ssh/
   docker cp workspace/authorized_keys alpine02:/root/.ssh/
   ```

   Change the *owner* and *group* of the `authorized_keys`

   ```
   docker exec alpine01 chown root:root .ssh/authorized_keys
   docker exec alpine02 chown root:root .ssh/authorized_keys
   ```

5. Add a password just for 1 time to each container.

   ```
   docker exec alpine01 ash -c "echo -e 'ad-inP1wX\nad-inP1wX' | passwd > /dev/null
   2>&1"
   docker exec alpine02 ash -c "echo -e 'ad-inP1wX\nad-inP1wX' | passwd > /dev/null
   2>&1"
   ```

6. Make sure you have connection without password prompt. This is in the `ansible` container.

   ```
   ssh alpine01
   ssh alpine02
   ```

   You can exit from the `alpine01` and `alpine02` containers.

When working with multiple containers, you can execute `PS1="[\h]\w $ "` to change the prompt.

Ansible Tests

Ansible depends of an inventory file located at `/etc/ansible/hosts`. A typical inventory file can list the managed host either by IP address or by domain names. It is also possible to list one managed host in more than one group.

1. Modify the `hosts` with the following content:

   /etc/ansible/hosts

   ```
   [web]
   alpine01
   alpine02

   [backend]
   alpine02
   ```

2. In order to avoid any *warning* add the following content to the `ansible.cfg` file:

   /etc/ansible/ansible.cfg

   ```
   [defaults]
   interpreter_python = /usr/bin/python3
   ```

Ansible Command Line

- List all hosts

  ```
  ansible all --list-hosts
  ```

- List by group

  ```
  ansible web --list-hosts
  ```

- Ping per group

  ```
  ansible web -m ping
  ```

- Ping per host

  ```
  ansible alpine01 -m ping
  ```

- List available modules

  ```
  ansible-doc -l
  ```

- Copy

  ```
  ansible backend -m copy -a "src=/root/text dest=/root/"
  ```

- Command (**default** module)

  ```
  ansible web -m command -a "ls -lrt /root"
  ```

- Shell

  ```
  ansible web -m shell -a "date"
  ```

Ansible Playbooks

playbook.yaml

```
---
- hosts: web
  tasks:
  - name: Update and build UI
    shell: |
      git pull
      npm run build
    args:
      chdir: /opt/web
```

```
ansible-playbook playbook.yml
```

Live Demo

This demo will demonstrate how to use Ansible with a connected WebHook from Github.

Cleanup

```
docker-compose down
docker rmi -f docker-compose_alpine01
docker rmi -f docker-compose_alpine02
docker rm -f $(docker ps -qa)
```

Adding Software in Alpine OS

```
apk add --no-cache --update-cache nodejs nodejs-npm git
```