

Раздел 1. Установка docker и ansible-модуля для работы с docker

До начала работ нам необходимо установить docker и ansible-модуля для работы с ним

Ссылки на дистрибутивы:

- ansible модуль для docker:

https://docs.ansible.com/ansible/latest/collections/community/docker/docker_container_module.html

- docker: <https://docs.docker.com/engine/install/ubuntu/>

--

Устанавливаем ansible-модуль для работы с docker (ubuntu):

```
$ sudo ansible-galaxy collection install community.docker
```

--

Устанавливаем docker (ubuntu):

```
$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

```
$ sudo mkdir -p /etc/apt/keyrings
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
gpg --dearmor -o /etc/apt/keyrings/docker.gpg
```

```
$ echo \
    "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
docker-compose-plugin
```

Раздел 2. Первый пример по работе с ansible

Переходим в директорию "nginx"

В файле `playbook.yml` содержится наш плейбук, в котором мы собираем docker-образ с nginx и запускаем его в контейнере

```

1 ---
2 - hosts: localhost                                # Указываем хост, на котором отработает плейбук
3   tasks:                                           # Указываем список задач.
4 # Первой задачей соберём Docker образ nginx с картинкой на веб-странице.
5   - name: Build an image
6     community.docker.docker_image:               # Используем модуль docker.docker_image.
7       name: myimage:latest                        # Указываем название:тег будущего образа.
8       build:                                     # Указываем путь до места сборки с Dockerfile.
9         path: .
10        source: build
11
12 # Второй задачей запускаем собранный Docker образ.
13 - name: Create a data container
14   community.docker.docker_container:             # Используем модуль docker.docker_container
15     name: example-container                      # Выбираем название будущего контейнера.
16     image: myimage:latest                       # Указываем название образа, который собрали в первой задаче.
17     published_ports: 80:80                      # Указываем порты, которые нужно перенаправить из контейнера.
18
19 # После отработки плейбука можно зайти на localhost (127.0.0.1:80) и увидеть веб-страницу.

```

В файле Dockerfile мы найдем указание, какой docker-образ нужно взять за основу (nginx последней версии)

```

1 # Pull base image.
2 FROM nginx:latest
3

```

Для запуска плейбука выполните команду:

```
$ ansible-playbook playbook.yaml
```

```

root@filipp-notebook:/home/filipp/Desktop/ans-edu/nginx# ansible-playbook pl
aybook.yaml
[WARNING]: provided hosts list is empty, only localhost is available. Note t
hat
the implicit localhost does not match 'all'

PLAY [localhost] *****
****

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

TASK [Build an image] *****
****
changed: [localhost]

TASK [Create a data container] *****
****
changed: [localhost]

PLAY RECAP *****
****
localhost                : ok=3    changed=2    unreachable=0    failed=0
    skipped=0    rescued=0    ignored=0

```

Для проверки того, что контейнер с nginx запущен, выполните команду:

```
$ docker ps
```

```
root@filipp-notebook:/home/filipp/Desktop/ans-edu/nginx# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
39efbf9e3498   myimage:latest "/docker-entrypoint...." 4 seconds ago Up
3 seconds     0.0.0.0:80->80/tcp example-container
```

Для проверки успешности выполненного запуска плейбука, выполните команду:

```
$ curl 127.0.0.1
```

```
root@filipp-notebook:/home/filipp/Desktop/ans-edu/nginx# curl 127.0.0.1
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Или воспользуйтесь своим браузером и перейдите по адресу <http://127.0.0.1/>

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Для остановки и удаления запущенного контейнера выполните команды:

```
$ docker stop example-container  
$ docker rm example-container
```

```
root@filipp-notebook:/home/filipp/Desktop/ans-edu/nginx# docker stop example-  
-container  
example-container  
root@filipp-notebook:/home/filipp/Desktop/ans-edu/nginx# docker rm example-c  
ontainer  
example-container
```

Раздел 3. Второй пример по работе с ansible

Переходим в директорию "boor"

В файле mastery-hosts, который выступает в качестве инвентори-файла, указано описание хостов

```
1 [demo]  
2 localhost  
3  
4 [demo:vars]  
5 ansible_connection=local
```

В файле docker-all.yaml указан наш плейбук, который мы будем запускать

```

1 ---
2 - name: build an image
3   hosts: localhost
4   gather_facts: false
5
6   tasks:
7     - name: start the container
8       community.docker.docker_container:
9         name: playbook-container
10        image: docker.io/fedora:34
11        ports: 8080:80
12        state: started
13        command: sleep 500
14        container_default_behavior: no_defaults
15
16     - name: make a host
17       ansible.builtin.add_host:
18         name: playbook-container
19         ansible_connection: docker
20         ansible_ssh_user: root
21
22 - name: do things
23   hosts: playbook-container
24   gather_facts: false
25
26   tasks:
27     - name: install things
28       ansible.builtin.raw: dnf install -y python-dnf
29
30     - name: install things
31       ansible.builtin.dnf:
32         name: ['nginx', 'cowsay']
33
34     - name: configure nginx
35       ansible.builtin.lineinfile:
36         line: "daemon off;"
37         dest: /etc/nginx/nginx.conf
38
39     - name: boop
40       ansible.builtin.shell: cowsay boop > /usr/share/nginx/html/index.html
41
42     - name: run nginx
43       ansible.builtin.shell: nginx &

```

Перед запуском плейбука убедитесь, что запускаемый контейнер еще не запущен (например, если вы выполняете этот пример не в первый раз)

```
$ docker ps -a --filter ancestor=fedora-moo
```

(вывод команды должен быть пустым)

Для запуска плейбука выполните команду:

```
$ ansible-playbook -i mastery-hosts docker-all.yaml
```

```

root@filipp-notebook:/home/filipp/Desktop/ans-edu/boop# ansible-playbook -i ma
tery-hosts docker-all.yaml

PLAY [build an image] *****
*

TASK [start the container] *****
*
changed: [localhost]

TASK [make a host] *****
*
changed: [localhost]

PLAY [do things] *****
*

TASK [install things] *****
*
changed: [playbook-container]

TASK [install things] *****

```

Убедитесь, что контейнер запущен:

```
$ docker ps
```

```

root@filipp-notebook:/home/filipp/Desktop/ans-edu/boop# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
386a78a8caad   fedora:34     "sleep 500"             2 minutes ago Up 2 minutes  0.0.0.0
:8080->80/tcp   playbook-container

```

Выполните проверочную команду:

```
$ curl http://localhost:8080
```

(если корова сказала “boop”, значит плейбук отработал корректно)

```

root@filipp-notebook:/home/filipp/Desktop/ans-edu/boop# curl http://localhost:
8080
< boop >
-----
      \  ^__^
         (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||

```

После проверки, для того, чтобы остановить и удалить контейнер выполните команды:

```

$ docker stop playbook-container
$ docker rm playbook-container

```

```
root@filipp-notebook:/home/filipp/Desktop/ans-edu/boop# docker stop playbook-co
ntainer
playbook-container
root@filipp-notebook:/home/filipp/Desktop/ans-edu/boop# docker rm playbook-cont
ainer
playbook-container
```