Automate Docker With Ansible for launching Webserver

Ansible is one of the most powerful tool for **configuration management**. This tool is very simple to use yet powerful enough to automate complex multi-tier IT application environments.

This is a small task in which we learn to launch our own Webserver using Docker and Ansible that use the YAML file to automate the process.

Task Description:

Write an Ansible PlayBook that does the following operations in the managed nodes:

- Configure Docker.
- Start and enable Docker services.
- Pull the Httpd server image from the Docker Hub.
- Run the Httpd container and expose it to the public.

- Copy the HTML code in /var/www/html directory and start the webserver.

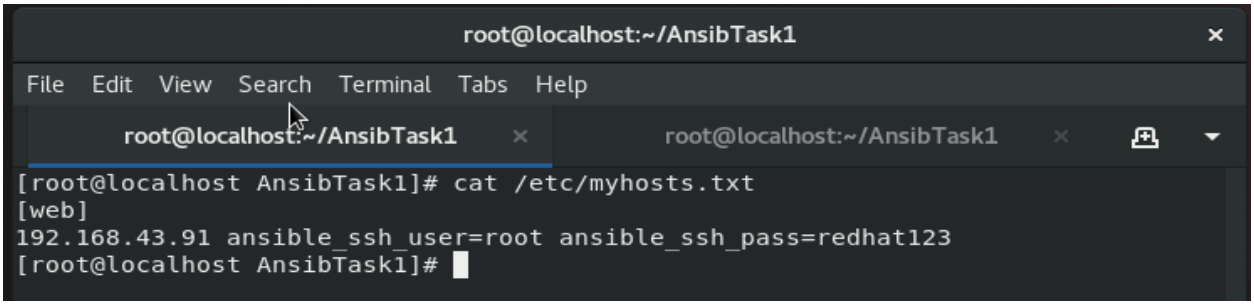
Installing Ansible in Linux:

I'm using RHEL 8 here. You only need to have Python3 software installed. Ansible uses ssh to go inside the host OS so we also need one more software **sshpass.** Run these two commands to install Ansible...

```
pip3 install ansible
yum install sshpass
```

Configuration of Ansible:

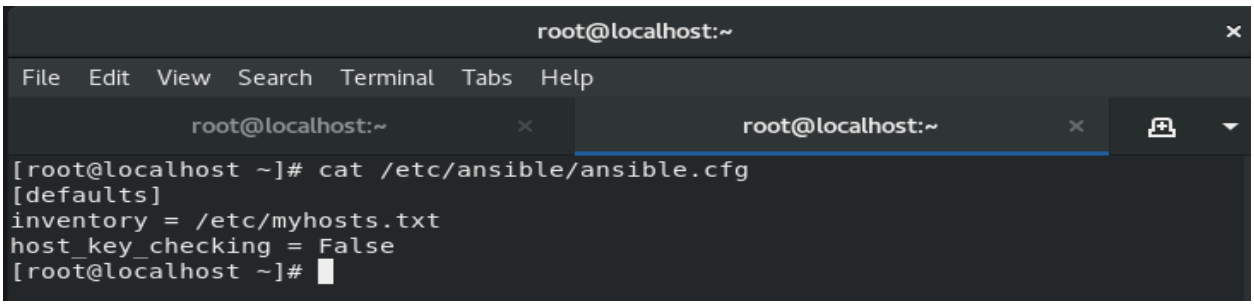
Now we need to create one file where we give IP of our host server...



Ansible uses this file for check hosts OS and after that Go inside the OS using username and password. This also known as Inventory File.

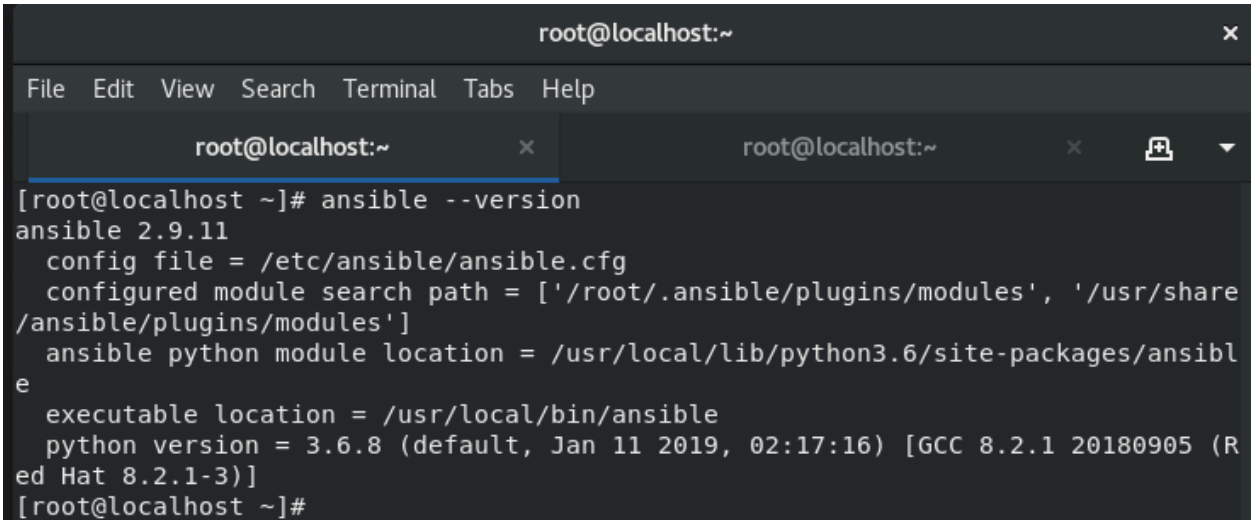
After that, create one config file for Ansible where we give the location of the host file which we created above. Go to /etc and create one folder named **ansible**. Go inside the folder and create one config file named **ansible.cfg**

```
mkdir -p /etc/ansible
cd ansible
gedit ansible.cfg
```



Now for check that Ansible is working fine, Run...

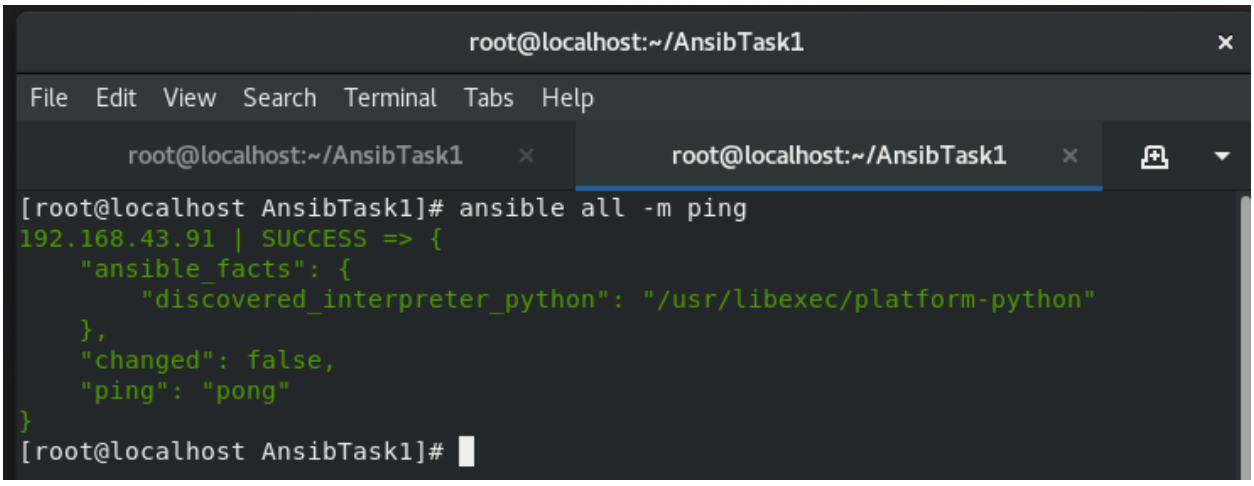
```
ansible --version
```



This is the output of the command.

For checking connectivity that all host node is properly connected, RUN

```
ansible all -m ping
```



This is the output of the command.

Now let us come to the task...

We create one file named apache.yaml which also known as Ansible PlayBook. In this, we write vars and tasks which we fill as we learn further.

```
- hosts: web
  vars:
    #varibles go there
  tasks:
    #tasks go there
```

Variables:

These are the variables which we use in our playbook.

```
  vars:
    - image_name: httpd
    - file: home.html
    - docker_volume: /webvolume/
    - docker_host_port: 8025
    - docker_container_port: 80
```

- **Configure Docker:**

For this, we need to Configure Docker Repo First...

```
    - name: Add Yum Repo
      yum_repository:
        name: docker
        description: Docker Yum Repo
        baseurl: https://download.docker.com/linux/centos/7/x86_64/stable/
        gpgcheck: no
```

Install Docker Task...

Redhat by default do not provide Docker Software now. So, we need ***--nobest*** option in command for installing Docker.

In ansible, **yum module** in Ansible does not have this argument till yet, so I use the **command module** for doing the installation.

```
    - name: Install Docker-CE
      command: yum install -y  docker-ce --nobest
```

Install Docker Requirement on hosts...

We need ***docker-py*** module to run docker from ansible tool.

```
    - name: Install Docker Requirement On Host
      command: pip3 install docker-py
```

- **Start and Enable Docker service:**

This task starts and enable the Docker Service in hosts.

```
    - name: start docker services
      service:
        name: "docker"
        state: started
        enabled: yes
```

- **Pull the Httpd Image from Docker Hub:**

This task pulls the docker image to launch the Container.

```
    - name: pull an image
      docker_image:
        name: "{{ image_name }}"
        source: pull
```

- Task for Copy Html code from controller node to managed Node:

```
    - name: Copy Web Page or HTML code to Hosts
      template:
        src: "{{ file }}"
        dest: "{{ docker_volume }}"
```

This code copy the HTML file to the managed Node.

- **Launch the Container and exposed it to outer world:**

```
    - name: Run docker container
      docker_container:
        name: webserver
        image: "{{ image_name }}"
        interactive: yes
        volumes:
          - "{{ docker_volume }}:/usr/local/apache2/htdocs"
        ports:
          - "{{ docker_host_port }}:{{ docker_container_port }}"
        command: httpd -D FOREGROUND
```

Final PlayBook:

When put the above code in one file...

```
- hosts: web
  vars:
    - image_name: httpd
    - file: home.html
    - docker_volume: /webvolume/
    - docker_host_port: 8025
    - docker_container_port: 80
  tasks:
    - name: Add Yum Repo
      yum_repository:
        name: docker
        description: Docker Yum Repo
        baseurl: https://download.docker.com/linux/centos/7/x86_64/stable/
        gpgcheck: no
    - name: Install Docker-CE
      command: yum install -y  docker-ce --nobest
    - name: start docker services
      service:
        name: "docker"
        state: started
        enabled: yes
    - name: Install Docker Requirement On Host
      command: pip3 install docker-py
    - name: pull an image
      docker_image:
        name: "{{ image_name }}"
        source: pull
    - name: Copy Web page or HTML code to Hosts
      template:
        src: "{{ file }}"
        dest: "{{ docker_volume }}"
    - name: Run docker container
      docker_container:
        name: webserver
        image: "{{ image_name }}"
        interactive: yes
        volumes:
          - "{{ docker_volume }}:/usr/local/apache2/htdocs"
        ports:
          - "{{ docker_host_port }}:{{ docker_container_port }}"
        command: httpd -D FOREGROUND
```

Running the PlayBook:

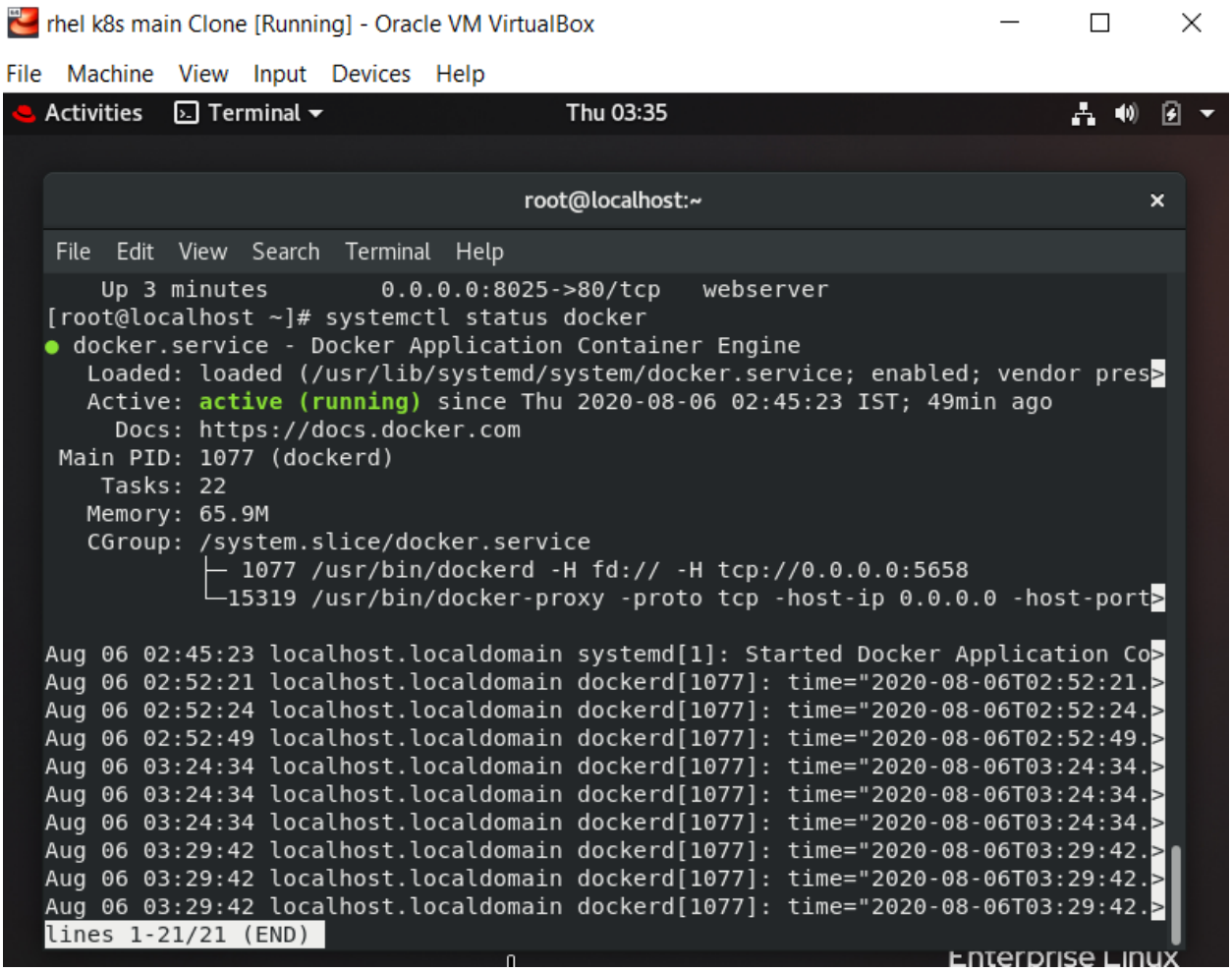
Command for run playbook...

```
ansible-playbook apache.yaml
```

```
[root@localhost AnsibTask1]# ansible-playbook apache.yaml

PLAY [web] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.43.91]

TASK [Add Yum Repo] ************************************************************
changed: [192.168.43.91]

TASK [Install Docker-CE] *******************************************************
[WARNING]: Consider using the yum module rather than running 'yum'.  If you
need to use command because yum is insufficient you can add 'warn: false' to
this command task or set 'command_warnings=False' in ansible.cfg to get rid of
this message.
changed: [192.168.43.91]

TASK [start docker services] ***************************************************
ok: [192.168.43.91]

TASK [Install Docker Requirement On Host] **************************************
```

```
TASK [Install Docker Requirement On Host] **************************************
changed: [192.168.43.91]

TASK [pull an image] ***********************************************************
ok: [192.168.43.91]

TASK [Copy Web page or HTML code to Hosts] *************************************
ok: [192.168.43.91]

TASK [Run docker container] ****************************************************
changed: [192.168.43.91]

PLAY RECAP *********************************************************************
192.168.43.91              : ok=8    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

Check the managed node using systemctl status docker

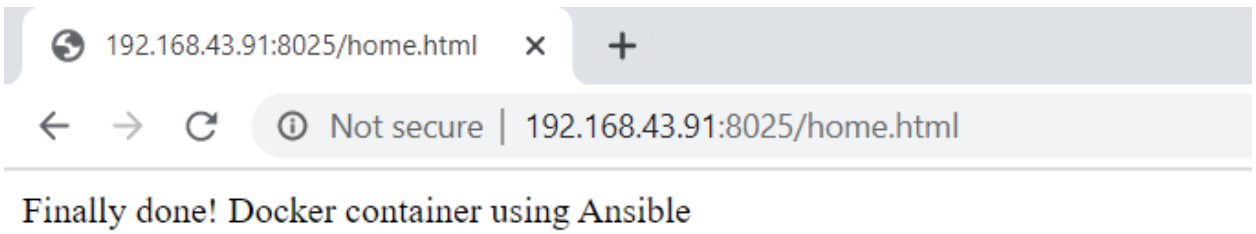


⇨ docker ps

```
[root@localhost ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
    STATUS              PORTS                    NAMES
11005fd926e1        httpd               "httpd -D FOREGROUND"    3 minutes ago
    Up 3 minutes        0.0.0.0:8025->80/tcp     webserver
[root@localhost ~]#
```

Testing the results:

Run **IP:port**(192.168.43.91:8025) for checking that your webserver working fine or not...



Task Completed

GitHub Repo: https://github.com/Anuddeeph/Ansible-Docker-Webserver.git

Thanks for Reading...

For any Query or Suggestions, feel free to DM me...