Create web server on docker container on EC2 instance automated using Ansible playbook

So in this repo , I will be deploying apache web server using Ansible on AWS EC2 instance. You can use your local VM also.

Necessary Requirements:

1. Basic Knowledge of launching and managing instances in Cloud.
2. Already installed Ansible.

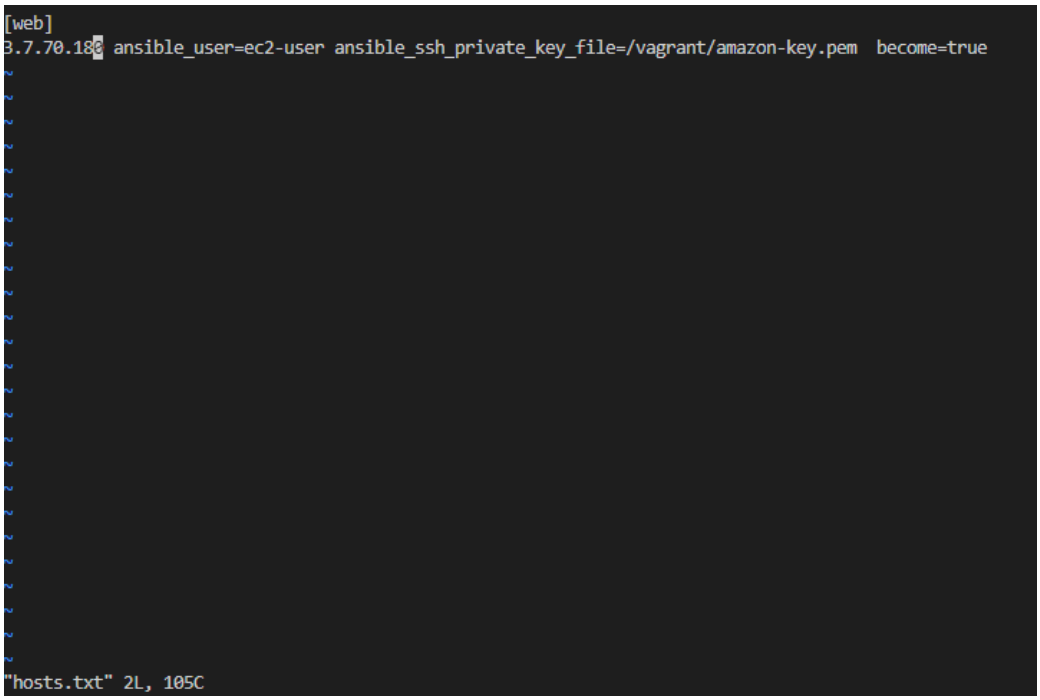
Step 1: Launch your VM on which you want to setup your web server. Then keep a record of it's IP address , username, password or key in case of cloud. That's all you have to do in that VM and leave it running.

Note: Now all the steps are to be performed in Base OS where you have ansible tool.

Step 2: Create an ***Inventory file*** anywhere with any name. Inventory file contains all information of remote machines where you want to perform your operation. In my case i have created in etc directory.

```
vim /etc/myhosts.txt
```

Step 3: Enter the information you gathered in *#Step 1*.
In case of local VM you have to give **ansible_ssh_pass = <password>** in case of **ansible_ssh_private_key_file.**



Step 4: Create a ansible configuration file if not.

```
mkdir /etc/ansible
vim /etc/ansible/ansible.cfg
```

Now enter below **path** of Inventory file and **disable host_key_checking in it**.

```
root@localhost:~/ansible-playbooks
[defaults]
inventory = /etc/myhosts.txt
host_key_checking = False
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/etc/ansible/ansible.cfg" 3L, 66C                    3,25          All
```

Step 5: You can check your connectivity to manger node by below command

```
ansible all -m ping
```

```
root@terraform:/vagrant# ansible all -m ping
3.7.70.180 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Step 6: Now you have to create an **ansible playbook** which contains all commands. You can create it anywhere with anyname but extension must be **yaml** or **yml**.
Then you have to enter commands for setting up web server.

Step 7: Now below will be steps for code completion.

- Copy Html Code to Controller Node
- Installing Docker Prerequisite packages

- Configuring docker-ce repo
- replacing rhel to centos in docker ce repo
- Installing Docker latest version
- Start service docker
- Installing python library for the Docker Remote API
- pull httpd image
- In the end creating docker container with httpd image, exposing port, mounting our volume

  Below is the code for your reference:

```
---
  - hosts: all
    remote_user: root
    become: true
    gather_facts: no
    ignore_errors: yes
    tasks:
      - name: Copy Html Code to Controller Node
        copy:
          src:  index.html
          dest: /tmp

      - name: "Installing Docker Prerequisite packages"
        yum:
          name: yum-utils
          state: present

      - name: Configuring docker-ce repo
        get_url:
          url: https://download.docker.com/linux/rhel/docker-ce.repo
          dest: /etc/yum.repos.d

      - name: replacing rhel to centos in docker ce repo
        replace:
          path: /etc/yum.repos.d/docker-ce.repo
          regexp: 'rhel'
          replace: 'centos'
```

```
    - name: " Installing Docker latest version"
      yum:
        name: "{{item}}"
        state: present
      loop:
        - docker-ce
        - docker-ce-cli
        - containerd.io
        - docker-compose-plugin

    - name: Install pip and apache
      yum:
        name: "{{item}}"
        state: present
      loop:
        - python-pip

    - name: Start service docker
      service:
        name: docker
        state: started

    - name: Installing python library for the Docker Remote API
      pip:
        name: docker==4.4.4

    - name: "pull httpd image"
      docker_image:
          name: httpd
          source: pull

    - name: Create a web container
      docker_container:
        name: web
        image: httpd
        state: started
        ports: 3000:80
        volumes: /tmp/index.html:/usr/local/apache2/htdocs/index.html
```

Step 8: Now run below command to run your playbook

```
ansible-playbook </path/to/your/playbbok/file>
```

```
TASK [Installing Docker Prerequisite packages] ***********************************************************
ok: [3.7.70.180]

TASK [Configuring docker-ce repo] ************************************************************************
changed: [3.7.70.180]

TASK [Installing Docker latest version] ******************************************************************
ok: [3.7.70.180] => (item=docker-ce)
ok: [3.7.70.180] => (item=docker-ce-cli)
ok: [3.7.70.180] => (item=containerd.io)
ok: [3.7.70.180] => (item=docker-compose-plugin)

TASK [Install pip and apache] ****************************************************************************
ok: [3.7.70.180] => (item=python-pip)

TASK [Start service docker] ******************************************************************************
changed: [3.7.70.180]

TASK [Installing python library for the Docker Remote API] ***********************************************
ok: [3.7.70.180]

TASK [pull httpd image] **********************************************************************************
ok: [3.7.70.180]

TASK [Create a web container] ****************************************************************************
changed: [3.7.70.180]

PLAY RECAP ***********************************************************************************************
```

If you will do curl *http://localhost 80* on the ec2 instance then you will be able you see the content of your webpage. Now you can host it in your local machine laptop/mobile through localhost.

```
[ec2-user@ip-172-31-5-14 ~]$ curl http://localhost 80
 <html> <body><h1> this is web page from docker through EC2 instance</h1></body></html>
```

If you are not able to connect, there can be issue of firewall security, you can stop by using below command

```
systemctl stop firewalld
```