🔰Ansible playbook that will retrieve new Container IP and dynamically update the inventory and Configure web-server inside that Docker Container🔰

ARTH — Task 14 🤖

Task Description📄

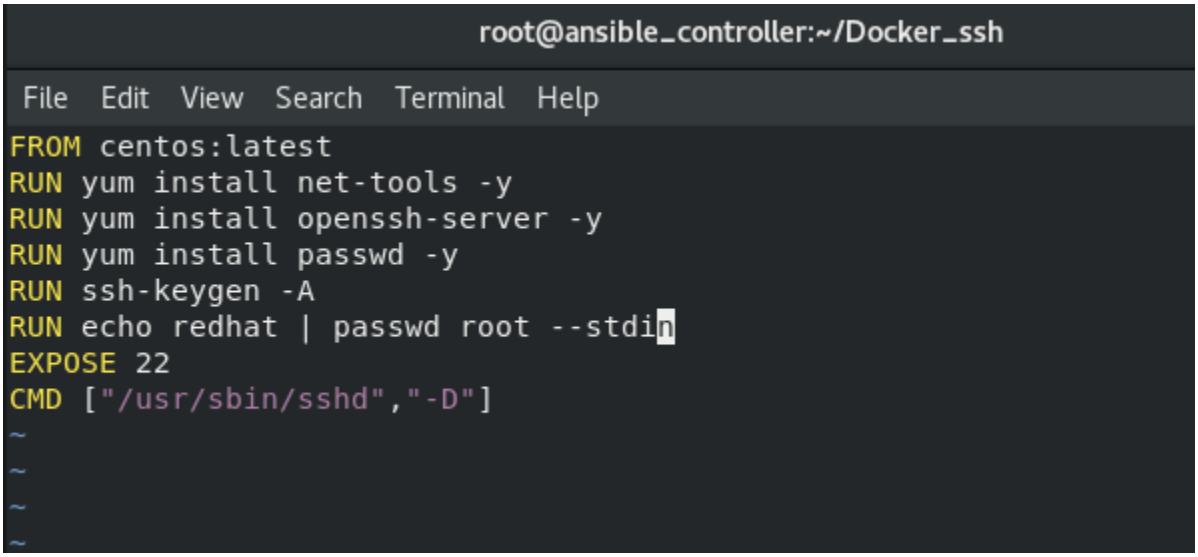
🔰14.2 Further in ARTH — Task 10 have to create an Ansible playbook that will retrieve new Container IP

and update the inventory. So that further Configuration of Webserver could be done inside that Container.

👉Lets get started… 😃

In this task I have used my own pre-created docker image which is enabled with ***ssh****. So that Ansible can use ssh protocol to login to new docker container and configure webserver inside it.*

Code of Dockerfile



Docker Hub
hub.docker.com

In my case I am having Controller Node having ***IP on which the ansible is installed***

✔ *Lets check the ansible version installed on my controller node:*

```
# ansible –version
```

root@localhost:~/Task14

```
[root@localhost Task14]# ansible --version
ansible 2.10.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.6/site-packages/ansible
  executable location = /usr/local/bin/ansible
  python version = 3.6.8 (default, Jan 11 2019, 02:17:16) [GCC 8.2.1 20180905 (Red Hat 8.2.1-3)]
[root@localhost Task14]#
```

Initially inventory file is empty

```
# cat /root/ip.txt
```

root@localhost:~

```
[root@localhost ~]# cat /root/docip.txt
[root@localhost ~]#
```

✔ *Configuration file of ansible:*

```
# vim /etc/ansible/ansible.cfg
```

root@localhost:~

```
[defaults]
inventory = /root/Task14/docip.txt
host_key_checking = False
deprecation_warnings=False
```

◈ *Ansible Playbook :*

```
# vim docker.yml
```

root@localhost:~/Task14

```
- hosts: localhost
  vars_prompt:
    - name: container_name
      prompt: "Enter Docker Container Name:"
      private: no
  vars:
          - image_name: anuddeeph/centos_ssh:v2
  tasks:
    - name: Add Yum Repo
      yum_repository:
        name: docker
        file: 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: "Launching {{ container_name }} Container"
      docker_container:
        name: "{{ container_name }}"
        image: "{{ image_name }}"
        state: started
        interactive: yes
        detach: yes
        tty: yes
    - name: "Container Info"
      docker_container_info:
        name: "{{ container_name }}"
      register: docker_info

    - debug:
        var: docker_info.container.NetworkSettings.IPAddress
    - name: "Retriving IP dynamically and updating in the inventory"
      template:
        src: "dockerip.j2"
        dest: "/root/Task14/docip.txt"

~
~
```

```
# vim docker-web.yml
```

root@localhost:~/Task14

```
- hosts: docker
  vars:
    - file_name: Task14.html
  tasks:
    - name: "Installing Httpd"
      package:
        name: "httpd"
        state: present

    - name: "Deploying webpage to /var/www/html"
      copy:
        src: "{{ file_name }}"
        dest: "/var/www/html"

    - name: "Starting httpd Service"
      command: "/usr/sbin/httpd"

~
```

dockerip.j2 file

root@localhost:~/Task14

```
[root@localhost Task14]# cat dockerip.j2
[docker]
{{ docker_info['container']['NetworkSettings']['IPAddress'] }} ansible_ssh_user=root ansible_ssh_pass=redhat ansible_connection=ssh
[root@localhost Task14]#
```

Task14.html file

root@localhost:~/Task14

```
[root@localhost Task14]# ls
docip.txt  dockerip.j2  docker_web.yml  docker.yml  Task14.html
[root@localhost Task14]# cat Task14.html

<h1 align="center"> TASK 14.2 SUCCESSFULL </h1>
[root@localhost Task14]#
```

Now let's run the main Playbook:

```
# ansible-playbook docker.yml
```

root@localhost:~/Task14

```
[root@localhost Task14]# ansible-playbook docker.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'
Enter Docker Container Name:: WebOS

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

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

TASK [Add Yum Repo] ************************************************************************
ok: [localhost]

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: [localhost]

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

TASK [Install Docker Requirement On Host] **************************************************
changed: [localhost]

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

TASK [Launching WebOS Container] ***********************************************************
changed: [localhost]

TASK [Container Info] **********************************************************************
ok: [localhost]

TASK [debug] *******************************************************************************
ok: [localhost] => {
    "docker_info.container.NetworkSettings.IPAddress": "172.17.0.2"
}

TASK [Retriving IP dynamically and updating in the inventory] ******************************
changed: [localhost]

PLAY RECAP *********************************************************************************
localhost                  : ok=10   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@localhost Task14]#
```

root@localhost:~/Task14

```
[root@localhost Task14]# docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS              PORTS               NAMES
b1a4ce34354a        anuddeeph/centos_ssh:v2   "/usr/sbin/sshd -D"   2 minutes ago       Up 2 minutes        22/tcp              WebOS
[root@localhost Task14]#
```

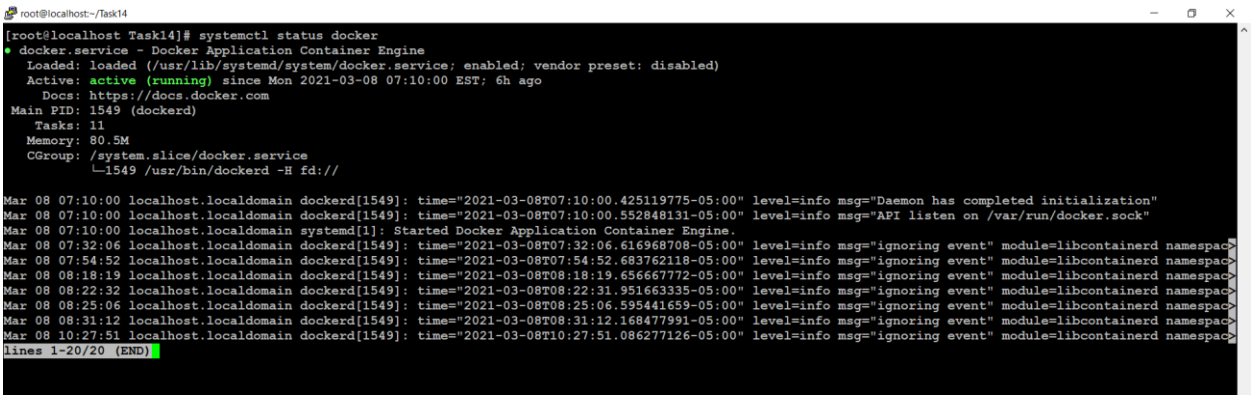
Container is successfully launched with name Webos.

Inventory file dynamically updated

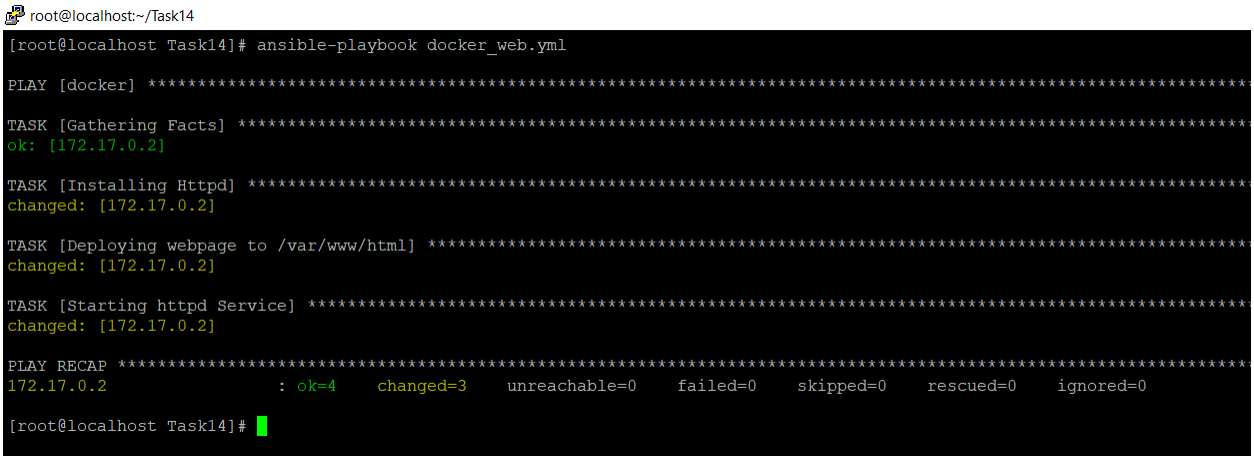
root@localhost:~/Task14

```
[root@localhost Task14]# cat docip.txt
[docker]
172.17.0.2 ansible_ssh_user=root ansible_ssh_pass=redhat ansible_connection=ssh
[root@localhost Task14]#
```

Check Docker Service started and enabled:

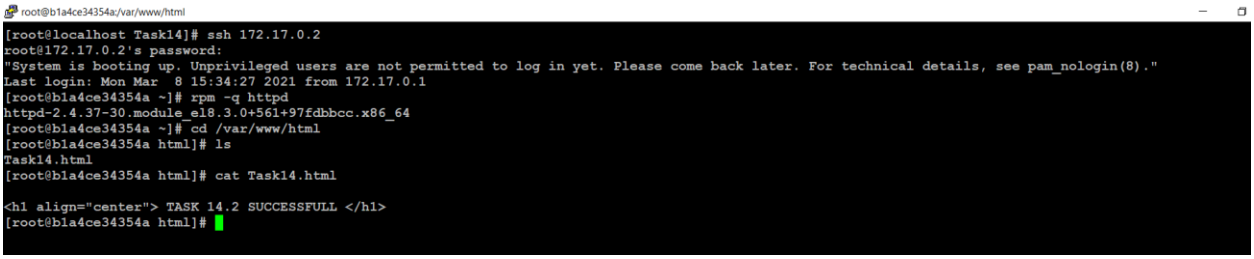


*Now we can run our **docker-web.yml** playbook to configure webserver inside container:*



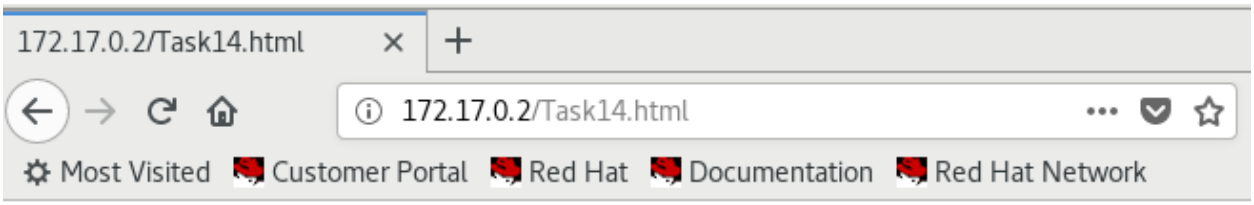
Done!

*Now checking **webos** container:*



Now the webserver has been deployed on the docker container.

So Let's check our webpage from the browser:



TASK 14.2 SUCCESSFULL

GitHub Link: https://github.com/Anuddeeph/Docker-Dynamic-inventory-using-Ansible.git

TASK COMPLETED Successfully✌💾

Thanks for reading !!!😊✨

🔰Keep Learning !!🔰Keep Sharing !!