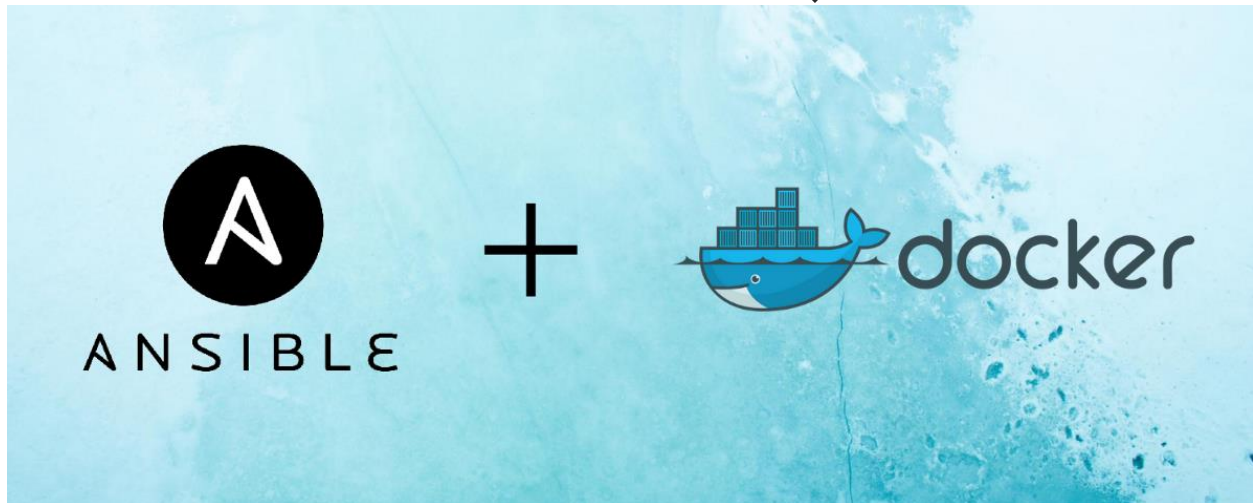


📖 Ansible playbook that will retrieve new Container IP and dynamically update the inventory and Configure web-server inside that Docker Container📖



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


```
root@ansible_controller:~/Docker_ssh
File Edit View Search Terminal Help
FROM centos:latest
RUN yum install net-tools -y
RUN yum install openssh-server -y
RUN yum install passwd -y
RUN ssh-keygen -A
RUN echo redhat | passwd root --stdin
EXPOSE 22
CMD ["/usr/sbin/sshd","-D"]
~
~
~
~
```

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
bla4ce34354a       anudeep/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:

```
root@localhost:~/Task14
[root@localhost Task14]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-03-08 07:10:00 EST; 6h ago
     Docs: https://docs.docker.com
   Main PID: 1549 (dockerd)
      Tasks: 11
     Memory: 80.5M
    CGroup: /system.slice/docker.service
            └─1549 /usr/bin/dockerd -H fd://

Mar 08 07:10:00 localhost.localdomain dockerd[1549]: time="2021-03-08T07:10:00.425119775-05:00" level=info msg="Daemon has completed initialization"
Mar 08 07:10:00 localhost.localdomain dockerd[1549]: time="2021-03-08T07:10:00.552848131-05:00" level=info msg="API listen on /var/run/docker.sock"
Mar 08 07:10:00 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
Mar 08 07:32:06 localhost.localdomain dockerd[1549]: time="2021-03-08T07:32:06.616968708-05:00" level=info msg="ignoring event" module=libcontainerd namespace=
Mar 08 07:54:52 localhost.localdomain dockerd[1549]: time="2021-03-08T07:54:52.683762118-05:00" level=info msg="ignoring event" module=libcontainerd namespace=
Mar 08 08:18:19 localhost.localdomain dockerd[1549]: time="2021-03-08T08:18:19.656667772-05:00" level=info msg="ignoring event" module=libcontainerd namespace=
Mar 08 08:22:32 localhost.localdomain dockerd[1549]: time="2021-03-08T08:22:31.951663335-05:00" level=info msg="ignoring event" module=libcontainerd namespace=
Mar 08 08:25:06 localhost.localdomain dockerd[1549]: time="2021-03-08T08:25:06.595441659-05:00" level=info msg="ignoring event" module=libcontainerd namespace=
Mar 08 08:31:12 localhost.localdomain dockerd[1549]: time="2021-03-08T08:31:12.168477991-05:00" level=info msg="ignoring event" module=libcontainerd namespace=
Mar 08 10:27:51 localhost.localdomain dockerd[1549]: time="2021-03-08T10:27:51.086277126-05:00" level=info msg="ignoring event" module=libcontainerd namespace=
lines 1-20/20 (END)
```

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

```
root@localhost:~/Task14
[root@localhost Task14]# ansible-playbook docker_web.yml

PLAY [docker] *****

TASK [Gathering Facts] *****
ok: [172.17.0.2]

TASK [Installing Httpd] *****
changed: [172.17.0.2]

TASK [Deploying webpage to /var/www/html] *****
changed: [172.17.0.2]

TASK [Starting httpd Service] *****
changed: [172.17.0.2]

PLAY RECAP *****
172.17.0.2 : ok=4 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[root@localhost Task14]#
```

Done!

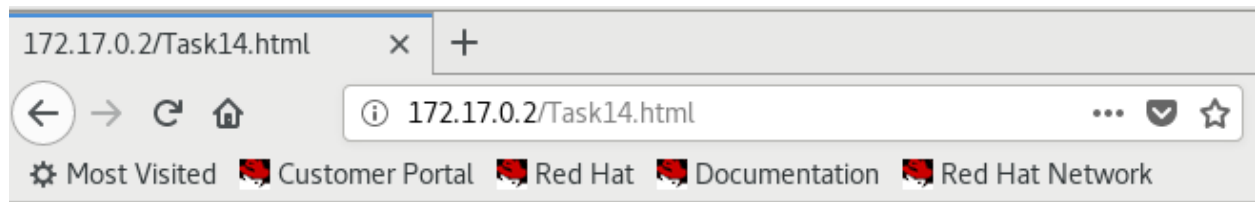
Now checking **webos** container:

```
root@b1a4ce34354a:/var/www/html
[root@localhost Task14]# ssh 172.17.0.2
root@172.17.0.2's password:
"System is booting up. Unprivileged users are not permitted to log in yet. Please come back later. For technical details, see pam_nologin(8)."
```

```
Last login: Mon Mar  8 15:34:27 2021 from 172.17.0.1
[root@b1a4ce34354a ~]# rpm -q httpd
httpd-2.4.37-30.module.el8.3.0+561+97fdbbcc.x86_64
[root@b1a4ce34354a ~]# cd /var/www/html
[root@b1a4ce34354a html]# ls
Task14.html
[root@b1a4ce34354a html]# cat Task14.html

<h1 align="center"> TASK 14.2 SUCCESSFULL </h1>
[root@b1a4ce34354a html]#
```

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 !!!