

Ansible Playbook to configure docker, enable it's services and start the webserver

1. First of all we will create a workspace

mkdir /wstask10

2. Create a repo file for Docker:

vim docker.repo

It basically provides the path from where docker shall be pulled.

```
root@localhost:/wstask10 x root@localhost:/ws11.3 x
[docker]
baseurl=https://download.docker.com/linux/centos/7/x86_64/stable/
gpgcheck=0
~
```

3. Create a html file to copy to the Target Node and Host the webpage

vim task.html

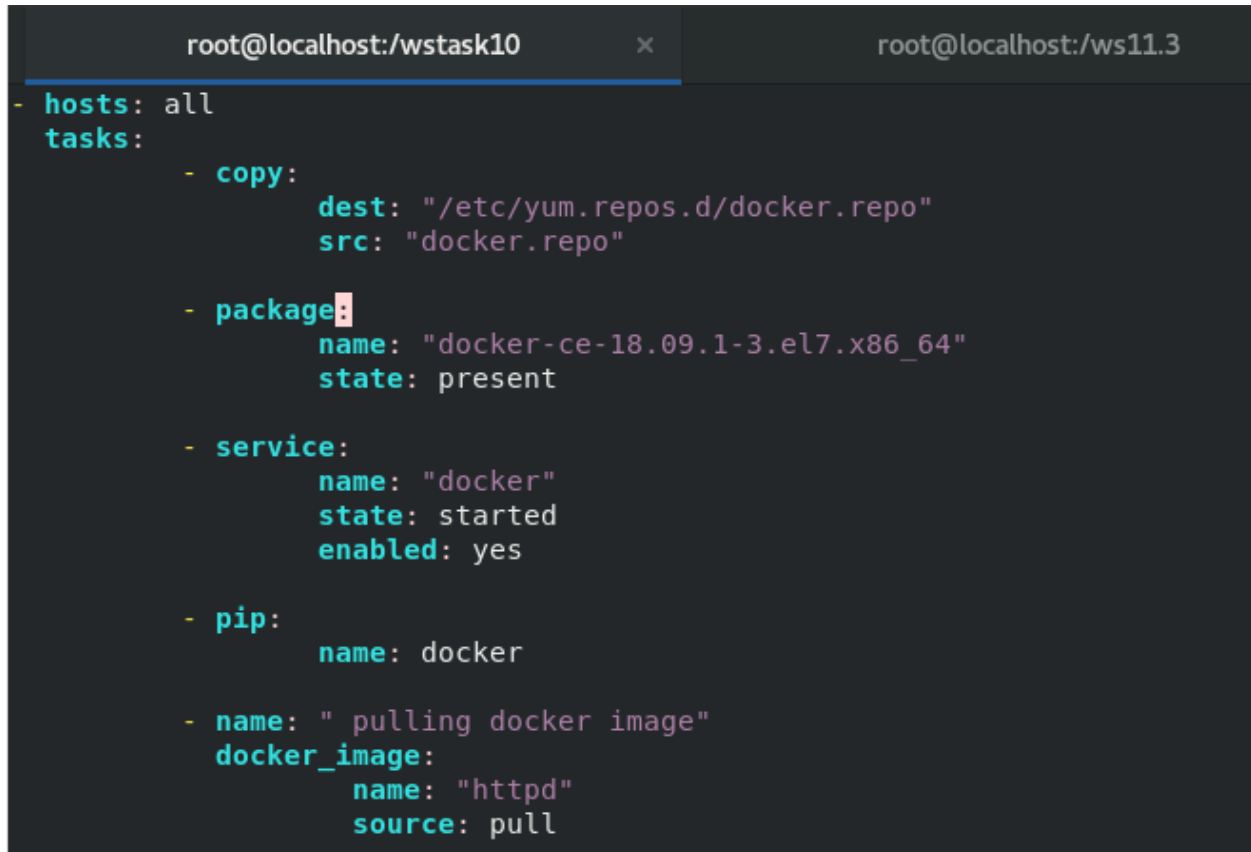
Content of html file is shown below:

```
[root@localhost wstask10]# cat task.html
<html>
<style>
body{
color:black; text-align:center;}
h1{ color:red;}
</style>
<body>
<h1>I would thanks to Mr. Vimal Daga Sir for giving thsi wonderful task as this task is
so insightful and helps me in understanding ansible more clearly.</h1>
</body>
</html>
```

4. Now the most important thing left, which is playbook. As Ansible uses Playbook to perform the tasks sent from the Controller node to the Target Node.

So will write yml file used by ansible

`vim preq.yml`



```
root@localhost:/wstask10 x root@localhost:/ws11.3
- hosts: all
  tasks:
    - copy:
        dest: "/etc/yum.repos.d/docker.repo"
        src: "docker.repo"

    - package:
        name: "docker-ce-18.09.1-3.el7.x86_64"
        state: present

    - service:
        name: "docker"
        state: started
        enabled: yes

    - pip:
        name: docker

    - name: "pulling docker image"
      docker_image:
        name: "httpd"
        source: pull
```

In this first we Copied the repo file of docker.

Then Confirm for the docker image to be pulled into that repo location.

After that starting the docker services.

Then finally pull docker image.

After this,

```

- name: "creating Docker container"
  community.general.docker_container:
    name: "honey"
    image: "httpd"
    state: started
    exposed_ports:
      - "80"
    ports:
      - "4001:80"

- name: "Allowing firewall for new port no."
  firewallld:
    port: "4001/tcp"
    state: enabled
    permanent: yes
    immediate: yes

- name: "Allowing firewall for new port no."
  firewallld:
    port: "80/tcp"
    state: enabled
    permanent: yes
    immediate: yes

```

After this we created Docker container with some syntax of Ansible.

Then, Firewall configuration for both port and exposed ports

```

- copy:
  dest: "/var/www/html"
  src: "task.html"

```

Finally I copied the html file in the Document root and I also shown the content of webpage above.

Now, here is the list of all the files in my workspace of this task.

```

[root@localhost wstask10]# ls
docker.repo  preq.yml  task.html
[root@localhost wstask10]#

```

Finally, We have to Run the Playbook and this can be done by

Ansible-playbook <filename>.yml

Anyone can also use verbose(-v) for more details in between filename and playbook.

So here is the result of running playbook.

```
[root@localhost wstask10]# ansible-playbook preq.yml
[DEPRECATION WARNING]: The firewall module has been moved to the ansible.posix
collection. This feature will be removed from community.general in version 2.0.0.
Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.43.174]

TASK [copy] *****
ok: [192.168.43.174]

TASK [package] *****
ok: [192.168.43.174]

TASK [service] *****
ok: [192.168.43.174]

TASK [pip] *****
ok: [192.168.43.174]

TASK [pulling docker image] *****
ok: [192.168.43.174]

TASK [creating Docker container] *****
[DEPRECATION WARNING]: The container_default_behavior option will change its default
value from "compatibility" to "no_defaults" in community.general 3.0.0. To remove this
warning, please specify an explicit value for it now. This feature will be removed
from community.general in version 3.0.0. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
changed: [192.168.43.174]

TASK [Allowing firewall for new port no.] *****
ok: [192.168.43.174]

TASK [Allowing firewall for new port no.] *****
ok: [192.168.43.174]

TASK [copy] *****
ok: [192.168.43.174]

PLAY RECAP *****
192.168.43.174 : ok=10  changed=1  unreachable=0  failed=0  skipped=
0  rescued=0  ignored=0
```

Finally Our work for control Node is over.

Now we can go to Target Nodes and check about the Port numbers running their. Webpage copied in document root or not and many more which is needed to verify you.

Just use

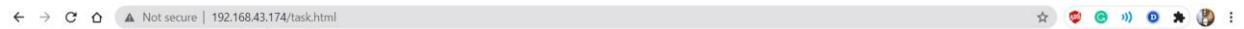
`tcpdump -tnlp`

to catch all the running ports.

I am running the Webpage hosted by target node. So I will use

`IP/filename` in the browser.

Here, 192.168.43.174/task.html



I would thanks to Mr. Vimal DAga Sir for giving thsi wonderful task as this task is so insightful and helps me in understanding ansible more clearly.