Automated Container Deployment and Administration CA

B9IS121 Network Systems and Administration 2024

Collins Kipngetich Student No: 20033523

Summary

This report shows how by use of Ansible, Apache Docker can be efficiently deployed. A public GitHub repository was created to store the files that were used in the project. Using the mentioned tools has shown how deployment can be more consistent and faster. This report will show the entire process of setting up Ansible and Apache Docker deployment.

Contents

1. Introduction

This project involves deploying Docker containers using Ansible Playbook. Below are prerequisites that were need for the project.

Requirements

Creating a public GitHub repository for saving the project. (https://github.com/CollinsDBS/Networking-CA)

Installing Ansible and creating an Ansible Playbook for the project.

Installing and Deploying Docker container using Ansible playbook and running Apache service.

2. Implementation

Task 1: GitHub Repository Setup

On my GitHub account (https://github.com/CollinsDBS) I create a new repository 'Networking-CA' where I would store al the project files. I then added a README.md file with the specifications of the project.

Task 2: Setting Up Ansible

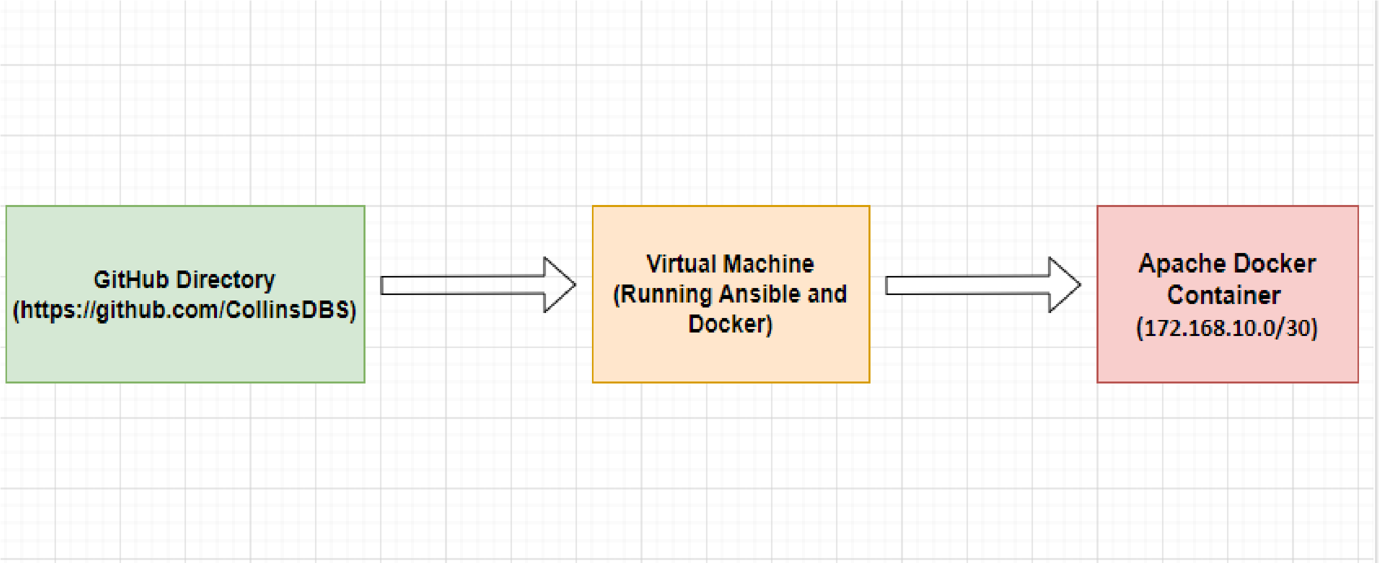
Using Windows Subsystem for Linux I cloned the 'Networking-CA', where I worked in. I then configured an Ansible playbook using 'nano' to type the commands of the hosts and tasks used in the project into the .yml file. The procedure of installing and setting up Docker network, with Apache container running. I then created an inventory file that describes the devices Ansible will connect to using the IP address allocated.

Task 3: Deploying Docker Container

The playbook executed and the Apache Docker container deployed which run on the specified Docker network on '172.168.10.0/30' subnet. This was be checked by typing the IP address on the browser to get the static web page that was on the repository. The exercise executed as expected and we got successful results.

3. Network Diagram

Below is a diagram that shows the connectivity between the GitHub repository, host machine and the Apache Docker Container that was utilized in this project.


GitHub Directory (https://github.com/CollinsDBS)
Virtual Machine (Running Ansible and Docker)
Apache Docker Container (172.168.10.0/30)

Conclusions

Deployment of the Apache Docker container using Ansible was successful. Automation of Deployment using Docker and Ansible seems to enhance productivity and reduces errors during releases. Docker which is renowned for containerization partnered with Ansible for automation creates an efficient solution for managing Docker containerization.

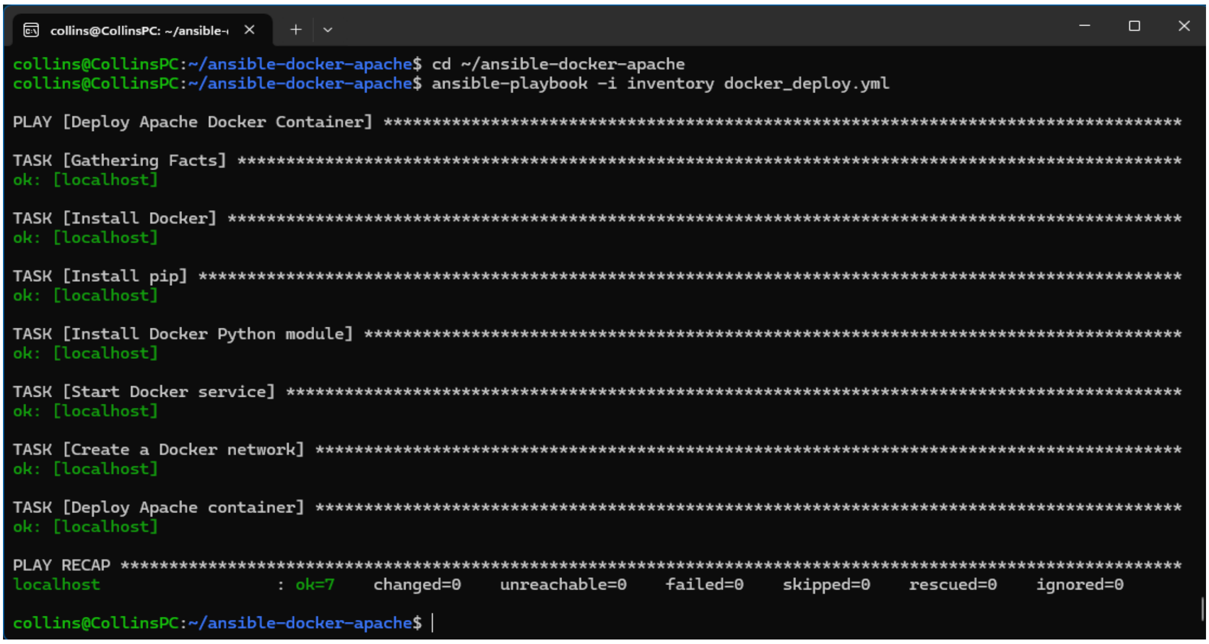
https://github.com/CollinsDBS/Networking-CA

References

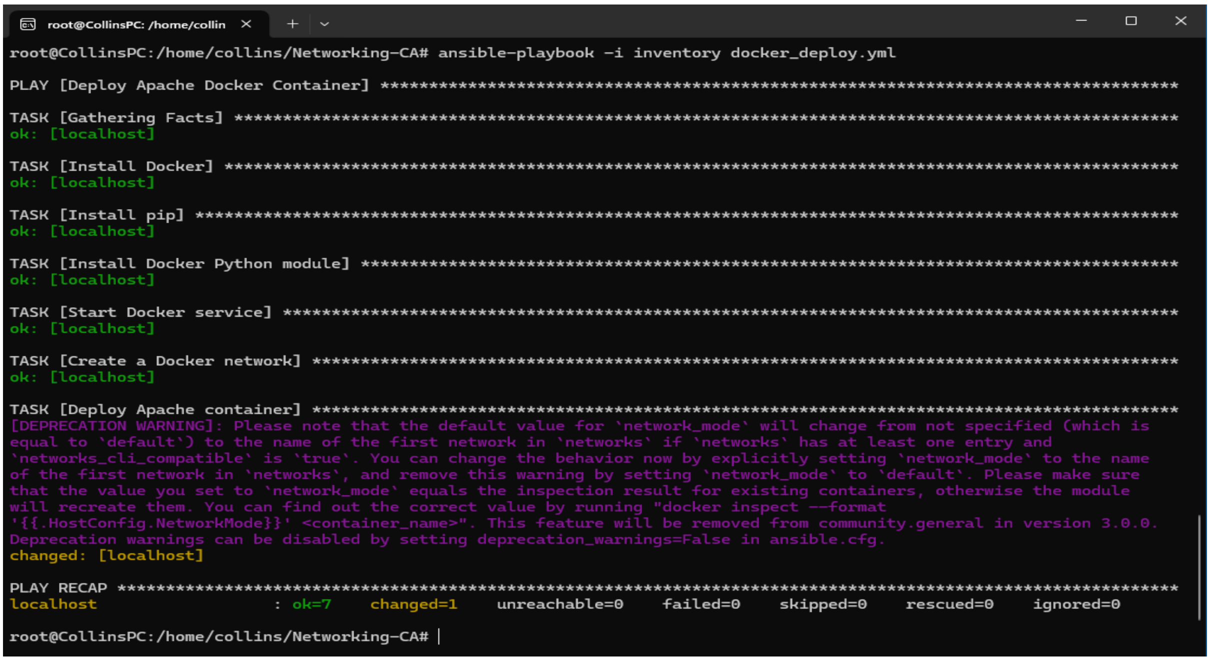
Jangid, A., 2023. *Integrating Ansible and Docker*. Medium. Available at: <https://medium.com/@aakashjangidme/integrating-ansible-and-docker-5e05f0de7305#b296-7502c6d5b575> [Accessed 16 June 2024].

Naik, V., 2023. *Configure Apache HTTPD Web Server on Docker using Ansible Playbook*. Medium. Available at: <https://medium.com/@vamsinaik452/configure-apache-httpd-web-server-on-docker-using-ansible-playbook-dfcdbe42decb> [Accessed 16 June 2024].

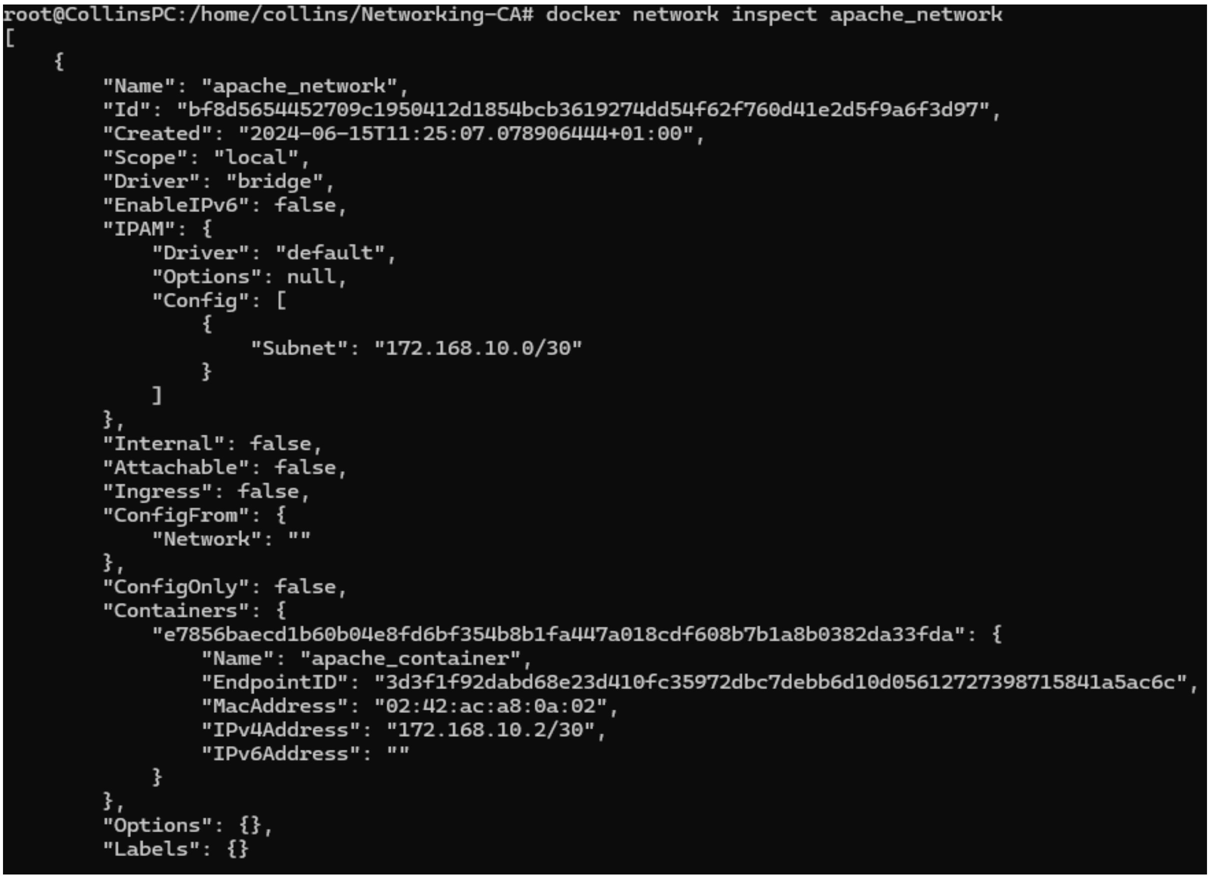
Appendices



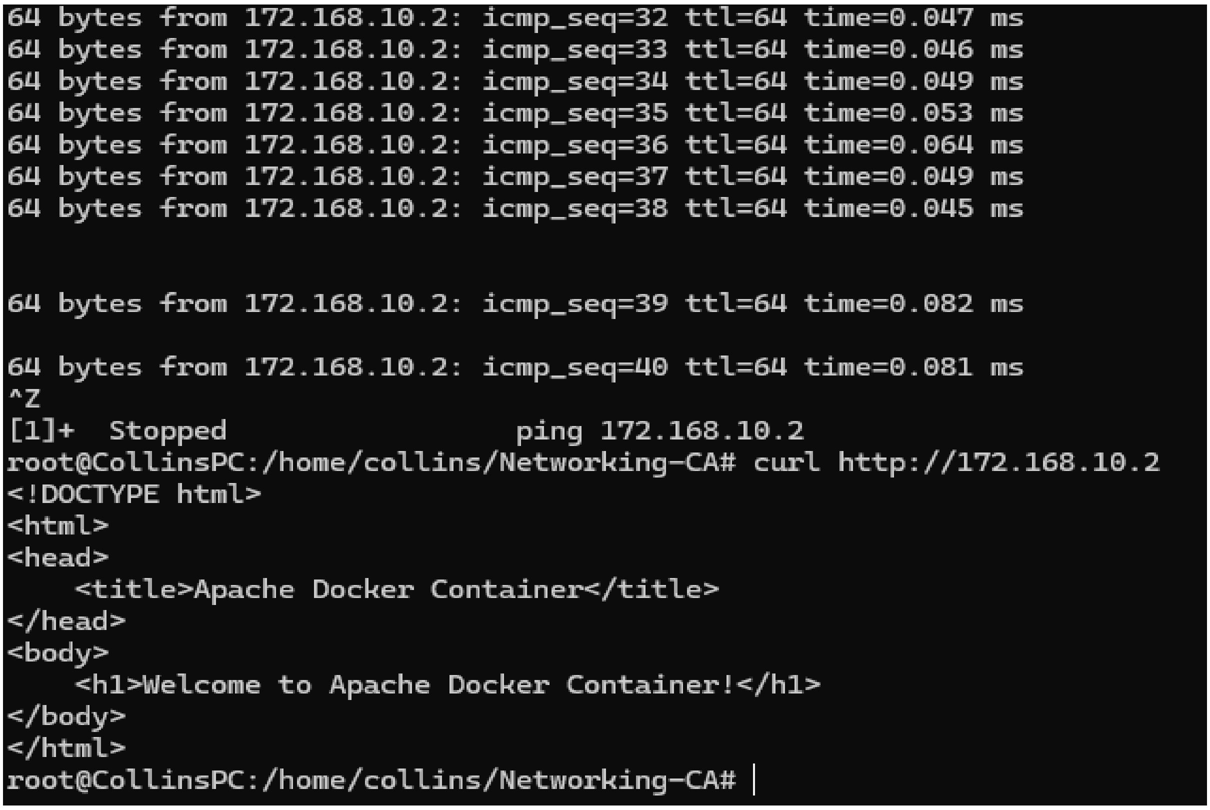
```
collins@CollinsPC:~/ansible-docker-apache$ cd ~/ansible-docker-apache
collins@CollinsPC:~/ansible-docker-apache$ ansible-playbook -i inventory docker_deploy.yml

PLAY [Deploy Apache Docker Container] *******************************************************************************

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

TASK [Install Docker] ***********************************************************************************************
ok: [localhost]

TASK [Install pip] **************************************************************************************************
ok: [localhost]

TASK [Install Docker Python module] *********************************************************************************
ok: [localhost]

TASK [Start Docker service] *****************************************************************************************
ok: [localhost]

TASK [Create a Docker network] **************************************************************************************
ok: [localhost]

TASK [Deploy Apache container] **************************************************************************************
ok: [localhost]

PLAY RECAP **********************************************************************************************************
localhost                  : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

collins@CollinsPC:~/ansible-docker-apache$ |
```



```
root@CollinsPC:/home/collins/Networking-CA# ansible-playbook -i inventory docker_deploy.yml

PLAY [Deploy Apache Docker Container] *******************************************************************************

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

TASK [Install Docker] ***********************************************************************************************
ok: [localhost]

TASK [Install pip] **************************************************************************************************
ok: [localhost]

TASK [Install Docker Python module] *********************************************************************************
ok: [localhost]

TASK [Start Docker service] *****************************************************************************************
ok: [localhost]

TASK [Create a Docker network] **************************************************************************************
ok: [localhost]

TASK [Deploy Apache container] **************************************************************************************
[DEPRECATION WARNING]: Please note that the default value for `network_mode` will change from not specified (which is
equal to `default`) to the name of the first network in `networks` if `networks` has at least one entry and
`networks_cli_compatible` is `true`. You can change the behavior now by explicitly setting `network_mode` to the name
of the first network in `networks`, and remove this warning by setting `network_mode` to `default`. Please make sure
that the value you set to `network_mode` equals the inspection result for existing containers, otherwise the module
will recreate them. You can find out the correct value by running "docker inspect --format
'{{.HostConfig.NetworkMode}}' <container_name>". 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: [localhost]

PLAY RECAP **********************************************************************************************************
localhost                  : ok=7    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@CollinsPC:/home/collins/Networking-CA# |
```



```
root@CollinsPC:/home/collins/Networking-CA# docker network inspect apache_network
[
    {
        "Name": "apache_network",
        "Id": "bf8d5654452709c1950412d1854bcb3619274dd54f62f760d41e2d5f9a6f3d97",
        "Created": "2024-06-15T11:25:07.078906444+01:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.168.10.0/30"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "e7856baecd1b60b04e8fd6bf354b8b1fa447a018cdf608b7b1a8b0382da33fda": {
                "Name": "apache_container",
                "EndpointID": "3d3f1f92dabd68e23d410fc35972dbc7debb6d10d0561272739871584 1a5ac6c",
                "MacAddress": "02:42:ac:a8:0a:02",
                "IPv4Address": "172.168.10.2/30",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
```



```
64 bytes from 172.168.10.2: icmp_seq=32 ttl=64 time=0.047 ms
64 bytes from 172.168.10.2: icmp_seq=33 ttl=64 time=0.046 ms
64 bytes from 172.168.10.2: icmp_seq=34 ttl=64 time=0.049 ms
64 bytes from 172.168.10.2: icmp_seq=35 ttl=64 time=0.053 ms
64 bytes from 172.168.10.2: icmp_seq=36 ttl=64 time=0.064 ms
64 bytes from 172.168.10.2: icmp_seq=37 ttl=64 time=0.049 ms
64 bytes from 172.168.10.2: icmp_seq=38 ttl=64 time=0.045 ms


64 bytes from 172.168.10.2: icmp_seq=39 ttl=64 time=0.082 ms

64 bytes from 172.168.10.2: icmp_seq=40 ttl=64 time=0.081 ms
^Z
[1]+  Stopped                 ping 172.168.10.2
root@CollinsPC:/home/collins/Networking-CA# curl http://172.168.10.2
<!DOCTYPE html>
<html>
<head>
    <title>Apache Docker Container</title>
</head>
<body>
    <h1>Welcome to Apache Docker Container!</h1>
</body>
</html>
root@CollinsPC:/home/collins/Networking-CA# |
```