

# B9IS121 - Network Systems and Administration CA 2024

Done by: Gayathri devi Dhamodharan. (20031606)

## Automated Container deployment and Administration

Assignment Cover Sheet

Lecturer’s Name: Kingsley Ibomo

Programme (degree): MSc in Cybersecurity

Year: 2024 - 2025

Module: Network Systems and Administration

Submission Date: 17/03/2024

Title of Assignment: Automated Container deployment and Administration

Declaration of Ownership: We declare that the attached work is entirely our own and that all sources have been acknowledged:



Date: 17/03/2024

# List of Contents

#### Introduction

#### Objective

#### Methodology

#### GitHub repository creation

#### Deployment of docker container using Ansible

#### Network configuration.

#### Network Diagram

1. **Results**
2. **Conclusion**
3. **Reference**

# Deployment of Docker Containers using Ansible

1. **INTRODUCTION**

The purpose of this project was to demonstrate proficiency in deploying Docker containers using Ansible, an automation tool. This report details the steps taken to achieve this objective, including the creation of a template file stored in GitHub, deployment of a Docker container running Apache service with a static web page, configuration of network communication, and verification of accessibility from the host machine.

1. **OBJECTIVE:**

The central aim of this assessment was to assess the comprehension and practical implementation of Docker container deployment utilizing Ansible. The assigned tasks entailed several specific steps:

Establishing a public GitHub repository dedicated to assessment.

Employing Ansible to deploy a Docker container equipped with an Apache service and a static web page.

Configuring the container to facilitate network communication and conducting checks to ensure accessibility from the host machine.

Developing a detailed network diagram illustrating the deployment scenario, specifically depicting the container operating within the 172.168.10.0/30 subnet and facilitating traffic flow from the host machine to the container subnet.

1. **METHODOLOGY**

##### **GitHub Repository Creation:**

A publicly accessible GitHub repository was established to serve as a centralized storage space for housing both the project files and the Ansible playbook. This repository served multiple purposes: foremost, it facilitated version control, allowing for the tracking of changes and revisions made to the project over time. Additionally, it served as a collaborative platform, enabling team members to work together on the project, share code, and contribute to its development in a coordinated manner.

A screenshot of a computer

Description automatically generated

##### **Deployment of Docker Container using Ansible:**

Ansible, a widely used open-source automation tool, played a pivotal role in deploying a Docker container hosting an Apache service. This tool streamlines various IT tasks such as orchestration, configuration management, and application deployment by automating repetitive processes, thus enhancing efficiency and reliability in system management. Following the installation of Ansible, an inventory was configured to list the target IP addresses required for deployment.

Additionally, an Ansible playbook, composed in YAML syntax, was employed to delineate the sequential deployment procedures and configurations. These playbooks offer a declarative approach for articulating the desired state of systems, thereby simplifying the management of both infrastructure and applications. Specifically, a playbook named "ansible\_playbook kk.yml" was crafted, containing instructions tailored to the project's requirements.

A computer screen with text and numbers

Description automatically generated

- name: Deploy Apache container

hosts: 172.168.10.1 # Target host

become: true

tasks:

# Task 1: Install Docker

- name: Install docker package

apt:

name: docker.io

state: present

# Task 2: Pull Apache image

- name: Pull apache image

become: true

docker\_image:

name: httpd:latest

source: pull

tags: docker

# Task 3: Copy HTML file to the container

- name: Copy HTML file to the container

become: true

copy:

src: /home/gayathri/sample.html

dest: /var/www/html/index.html

owner: root

group: root

mode: '0644'

# Task 4: Run Apache container

- name: Run Apache container

become: true

docker\_container:

name: apache\_container

image: httpd:latest

ports:

- "80:80"

state: started

restart\_policy: always

tags: docker

# Task 5: Run Apache container with mounted volume

- name: Run Apache container with mounted volume

become: true

docker\_container:

name: apache\_container\_mounted

image: httpd:latest

ports:

- "80:80"

volumes:

- "/var/www/html:/usr/local/apache2/htdocs/"

tags: docker

Tasks were incorporated into the playbook to install Docker, pull the Apache image, and initiate the container. After saving the "deploy\_apache.yml" file, the playbook was executed using the following command.

A screen shot of a computer

Description automatically generated

##### **Network Configuration:**

Network configurations were meticulously adjusted to facilitate seamless communication between the host machine and the Docker container, thereby guaranteeing unfettered accessibility. This involved fine-tuning various parameters within the network infrastructure to establish a robust and reliable connection pathway, ensuring efficient data exchange and interaction between the host and the container.

A computer code on a black background

Description automatically generated

A screenshot of a computer

Description automatically generated

##### **Network Diagram:**

To ensure a thorough understanding of the deployment setup, a detailed network diagram was meticulously crafted. This diagram serves as a visual aid, providing a clear representation of the deployment scenario. It highlights the flow of traffic within the network and elucidates the configuration of subnets. By depicting the various components and their interconnections, the network diagram offers valuable insights into the architecture and connectivity of the deployed system.

A diagram of a diagram

Description automatically generated

1. **RESULTS**  
     
   The deployment procedure was effectively carried out, successfully achieving all predefined project objectives. Utilizing Ansible automation, Docker containers were deployed to host Apache services, demonstrating seamless integration and deployment capabilities. Furthermore, meticulous network configuration facilitated uninterrupted communication channels between the host machine and the deployed Docker containers. Rigorous verification procedures were conducted to ensure accessibility to the Apache service, thereby affirming the efficacy and reliability of the deployment process.

**GITHUB link :**

1. **CONCLUSION**

In summary, this project offered significant practical insight into the deployment of Docker containers using Ansible, an automation solution renowned for its capacity to streamline various IT tasks. Through the utilization of Ansible playbooks scripted in YAML syntax, in conjunction with Docker containerization technology, the deployment workflow was optimized, resulting in enhanced efficiency and effectiveness. Moreover, this endeavor underscored the critical role of automation tools in contemporary IT landscapes, highlighting their ability to expedite the deployment and administration of containerized applications, thereby fostering agility and scalability in operational environments.

1. **REFERENCES**

Ansible Documentation: https://docs.ansible.com/ansible/latest/index.html

Docker Documentation: https://docs.docker.com/

GitHub Guides: https://guides.github.com/