**Ansible Deployment of Docker Containers with Apache**

**Lecturer-Kingsley Lbomo**

**Course-Network System and Administration CA 2024**

**(B9IS121)**

**Submitted By:**

**Sarthak Singhal**

**20026292**

**Submission Date:**

**17th March 2024**

**Summary:**

This report traces the handle of conveying Docker holders utilizing Ansible to run Apache administrations with a inactive web page. The report incorporates the setup of a GitHub store, creation of an Ansible playbook, sending of Apache Docker holder, arrangement of organizing for the holder, and creation of a organize graph. The confirmation of effective sending and openness of the Apache benefit is moreover talked about. The report concludes with a outline of the assignments completed and screenshot prove of confirmation comes about. All things considered, the "Outline" segment serves as a speedy diagram of the report's substance, empowering peruses to rapidly get it the key ideas.

**Table of Content**

1. Introduction
2. Lab Setup
3. GitHub Repository Setup
4. Ansible Playbook Creation
5. Sending Apache Docker Container
6. Arranging Organizing for Apache Container
7. Creating Network Diagram
8. Conclusions
9. References

**Introduction:**

The objective of this appraisal is to illustrate the sending of Docker holders utilizing Ansible for running Apache administrations. The assignments include setting up a GitHub store, making an Ansible playbook, conveying Apache Docker holder, designing organizing, making a organize graph, and confirming the availability of the Apache service.

1. Introduce Docker: To make beyond any doubt the Docker daemon is accessible, Ansible will introduce Docker on the target have or hosts.
2. Drag Docker pictures: If fundamental, Ansible will create the important Docker pictures locally or drag them from a registry.
3. Construct Docker holders: Ansible will utilize the pictures to construct the holders and design them as needed.
4. Begin Docker holders: Ansible will dispatch the holders and make beyond any doubt they are reachable and operational.
5. Check for indicated yield: Ansible will perform tests or see for particular yield to affirm that the Docker holders are working appropriately.

**Get UTM** –

Ubuntu 23.10 here.

The most recent iteration of the Ubuntu operating system (https://ubuntu.com) serves as a Windows emulator and virtual machine host. In essence, UTM lets you use your Mac, iPhone, and iPad to run Windows, Linux, Ubuntu, and other operating systems.

We'll use Ubuntu 22.04.4 LTS as our virtual environment at UTM.

**Set up an SSH server on Ubuntu:**

Secure Shell, or SSH for short, is a method of safely gaining network access to a distant PC or server. You can use it to make sure that, even while connected to a computer on the other side of the globe, your communication remains private and safe (cyberciti.biz, 2018).

In Ubuntu Terminal, install SSH server by running by running the following command:

Sudo opt-get install open SSH server

To enable ssh server, type

Sudo systemctl enable ssh

Finally, start the SSH service by running

Sudo systemctl start ssh

Currently, Ubuntu and a Windows computer are virtually connected.

**Lab Setup:**

- GitHub repository created for storing deployment artifacts

- Docker installed on target machine(s)

- Ansible installed on the control machine

**GitHub Repository Setup:**

We created the "Software Network" public GitHub repository and added a README.md file to it. This repository contained the "new Ansible playbook" Ansible playbook. The first step towards collaborative work and version control is creating a GitHub repository. As part of the project, we established the "Software Network" repository and filled it with necessary project documentation by starting with a README.md file. The "new Ansible playbook" Ansible playbook, which contains the deployment logic for Apache Docker containers, is hosted in this repository.

To guarantee flawless execution, there are multiple processes involved in crafting the Ansible playbook. SSH pass is first installed on the Windows computer in order to enable safe system-to-system communication. Afterwards, host key checking is turned off in the Ansible configuration file to reduce the possibility of authentication problems. The playbook itself provides a complete automation framework by outlining operations from Docker installation to container configuration in YAML format.

A close-up of a logo

Description automatically generated

**Ansible Playbook Creation:**

The Ansible playbook was structured with hosts, tasks, and handlers sections to facilitate the deployment of Docker containers with Apache services. Relevant tasks were defined for each stage of the deployment process.

Create the nano ansible.cfg configuration file for Ansible.

Add the line that follows to the configuration file:

[Default]

False for host\_key\_checking

Since host key checking is set to false in this case, Ansible won't verify the remote server's host key.

Make an Ansible playbook by running the command below:

nanoscale newansible.yml <— Our playbook is named newansible.yml.

We will develop our script to automate Ubuntu using Docker in this playbook.

The command to launch the playbook is as follows: ansible-playbook newansible.yml.

# **Install Ansible on Windows Using Cygwin**

## **1. Install Cygwin**

Go to <https://www.cygwin.com/install.html> and download setup-x86\_64.exe Run it

* Select 'Install from Internet'
* Choose root directory (default)
* Choose the package directory to store installation files for the dropdown
* Use System Proxy Settings
* Select any mirror site to download

### In 'Select Packages'

* Select Category dropdown and search for lynx
* Go to All -> Web -> lynx: A text-based Web Browser
* Select latest version
* Click next to complete installation

## **2. Install apt-cyg**

This is a package manager for cygwin

To install:

* lynx -source rawgit.com/transcode-open/apt-cyg/master/apt-cyg > apt-cyg
* install apt-cyg /bin

## **3. Install dependencies for ansible**

To install altogether

* apt-cyg install binutils curl gcc-core gmp libffi-devel libgmp-devel make python python-crypto python-openssl python-setuptools python-devel git nano openssh openssl openssl-devel

You can also install individually by using

* apt-cyg install or by using the installation GUI

## **4. Install Ansible**

To install

* easy\_install-2.7 pip
* pip install ansible -vvv

## **5. Test ansible**

To test installation

* ansible
* Should receive list of options

# Some configuration

Cygwin has some strange default settings, if we are using vi or vim.

Open a .virc file.

* vi ~/.virc

Type in

set nocompatible

set backspace=2

## Run a playbook with Ansible

We have to make new directory with the name ansible playbook.

mkdir ansible-lab  
cd ansible-lab

Ansible works on a different configuration for windows. Our name machine is called the control machine. Ansible do not work on windows machine so we need a configuration file.

Create an ansible configuration file by using below command =

vi ansible.cfg

upload below code in configuration file.

[ssh\_connection]  
ssh\_args = -o ControlMaster=no

Now we have created our ansible playbook to test.

vi playbook.yml

upload below code in following into the file.

- hosts: all

sudo: yes

tasks:

- name: install apache2

apt: name=apache2 update\_cache=yes state=latest

**Sending Apache Docker Container:**

The playbook contained a task to use the Docker container Ansible module to launch an Apache Docker container. To operate the Apache service with a static web page, the container was configured using the Apache image.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Arranging Networking for Apache Container**

A screen shot of a computer

Description automatically generatedThe configuration of networking for the Apache Docker container was designated as separate task. The host machine configured the container to operate on the 172.168.10.0/30 subnet in order to facilitate network connectivity and guarantee accessibility

**Creating Network Diagram:**

Using the relevant tools, a network diagram was made to show the host computer, the GitHub repository, the Apache Docker container with its designated subnet, and the connections between them. The diagram gave the deployment environment a visual representation.

Ubuntu Host Machine

IP Subnet

172.168.10.1/30

Docker Host

(Ubuntu Target)

172.168.10.2/30

Docker Container

(Apache Web Server)

172.168.10.1

Port 80 (HTTP)

**Conclusions:**

In conclusion, it was successful to deploy Docker containers using Ansible to run Apache services. The assignments were finished in accordance with the specifications, and the Apache service's accessibility was confirmed. Ansible's adoption expedited the deployment procedure and guaranteed uniform setup across many environments**.**

**References:**

- Provide references for any materials or resources that were used to prepare this assessment.

<https://www.ansible.com/overview/how-ansible-works>

<https://devdocs.io/ansible/>

<https://github.com/ansible/ansible-documentation>

<https://ubuntu.com/download>

<https://github.com/LMtx/ansible-lab-docker>

<https://gcore.com/learning/integrating-ansible-and-docker-in-ci-cd-process-using-jenkins-job/>

<https://medium.com/@mdarsuhack/configure-the-apache-web-server-on-the-docker-container-6be575cf1ba9>