1. Create a Docker Container in MSR-test-Instance-1 using Docker Compose file and ensure apache webserver is installed. Try to use configuration management tools to automate the entire installation of apache and deploy a sample html file from a GitHub repository.

**Additional Information**

* You can create your own GitHub repository with a sample html file.

Ans:

Treee

.

├── dockercompose

│   ├── docker-compose.yaml

│   └── webapp

│   ├── Dockerfile

│   └── sample.html

├── get-docker.sh

├── playbook.retry

├── playbook.yaml

└── sample.html

3).

sudo vim playbook.yaml

---

- name: set of tasks

hosts: localhost

become: yes

connection: local

gather\_facts: no

tasks:

- name: docker compose

file:

name: ./dockercompose

state: directory

mode: 0755

- name: create webapp dir in dc

file:

name: ./dockercompose/webapp

state: directory

mode: 0755

- name: git clone

git:

repo: https://github.com/satish916/DevOps-Test.git

clone: yes

dest: ./dockercompose/webapp

- name: change directory

shell: cd /home/ubuntu/dockercompose;docker-compose build;docker-compose up -d

...

2)Create Dockerfile In webapp:

$cd dockercompose/webapp

$sudo vim Dockerfile

FROM ubuntu

RUN apt-get update \

&& apt-get install -y apache2

COPY sample.html /var/www/html/

WORKDIR /var/www/html

CMD ["apachectl", "-D", "FOREGROUND"]

EXPOSE 80

save&quit

3)create docker-compose.yaml in dockercompose dir:

cd /home/ubuntu/dockercompose

sudo vim docker-compose.yaml

version: '2'

services:

web:

image: apache

build: ./webapp

container\_name: apache\_web

restart: always

ports:

- "8080:80"

save&quit