Frankfurt University of Applied Sciences

–Faculty of Computer Science and Engineering–

HIS Project Open Twin Deployment

Master of Science (B.Sc.)
High Integrity Systems(HIS)

Presented By

Sohail Dua

Marticulation Number: 1322512

Submitted To

Prof. Peter Thoma

DECLARATION

I hereby confirm that I have written the present work independently and have not used any other sources than those given in the bibliography.

All passages taken literally or in spirit from published or as yet unpublished sources are identified as such.

Drawings or illustrations in this paper have been prepared by me or have been appropriately referenced.

This work has not been submitted in the same or similar form to any other examining authority.

Frankfurt, 15. March 2021

Sohail Dua

ABSTRACT

This report researches about the Development environment and deployment of Open Twin project using Docker and bat script. The main objective is to create a Dockerfile, which forms the base image for the deployment. The image is then used to deploy and copy the base files of deployment using windows docker container.Although automation is not included in this report, the basis is made from which the creation of the automated deployment pipeline can be started.

Open Twin is an Open Source software that is being developed at Frankfurt University of Applied Sciences to support the industrial digital transformation. In particular, the software provides a powerful and scalable infrastructure for virtual prototyping by using physics-based simulation tools.

Docker is a container technology and is also called as container-based virtualization which has multiple instances on operating system running over a single kernel. Here the operating system's kernel runs on the underlying hardware with isolated virtual machines called as containers. Microservice architecture is not a new thing but started getting attention when the docker was introduced and many companies around the world are shifting from a standalone architecture to micro-service architecture. With lot of pros, Docker is a very good fit for micro- service architecture application.

The goal of this report is to find a quick and efficient way to deploy new versions of the application in a test- and potentially a production environment. The research for this report conducted from a practical viewpoint.

Keywords : Open Twin, Docker , Containers , Deployment ,Windows

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

VM	Virtual Machine
OS	Operating System
YAML	Yet Another Markup Language
HPC	High Performance Computing
AWS	Amazon Web Services

Part I

REPORT

1 INTRODUCTION

1.1 MOTIVATION

In the past few decades, the use of scientific software has been increased dramatically to advance science and engineering. The advancement in software engineering catalyzes the accuracy in results produced by software. Traditional scientific and software engineering software had only a few dependencies and were mostly developed by small groups of researchers in both stable and well-established programming languages. However, modern software and tools are developed by a large group of people using a variety of languages and tools.

Agile software development has been used in companies for years now. According to the survey ,The 14th annual State of Agile Report [2] 59% of the participating companies use this approach to speed up their product delivery chain and increase team productivity.This methodology has procedures ,methods that need to be used for a particular task.The agile also helps in choosing methods and procedure to achieve the agility.Agility mean that we are able to adapt with the changes that are required in the software development scheme and also making further improvements.To establish agile processes in the enterprise, containerization is often used [3]

For containerization we use Docker containers as lightweight software components that bundle the application, its dependencies, and its configuration in a single image, running in isolated user environments on a traditional operating system on a traditional server or in a virtualized environment.

Why Docker and Agile are a "Good Couple"?. Docker and agile both can work in parallel and deploy the code in a faster as well as in an automated way.The Lack of communication between the teams working together decreases as same data is shared.Agile focuses on Software development whereas Docker Culture focuses on how the created software spans efficiently on the number of servers with Docker containers.With the integration of Agile Methodology and Docker Culture, now organizations are doing rapid deployments every day without any worry about the hardware resources [4].

2 PROBLEM STATEMENT

Containers are very new technology that is made to address the issues that were present in the servers such as resource utilization.The performance factor acts as an advantage as compared to VM which always have a usage issue.This report aims at using the docker container for the deployment of the Open Twin simulation project.As a result the following questions will be answered:

1.What is Docker?

2.How data on docker can be stored?

3.How we can store and run docker data anywhere?

4.How to deploy docker data on cloud?

2.1 PROBLEMATIZATION

Because containers are becoming more frequently used amongst people working in IT, several different container software have been developed. Thus, the scope of this thesis is limited to one of the most prominently used container software called Docker and its deployment

3

THEORY

The main point is here to deploy our application on docker. So to do that we need to have a background about the how docker works. This chapter aims to provide knowledge of what Docker containers are, how they function, and related tools that will be used to examine them.

3.1 WHY USE DOCKER?

- Runs on my machine = runs anywhere:-

  your app is dockerized and it runs with out any error on your machine.Then 99% of the times the app will run fine anywhere.For example if your tried it on your staging environment and it will work same as on production environment too.

- New team member can be productive from day 1:-

  In a organisation whenever a new person joins a team so to setup the system it generally takes 2-3 days. With app deployed on docker a new person who joined can set up the development environment in some commands.

- Test app's compatibility with the newer version of language/database Whenever a new version of programming language is launch many things are added but some are deprecated plus the new version also provides stability.The team in a organisation can test by just replicating the container with new version of programming language.

3.2 DOCKER OR DOCKER ENGINE

Docker Engine is the industry's de facto container runtime that runs on various Linux (CentOS, Debian, Fedora, Oracle Linux, RHEL, SUSE, and Ubuntu) and Windows Server operating systems [3].

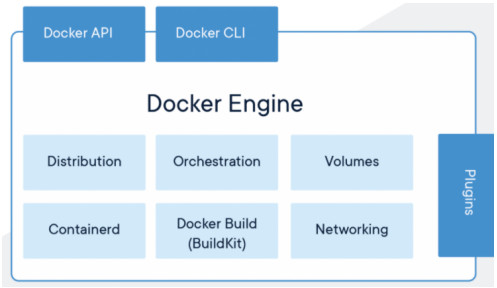


Figure 3.1: Docker Engine

Docker creates a simple tool and a universal packaging approach that combine up all the dependencies of a application inside a container which is then run on docker Engine.Docker Engine enables containerized applications where we can run consistently on any infrastructure, solving "dependency hell" for developers and operations teams.Docker engine relies on images and containers

3.3 DOCKER IMAGES

Docker image is a file that has data and information needed to create a group of processes with properties well defined.It has only read only layers.Each layers is created with each command gets executed.We can use the images that already built and are on the docker hub or we can make own custom images for our own purpose.The commands in the image corresponds to anything including creation of folder , copying , installing dependencies or configuring network ports.When executed, a well-isolated environment called a container is created based on the specifications of the image file. [8]

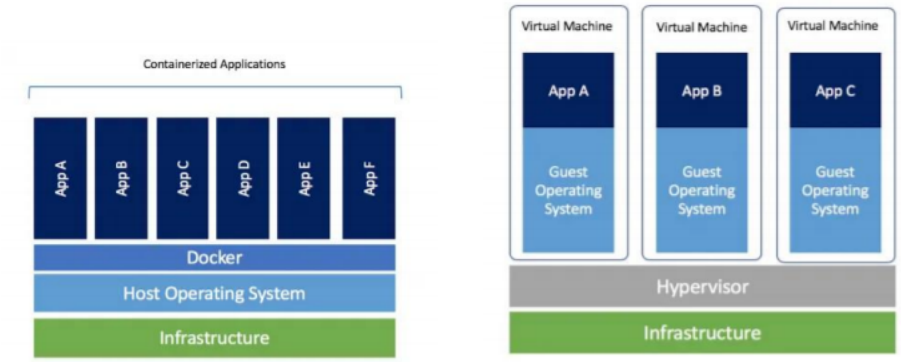
Docker images can create identical environments on different OSs which make it easily shareable.The benefit of being able to run a given container on any OS is that applications on different systems will not encounter any compatibility issues since they are running inside the isolated environment created from the docker image [8].Images can exist without containers, whereas a container needs to run an image to exist. Therefore, containers are dependent on images and use them to construct a run-time environment and run an application

3.4 DOCKER CONTAINERS

Containers are quite similar to VMs , as both allow varying type of softwares to run in isolated environments , but their functioning is totally different.A container is totally a isolated environment , we can say that read and writable layer created on top of the read-only layers of the image.These are called container layers and are located in C:\ProgramData\DockerDesktop directory.The container layer saves all the changes that are made to running container such as creating or deleting files .If you delete the container the writable layers are also removed while the underlying image remains same.

The containers involves grouping of application with all the related data, libraries and dependencies .This makes container very lightweight as compared to VM which has all the dependencies plus the OS installed which is a big difference between VM and containers.As shown in figure 3.2 , several VM run on same host with hypervisor technology .So containers communicate with the OS can have advantages and as well as disadvantages .The advantages are lower storage space usage and performance since the container does not have to load an entire OS. One disadvantage, however, is

that sharing an OS with the host is a concern from a security perspective as this makes containers less isolated than a VM. [8] [1]



(a) Containers (b) Virtual Machines

Figure 3.2: Comparison between container and virtual machine structure [7]

3.5 DOCKERFILE

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image. Using docker build users can create an automated build that executes several command-line instructions in succession [5].

The advantage of Dockerfile is that your application will be updated with latest version present now.This is a good thing it also ensure you are not using or installing any vunerable software keeping it secure

In docker fine we can do many task such as:

- To run commands such as delete or copy commands
- To add file or create directory
- To install any dependency
- To create envoirnment variable

To build dockerfile we use **docker build** command.Then then a container is created and then we run this container using **docker run** command

3.6 DOCKER COMPOSE

Docker compose is used for multiple containers to be a single entity.Suppose we have containers to connect them we have to do it manually.We have to

take care of dependencies.As Docker compose provides in connection between the containers.Docker compose files are very easy to write as we use a scripting language called YAML which is a XML-based language and it stands for Yet Another Markup Language.Another thing is Docker compose user can run all services with a single command.

For example:

If you have an application that requires an NGINX server and Redis database, you can create a Docker Compose file that can run both the containers as a service without the need to start each one separately.[11]

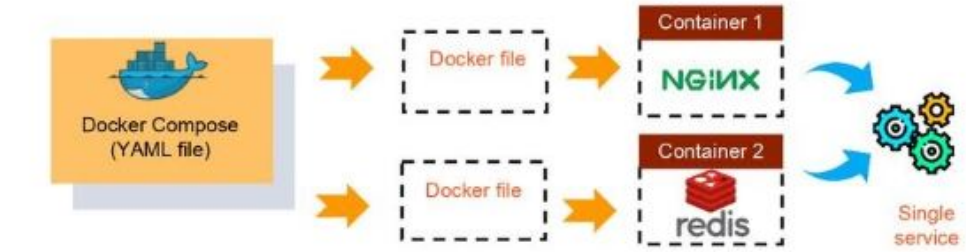


Figure 3.3: Docker Compose
[11]

Advantages of Docker Compose:

- Single Host for deployment:-For this purpose means you need only one piece of hardware to run everything
- Easy configuration:- Due to YAML scripts
- High productivity - It reduces the time it needed to perform tasks
- Security :- As the container are isolated and work of their own so it reduces threat

3.7 DOCKER HUB

Docker Hub is a cloud-based repository in which Docker users and partners create, test, store and distribute container images. Through Docker Hub, a user can access public, open source image repositories, as well as use a space to create their own private repositories, automated build functions, webhooks and work groups.[10]

Docker hub has these following features:-

- Image Repositories-helps us pulling and finding images in docker hub.We can push public as well as private repositories.
- Team and Organizations:-We can create a private repository within a organisation and team.

- GitHub and Bitbucket Integration:-It allows integration with source code repositories such as Github and BitBucket.
- Automated Builds :- If there is any change in the source code of github then the docker hub detects it and updates the image accordingly.
- Official and Publisher Images:- The high-quality images provided by dockers are considered official images, and it can be pulled and used. Similarly, high-quality images provided by external vendors are publisher images, also called certified images, which gives support and compatibility guarantee with Docker enterprise.[6]

3.8 DOCKER COMMANDS

These are some docker commands which are generally used:-

- docker –version:-To get the current version of docker
- docker pull <image name> :- It is used to pull images from docker repository.
- docker run -it -d <image name>:-It is use to run a container from image
- docker ps:-To get the list of all running containers.
- docker ps -a:-To get all the running and exited containers.
- docker exec -it <container id> bash:-To access the running container
- docker stop <container id>:-To Stop running container
- docker kill <container id>:To kill the container with giving it time to stop
- docker commit <conatainer id> <username/imagename>:-To create a new image of an edited container on the local system.
- docker images:-To list all the locally stored docker images
- docker rm <container id>:-To delete a stopped container
- docker rmi <image-id>:-To delete an image from local storage.
- docker build <path to docker file>:-To build an image from a specified docker file.
- docker-compose –version:-To check the version of docker compose
- docker-compose build:-This command builds images in the docker-compose.yml file.
- docker-compose up: This command does the work of the docker-compose build and docker-compose run commands. It builds the images if they are not located locally and starts the containers

4

OPEN TWIN DEPLOYMENT TASKS

The following table defined the tasks that were done with agile method per week:

Task No.	Task Date	Task
1	22nd Nov 2020	Setting up open twin environment
2	29nd Nov 2020	Researching about Docker Deployment and AWS
3	7th Dec 2020	Creating Docker file and research for DLL's in docker
4	14th Dec 2020	Getting Access to High performance computing(HPC) Server
5	21th Dec 2020	Testing of Open twin Development Docker file
6	4th Jan 2021	Taking git user credentials python script
7	11th Jan 2021	Testing of batchfile and create a log file
8	18th Jan 2021	Finding out solutions related to error
9	25th Jan 2021	Creating a bat file for Deployment and polish the contents
10	1st Feb 2021	Presentation on Docker for Open Twin
11	8th Feb 2021	Create a python file for Docker Hub

Table 4.1: Table of all tasks defined with the dates

1. **Setting up open twin environment**:-
   In the first week we set up environment for Open Twin project.There were couple of steps that are need to set up environment for project to run.The steps include setting up MongoDB and installing rust. We use Visual studio 2017 as Integrated development environment(IDE) for this project.

2. **Researching about Docker Deployment and AWS** :-
   I researched about how the docker.As in the theory chapter 3 i have discussed all my finding regarding about docker.

3. **Creating Docker file and research for DLL's in docker**:- It includes creating a docker file to install necessary dependencies. The following dockerfile is as follows:

```
# escape=‘

# To know windows servercore
ARG WIN_VER="ltsc2019"
FROM mcr.microsoft.com/windows/servercore:$WIN_VER

# Setup Envoirnment variables
```

```
ENV chocolateyUseWindowsCompression "true"
ENV RUST_TOOLCHAIN="1.30.1"

# To install Visual C++ build tools
ADD https://aka.ms/vs/16/release/vs_buildtools.exe C:\TEMP\vs_
    buildtools.exe
ADD https://win.rustup.rs/x86_64 C:\TEMP\rustup-init.exe
ADD https://chocolatey.org/install.ps1 C:\TEMP\choco-install.ps1

# Let's be explicit about the shell that we're going to use.
SHELL ["cmd", "/S", "/C"]

# To install Rust
RUN powershell C:\TEMP\choco-install.ps1
RUN powershell C:\TEMP\rustup-init.exe -y --default-toolchain $env:
    RUST_TOOLCHAIN

RUN rustup toolchain install nightly-2018-08-01
RUN rustup default nightly
RUN rustup update
```

4. **Getting Access to High performance computing(HPC) Server**:- I was using Amazon web Services(AWS) free services but it very low space so i need something with more computational.So due to performance issues i have a access to HPC server.High performance computing (HPC), also known as supercomputing, refers to computing systems with extremely high computational power that are able to solve hugely complex and demanding problems.[9]

5. **Testing of Open twin Development Docker file**:- The above dockerfile was tested on HPC server and changes were made according to errors as the Amazon web services have some issues while testing constraints.

6. **Taking git user credentials python script** :- I created a python script for download code from the github .For this we have to install git library in python using *pip install git* The python script is as follows:

```
from git import Repo
import getpass
import os

full_local_path = os.getcwd()
full_local_path_new=''

try:
    full_local_path_new=os.mkdir(full_local_path+ '\project')
    print(full_local_path_new)
except OSError:
    print ("Creation of the directory %s failed" % full_local_path_
        new)
else:
```

```
    print ("Successfully created the directory %s " % full_local_
        path_new)
username = input("enter your github username")
password = input("enter your github password")
remote = f"http://username:password@github.com/sdgamer007/
    PythonScripts.git"

Repo.clone_from(remote, full_local_path_new)
```

7. **Testing of batchfile and create a log file**:-
Then above python file and docker file needs to run as a batchfile in windows.So for that i created a batchfile or .bat file in windows to automate the process.

```
 @echo off
call :isAdmin

if %errorlevel% == 0 (
    goto :run
) else (
    echo Requesting administrative privileges...
    goto :UACPrompt
)
exit /b
:isAdmin
    fsutil dirty query %systemdrive% >nul
exit /b

:run
 Pushd "%~dp0"

python gitprd.py
pause
:UACPrompt
    echo Set UAC = CreateObject^("Shell.Application"^) > "%temp%\
        getadmin.vbs"
    echo UAC.ShellExecute "cmd.exe", "/c %~s0 %~1", "", "runas", 1
        >> "%temp%\getadmin.vbs"

    "%temp%\getadmin.vbs"
    del "%temp%\getadmin.vbs"
exit /B

choco install git -fy
```

8. **Finding out solutions related to error** :-
While working with windows docker environment i came across 2 error broadly

- docker-compose fails with invalid volume specification on mongodb init script:- Driver options are not supported for local volumes.As You can't create a volume at a specific location, you have to use the defaults.We can create a volume using the command:

```
docker volume create vol1
```

This will creates a volume where the mountpoint is something like `C:\ProgramData\docker\columes\vol1\_data`.If you have a RAID array on E: you can't use that for the mountpoint as you could in Linux, because options aren't supported:

```
> docker volume create vol2 --opt device=e:\data
Error response from daemon: create vol2: options are not
    supported on this platform
```

There is no workaround for this right now and for more indepth of data we can look into the github https://github.com/docker/compose/issues/5842

- Visual Studio 2017 cannot be install through docker:- There is no option to download visual studio 2017 through docker except through some 3rd party applications.

9. **Creating a bat file for Deployment and polish the contents** :-
I create a file for deployment a bat file which will copy all the data from the environment a deployment folder to the container and also install mongodb script to install mongodb with some queries to run .The code is as it follows.The files are as follows:

```
#docker-compose.yml

version: "1"                              # Giving it a versiom
services:                                 # Starting of a service
  mongoDB:                                # Name of service
     image: mongo                         # Name of image
     build: ./Framework                   # Dockerfile build from
                                           # the folder
     container_name: "mongo_database"     # Name of container
     ports:                               # Ports to be use
        - "27017:27019"


# Dockerfile present in Framework folder
# escape=‘
FROM microsoft/iis:latest

SHELL ["powershell"]

VOLUME C:/data/db C:/data/configdb
ADD mongo-init.js C:/data/mongo/
ADD mongo-init.js /docker-entrypoint-initdb.d/

COPY mongo-init.js /docker-entrypoint-initdb.d/

EXPOSE 27017
EXPOSE 27018
EXPOSE 27019
```

```
# mongoinit.js is present int same folder

mongo <<EOF

use System3;

db.createCollection("Sites");

db.Sites.insert({
    "NetworkPath" : "C:\\Program Files\\MongoDB\\Server\\4.4\\data
        ",
    "DocStoragePath" : "C:\\Program Files\\MongoDB\\Server\\4.4\\
        data\\docs",
    "SiteId" : NumberInt(1)
});

db.createCollection("Users");
db.Users.insert({
    "UserName" : "canbeanything",
    "Password" : "a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e
        998e86f7f7a27ae3",
    "Version" : NumberLong(1)
});


EOF

#batscript deployment.bat

@ECHO ON

docker-compose up --build -d
docker run -p 27019:27019 -d -v mongovol:c:\data\db --name mongo_
    tool mongo
docker create  --name deploytool microsoft/iis
docker cp Deployment deploytool:c:/deployment/
docker start deploytool

PAUSE
```

10. **Presentation on Docker for Open Twin** :-
    I did the presentation regarding the docker can be found on here https://drive.google.com/file/d/1urogPh70DAoMm7_ATyFTOYs0txvV3dtb/view?usp=sharing

11. **Create a python file for Docker Hub**:
    I use docker python sdk can be installed using *pip install docker*

```
import docker
import json
```

```
client =docker.from_env()
apiVersion = docker.APIClient
print(client.containers.get("deploytool").attrs)
def login(username, password) :

    try:
        loginData=client.login(userName,password)
        return True
    except docker.errors.APIError as e:
        return False


count = 0
while True:
    userName = input("Hello! Welcome to Dockerhub! \n\nUsername: ")
    password = input("Password: ")
    val=login(userName, password)
    if val == True:
        tag = input("Enter tag name:")
        repo = input("Enter repo name:")
        message = input("Enter message:")
        author = input("Enter author name:")
        apiVersion.commit(container="deploytool",tag=tag,repository
            =repo,message=message,author=author)
        break
    else:
        pass
```

5

CONCLUSION AND FUTURE WORK

The report has presented Docker as an emerging implementation of software containers and setting up development as well as deployment of the Open Twin software.

As we have discussed the chapter 3 Theory , the Docker's main architectural parts such as docker engine , docker images, docker containers and also docker compose.We have also seen all the commands that need to used for the deployment and development of Open Twin simulation methods.

In chapter 4 Open Twin Deployment Tasks , i have discussed about the creating docker file of the development of the Open Twin simulation Project.I have created a bat script to download the project of Open Twin software.I have faced some issues regarding the development of the Open Twin Project but I still managed to create a docker file for deployment.

We can still do many things in this project as we can still separate deployment dlls into the different folders.Yes still today window containers are still bounded and have some restrictions but as new updates are coming it will be more stable and more user friendly

Part II

APPENDIX

BIBLIOGRAPHY

[1] Bowen. *Docker Images, Containers, and Storage Drivers*. 2017. URL: https://bowenli86.github.io/2017/01/10/tools/docker/Docker-Images-Containers-and-StorageDrivers/ (visited on 03/06/2021).

[2] *Digital.ai. The 14th annual state of agile report*. 2019. URL: http://math.tntech.edu/rafal/cliff11/index.html (visited on 03/04/2021).

[3] *Docker Inc. Introduction to container security: Understanding the isolation properties of docker.* 2016. URL: https://www.docker.com/sites/default/files/WP_IntrotoContainerSecurity_08.19.2016.pdf (visited on 03/04/2021).

[4] *Docker The Industry-Leading Container Runtime*. URL: https://www.docker.com/products/container-runtime/ (visited on 03/05/2021).

[5] *Dockerfile Overview*. URL: https://docs.docker.com/engine/reference/builder/#:~:text=A\%20Dockerfile\%20is\%20a\%20text,can\%20use\%20in\%20a\%20Dockerfile\%20 (visited on 03/08/2021).

[6] *Everything that You Should Know About Docker Hub*. 2020. URL: https://geekflare.com/docker-hub-introduction/ (visited on 03/09/2021).

[7] Jenny Fong. *Are Containers Replacing Virtual Machines?* 2018. URL: https://www.docker.com/blog/containers-replacing-virtual-machines/ (visited on 03/07/2021).

[8] Jenny Fong. *Docker overview*. 2018. URL: https://docs.docker.com/engine/docker-overview/ (visited on 03/06/2021).

[9] *High Performance Computing*. URL: https://en.wikipedia.org/wiki/HPC/ (visited on 03/09/2021).

[10] Emilly Mell. *Docker Hub*. 2020. URL: https://searchitoperations.techtarget.com/definition/Docker-Hub (visited on 03/09/2021).

[11] *What is Docker Compose: Benefits and Basic Commands*. 2020. URL: https://www.simplilearn.com/tutorials/docker-tutorial/docker-compose (visited on 03/08/2021).