**INTERNSHIP: PROJECT REPORT**

---

| | |
|---|---|
| Internship Project Title | RIO:125: Set up docker container for application development using BlockChain Network |
| Project Title | Set up Docker container for application development using BlockChain network |
| Name of the Company | Tata Consultancy Services |
| Name of the Industry Mentor | Mr. Debashish Roy |
| Name of the Institute | Sister Nivedita University |

| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
|---|---|---|---|---|
| 02/11/2021 | 16/01/2022 | 156 | Linux/Ubuntu | GitHub, Docker, Ethereum, JDK, Eclipse, Solc, NodeJs, Truffle, Ganache, Solidity, VIM, Git, etc |

Project Synopsis:

Docker is a computer program that performs operating-system-level virtualization, also known as containerization. It runs software packages as “containers.” The word “container” is borrowed from the transport industry. The container we see on the back of a truck, on a train, or on the ship, are all the same containers. Because these containers were standardized, it made transportation a whole lot easier. Cranes could be built to lift a container from a ship to a train. Imagine what we’d have to do before that? We had to open every container and unpack and pack goods from ship to train, train to truck, and it was all so inefficient.

Despite all the above, in the software development industry, this mistake is made each and every time. Every single time we have an application we need to deliver, we go through the same old rigmarole of setting up a Web server, setting up a website, a database, a firewall.

The advent of the cloud made these tasks seem possible with efficiency and ease because we want to control the environment our application runs in. We don't want other people's applications on a shared infrastructure to interfere with ours. We want efficiency and reliability. We want to ship our application packaged up as a container, easily configurable by our customers, so they can set things up quickly and easily. Most of all, we want reliability and security!

Docker simplifies all of the above. It is originated on Linux, but now it is over the Windows platform also. Docker can package an application along with all its dependencies in a virtual container and run it on any Linux server. This means that when we ship our application, we gain the advantages of virtualization, but we don't pay the cost of virtualizing the operating system.

What needs to be done in this project is set up a docker container from application development using a Blockchain network. The topic when broken down simplifies to:

1. Creating a dynamic docker image with docker compose and its variants
2. Test the dynamic docker image through the same docker
3. Creating a Blockchain network with Ganache

4. Install necessary plugins to support solidity
5. Once all of these are done, we also need to test the running processes and create a blockchain network.

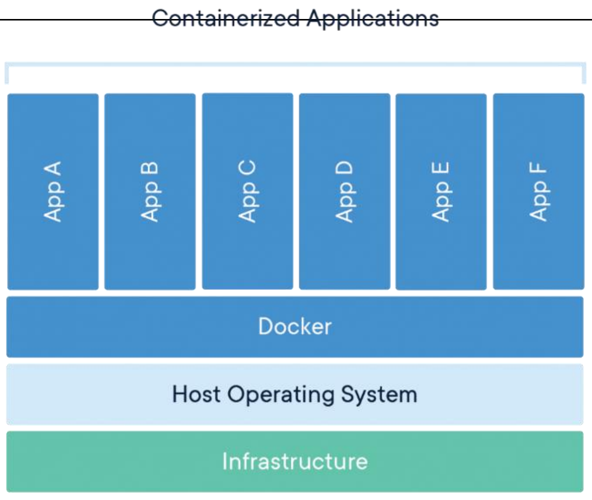
Solution Approach:

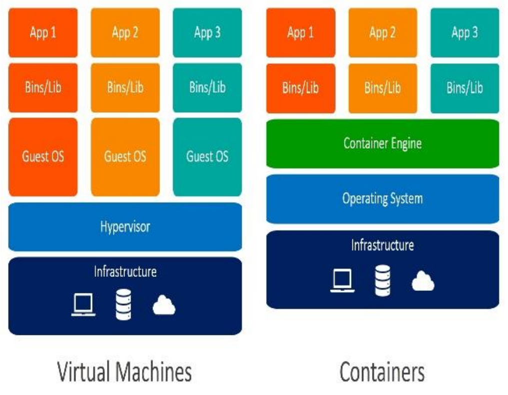
- Creating a docker file with the following specifications:
  a) Installing git (for version control)
  b) Installing vim (required for editing the files)
  c) Installing build-essential
  d) Installing openJDK (Java Development Kit)
  e) Configuring OpenJDK
  f) Setting the environment variables if not already configured in the GUI setup
  g) Installing Eclipse IDE and configuring the same in GUI setup
  h) Install the YAKINDU plugin for Eclipse to support solidity
  i) Set the YAKINDU plugin
  j) Installing EVM and configuring the same
  k) Installing Solc and configuring the same
- Creating a docker file and creating an image
- Once Image is created and up and running testing the image
- Once Image is up and running update the Docker file and add the following
  a) Install and configure NodeJS
  b) Install and configure truffle packages
  c) Install and configure testrpc
  d) Initialize truffle projects
  e) Deploying the contracts
  f) Creating DAPP
  g) Launch the DAPP server
  h) Install and configure Ganache
  i) Exposing the port
  j) Configure the environment variables if needed
- Create the docker image after updating the Docker files
- Create a blockchain network with Ganache using the image created after updating the Docker file
- Test the workspace and interface
- Perform transactions

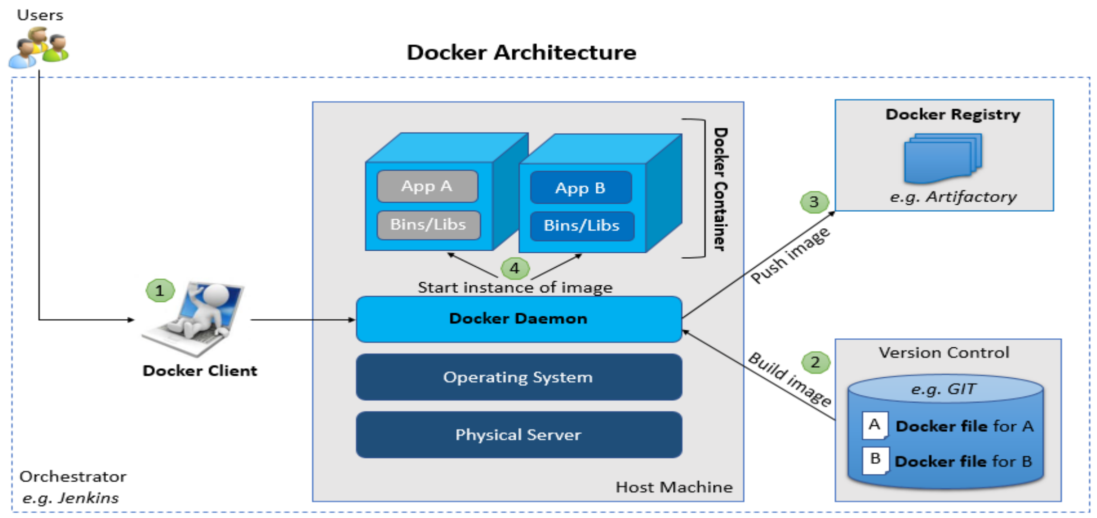
Assumptions:

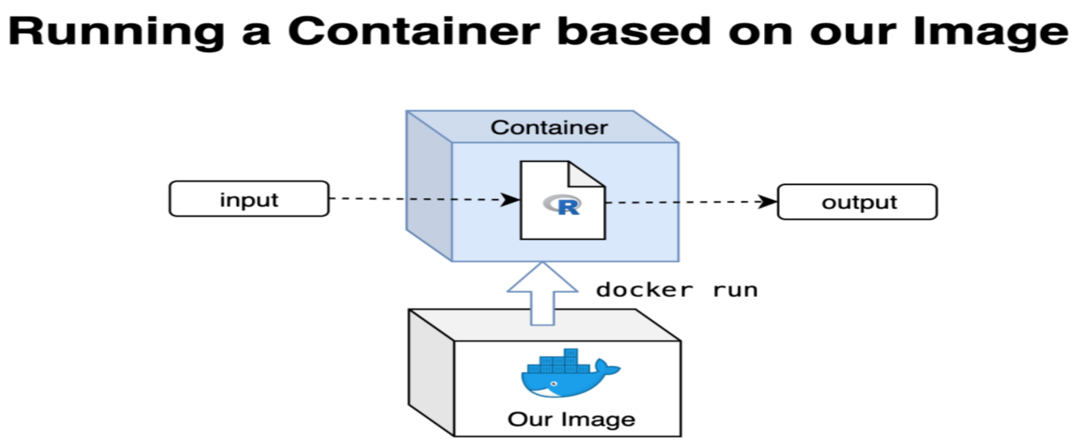
- The OS used is Linux (Ubuntu)
- Git is pre-installed and a root folder is created.
- Docker is updated
- The IDE used is Eclipse and the text editor used is VIM
- Docker's official GPG key is used

Project Diagrams:

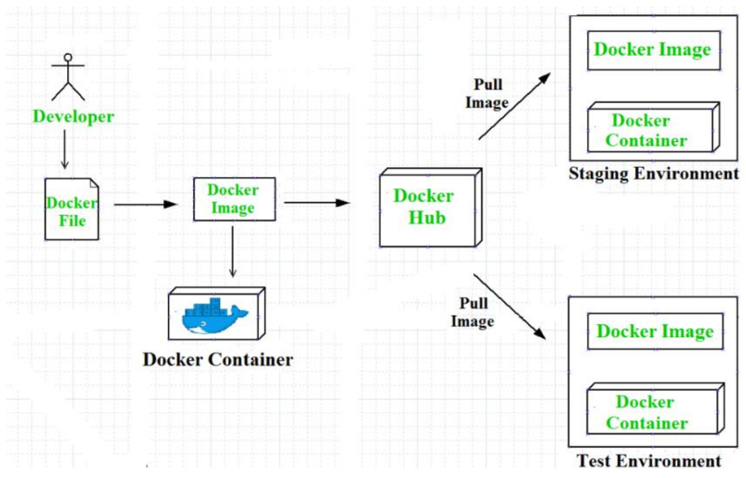

Containerized Applications
App A
App B
App C
App D
App E
App F
Docker
Host Operating System
Infrastructure


App 1
App 2
App 3
Bins/Lib
Guest OS
Hypervisor
Infrastructure
Virtual Machines
App 1
App 2
App 3
Bins/Lib
Container Engine
Operating System
Infrastructure
Containers


Users
Docker Architecture
Docker Client
1
App A
Bins/Libs
App B
Bins/Libs
Docker Container
4
Start instance of image
Docker Daemon
Operating System
Physical Server
Host Machine
Docker Registry
e.g. Artifactory
3
Push image
2
Build image
Version Control
e.g. GIT
A Docker file for A
B Docker file for B
Orchestrator
e.g. Jenkins


Running a Container based on our Image
Container
input
output
docker run
Our Image

Algorithms:



- Setting Base ubuntu as ubuntu 18.04
- Running an update and installing git, vim, curl, OpenJDK
- Install JRE
- Installing the Eclipse IDE
- Set environment variable:

  ENV JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/jre/bin/java

- Installing Ethereum
- Installing NodeJS and NPM
- Installing ganache, express and solc
- Updating the system again after the installations are done
- Making a directory and installing truffle@4.1.15
- Exposing the port: 8080 in my case
- While setting the truffleconfig set host to 127.0.0.1, port to 9575 and network id to *
- Set contracts and migration
- Creating a simple bank application using Java script and testing it.

Outcome:

- Dynamic Docker Image is up and running
- Docker compose file is ready
- Image was tested and ready
- BlockChain network created with Ganache CLI
- The application development container is ready
- The platform is tested using smart contracts

Exceptions considered:

- The image will not run when antivirus is up and running
- The Ubuntu version installed is 18.04
- The system was updated at the very first place

|  |
|---|
| ● Environment variable for Eclipse and the relevant packages are installed |
| Enhancement Scope:<br><br>● The docker file can be enhanced and the execution can be made fast<br>● The security of the whole thing can be made better<br>● Other than the two above points the all the other things are covered to the best of my knowledge |
| Link to Code and executable file: https://github.com/CoderShubham2000/Docker-Container-Setting |