**INTERNSHIP: PROJECT REPORT**

------------------------------------------------------------------------------------------------------------------------------------------

|  |  |
| --- | --- |
| Internship Project Title | Rio 45 Internship |
| Project Title | Set up virtual application development environment with docker containers on AWS Cloud |
| Name of the Company | TCS |
| Name of the Industry Mentor | Debashis roy |
| Name of the Institute | Welfare institute of science and technology & management |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 24/7/22 | 2/8/2022 | 9 | AWS Cloud | AWS Cloud Account EC2 Service, Docker, Ubuntu Instance via CLI. |
| **Project Synopsis:**  The scope of this report is about the Dockerized application that I have created which I was working on it.  I have created the Docker file to install Git ,VIM , Build -Essential , JDK and Apache Tomcat in Ubuntu Instance hosted on AWS Cloud. | |  |  |  |
| Solution Approach:   1. First I have created my AWS Free tier account which helps me with the service to deploy my Ubuntu Instance. 2. Then I have learned about EC2 service what it is used for and deployed Ubuntu 18.04 Instance of type t2.micro 3. Then I have tested manually through command line by installing the softwares like Git, VM, Build essential to know about the Commands and the installation process. 4. Then I have learnt about Docker, what it is, its uses , and knowledge like containers, images. etc 5. Then I have installed the Docker software using the official documentation present in docs.docker.io 6. Then I have learnt about the docker file and the components used to build a Docker file like FROM , RUN , EXPOSE , WORKDIR etc. 7. Then I wrote the Docker file to install the Git , VIM , build-essential and JDK. 8. Then I installed Apache Tomcat on Ubuntu Instance 9. Then I created a Docker image from the Dockerfile 10. Then I ran the container from the created image by exposing the container port 8080 to the host port 8080. 11. Then I tested the Tomcat application whether it is working or not. | |  |  |  |
| Assumptions: Not applicable | |  |  |  |
| Project Diagrams:  Not applicable | |  |  |  |
| Algorithms:  Installing Docker on Ubuntu from the Official Documentation  Commands  sudo apt-get update  $ sudo apt-get install \  ca-certificates \  curl \  gnupg \  lsb-release  $ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin | |  |  |  |
| Outcome:  Ubuntu Installation – Successful on AWS    Docker installation – Successful on Ubuntu    Creation of Docker file including  FROM ubuntu:18.04  RUN apt-get update  RUN apt-get install git -y  RUN apt-get install vim -y  RUN apt-get install build-essential -y  RUN apt-get install default-jdk -y  RUN apt-get install default-jre -y  RUN apt-get -y update && apt-get -y upgrade  RUN apt-get -y install wget  RUN mkdir /usr/local/tomcat  RUN wget http://dlcdn.apache.org/tomcat/tomcat-8/v8.5.81/bin/apache-tomcat-8.5.81.tar.gz -O /tmp/tomcat.tar.gz  RUN cd /tmp && tar xvfz tomcat.tar.gz  RUN cp -Rv /tmp/apache-tomcat-8.5.81/\* /usr/local/tomcat/  EXPOSE 8080  CMD /usr/local/tomcat/bin/catalina.sh run  Creating an image from the Docker file through docker build -t tcsfinal.    Creating a container from the above image.    Next testing the Tomcat application whether it is accessible through the Public IP or not after exposing the container port 8080 to the host port 8080. | |  |  |  |
| Exceptions considered:  We have considered the Ubuntu 18.04 version  For installing the Java the default versions can be considered or the openjdk -11 can be considered.  Eclipse has been omitted as it was said it is not mandatory by the Mentor  As it required Ubuntu GUI version for the installation. | |  |  |  |
| Enhancement Scope:  We can install the higher version of Java and the Docker specified versions we want instead of the lower and default version. | |  |  |  |
| Link to Code and executable file: <https://github.com/venkupudi/TCSRIOProject>  Tomcat Application accessible link.  <http://54.226.37.109:8080/> | |  |  |  |

1