

**DevOps Tools**

Table of Contents

[1](#_Toc531965119)

[Table of Contents 2](#_Toc531965120)

[1 Lab Details: 3](#_Toc531965121)

[2 DevOps Tools: Guided Exercises 3](#_Toc531965122)

[2.1.1 GIT –Problem Statement. 3](#_Toc531965123)

[Guided Exercise 1: 3](#_Toc531965124)

[Guided Exercise 2: 6](#_Toc531965125)

[Guided Exercise 3: 7](#_Toc531965126)

[Guided Exercise 4: 11](#_Toc531965127)

# Lab Details:

You will be provided with the below DevOps Lab for practicing the guided exercise in this document.

1. Enroll for DevOps Tools - Practice Lab to practice these guided exercises.

# DevOps Tools: Guided Exercises

### Docker –Problem Statement.

A Leading Sport Promoters Company in India is planning to develop a web based fantasy sports platform that allows users to play fantasy cricket, football, kabaddi and basketball games with their statistical skills and game knowledge. They have introduced this platform as a mobile application initially and now planning to create a web application on the same.

There are lakhs of fans and players for this game and the company is planning to develop and deploy the project in DevOps environment. The project is under development and the daily project builds is planned to deploy in a Docker container as a part of continuous integration and delivery.

**Scope:**

You have been assigned the task of managing the containerization of project builds using Docker in DevOps environment.

Given the project build, you need to perform the below mentioned tasks.

* Configure the project for dockerization
* Build the docker image
* Push your docker image to Docker Hub.

**Steps:**

1: Install Docker Container

2: Install Docker tool box

3: Verify Docker installation

4: Import the REST Project to dockerize

5: Add Docker configurations in Docker file

7: Create Docker image

8: Deploy and run Docker image

9: Verify the Docker images in a Docker Container

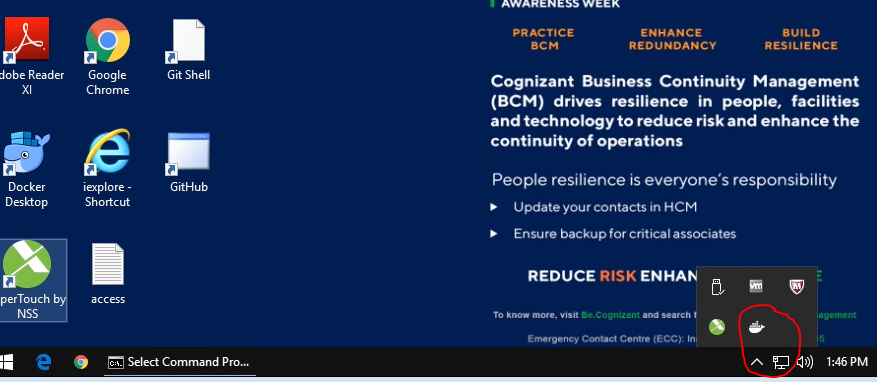
10: Create Docker cloud repository

11: Push image to Cloud repository

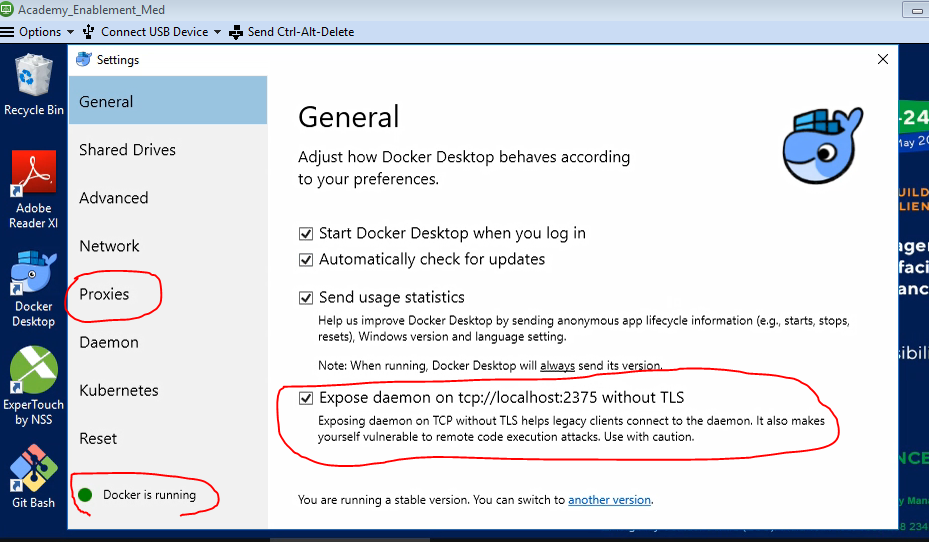
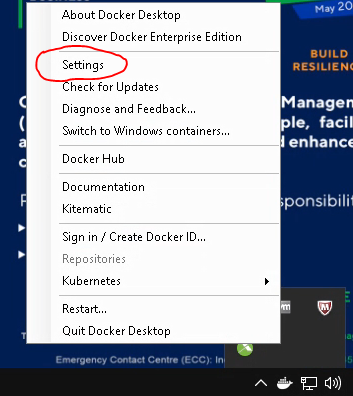
Guided Exercise 1:

**Step 1: Ensure Docker is running**

1. Next to the clock on the bottom right hand corner of the desktop, look for a Docker icon. In the Docker settings, you can see if Docker is running.

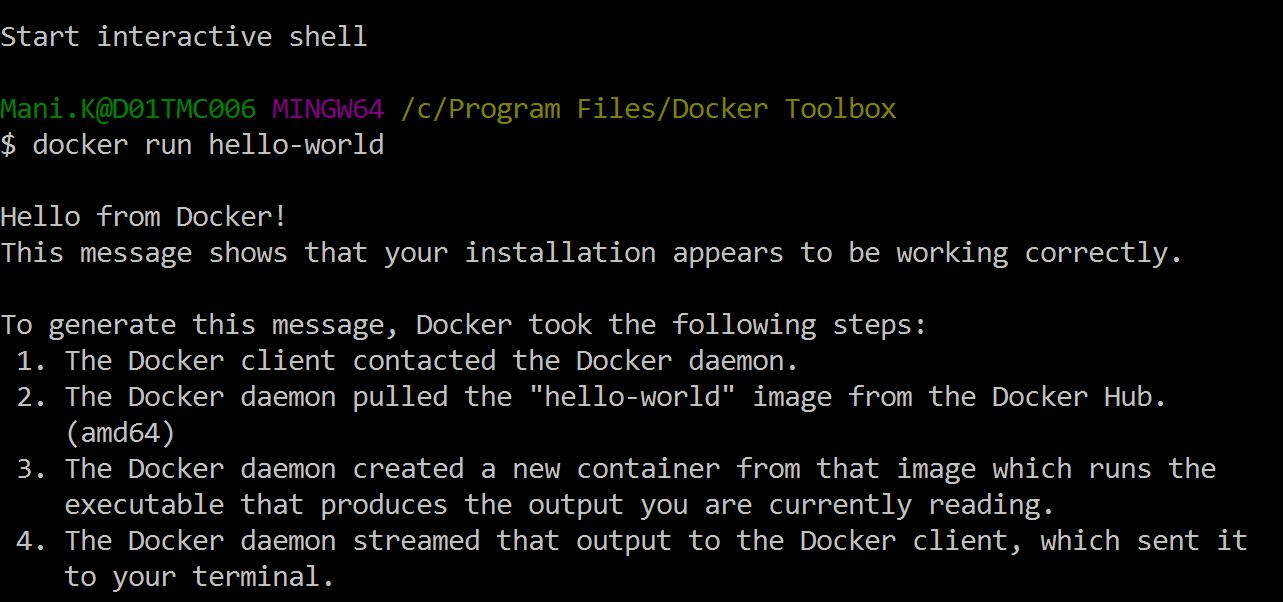


1. If Docker is not running, then double click on the Docker for Desktop icon on the Desktop. Wait for Docker to start running.
2. Once Docker is running, open the Docker settings and select the checkbox to Expose daemon on tcp://localhost:2375 without TLS.



1. To test your installation, open command prompt and execute the docker run command as shown below.

**Solution:**

**Command: docker run hello-world**

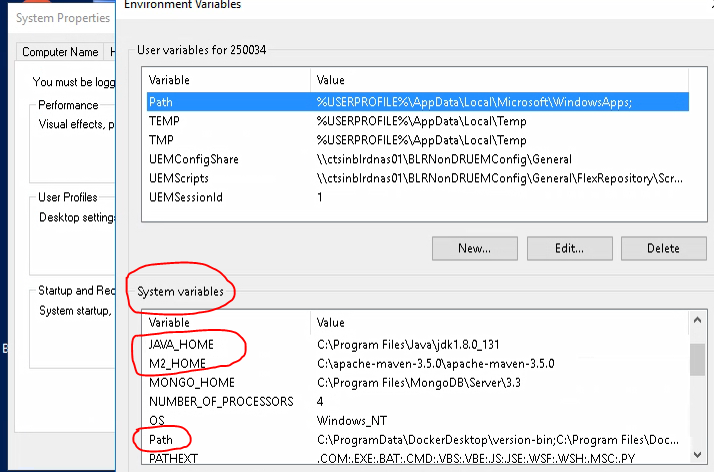
[Fig 1.4]

**Step 2: Set the Maven environment variable, M2\_HOME**

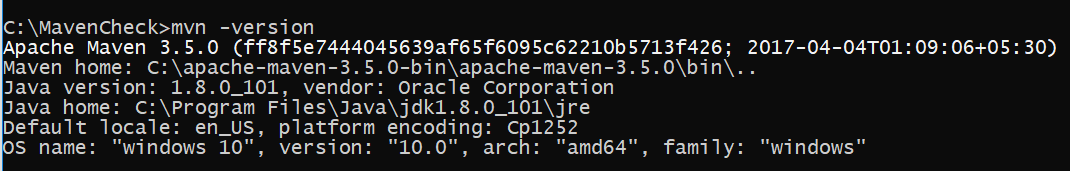
1. Install Maven in the desired directory. The Maven zip file can be found here :   
   C:\Soft\Apache Maven 3.5.0\3.5.0\apache-maven-3.5.0-bin.zip
2. Open Windows Advanced System Settings and edit the Environment variables. Set the environment variables, M2\_HOME to the path where you extracted Maven in the above step and append M2\_HOME\bin to the PATH environment variable.

M2\_HOME= C:\apache-maven-3.5.0\apache-maven-3.5.0

PATH should include %M2\_HOME%\bin (*This is to ensure that maven will be recognized as a command when executing command line tasks*)



**Step 3: Test the Maven Installation Ref [Fig 1.1]**

Maven installation is complete. Now let’s test it using the Windows Command Prompt. Type mvn -version in the command prompt. If you see the Maven and Java versions displayed correctly, you are good to proceed.

[Fig 1.1]

Guided Exercise 2:Configure the Application to deploy it in Docker

**Estimated Completion Time:** 20 Minutes

**Objective:** To configure the given project build to deploy it inside docker container.

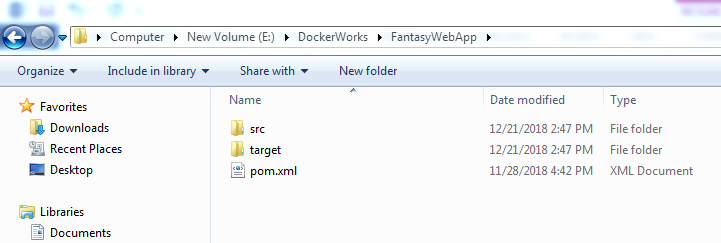
**Steps to follow:**

**Step 1:** Download the attached code template and build the application.

Code Template:



* Extract the above attached project into any location. The location used in this demonstration is E:\DockerWorks
* The folder structure will be shown as Figure 2.1.

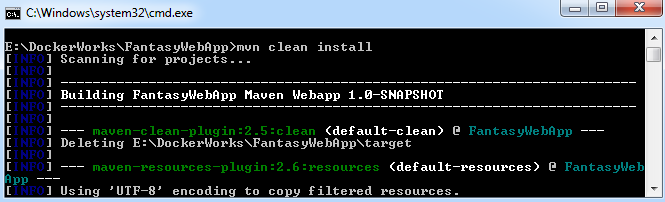


[Fig 2.1]

* Open a new command prompt instance and navigate to the location where the project is extracted.
* You can build the project and run the tests. . Refer Figure 2.2

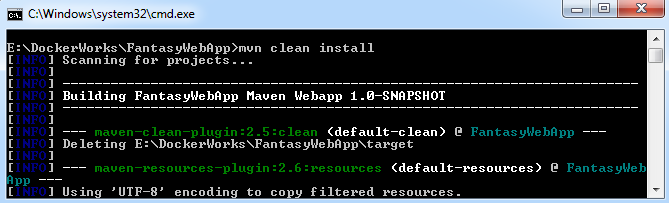
Solution:

Command: mvn clean install



[Fig 2.2]

* After the project is built successfully , use mvn clean package command to make the war file as shown in Figure 2.3

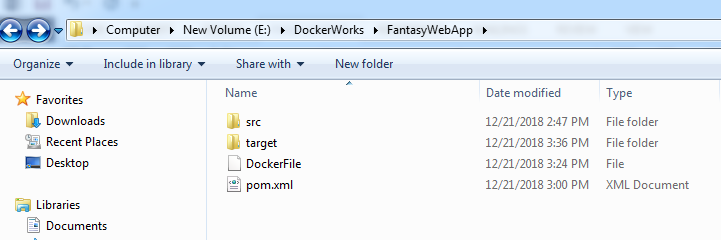


[Fig 2.3]

**Step 2:** Configure the application to add containerization.

* Create a file named “DockerFile” inside the project. Docker file has to be created without any extension and should be in the location where pom.xml exists.

Refer Figure 2.5



[Fig 2.5]

* The following code needs to be filled in docker file.
  + Specify the base image on which the application to be deployed. Refer Figure 2.6



[Fig 2.6]

#Download an image with tomcat instance

FROM tomcat:8.0.20-jre8

* Copy the application war to the base image downloaded. Refer Figure 2.7



[Fig 2.7]

#Copy the war file into tomcat webapps directory

ADD /target/fantasywebapp.war /usr/local/tomcat/webapps/

Guided Exercise 3:Dockerize the application and Verify.

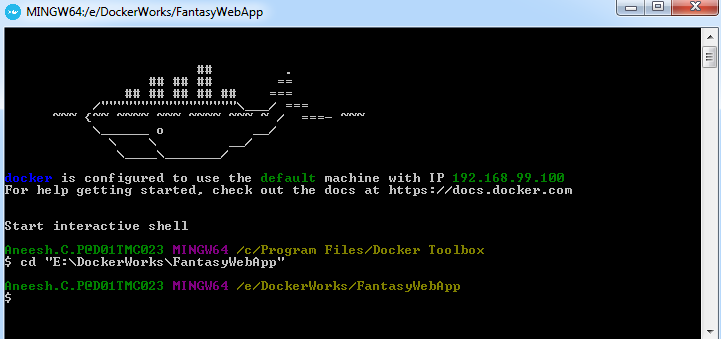
**Estimated Completion Time:** 20 Minutes

**Objective:** To containerize the application inside docker container.

**Steps to follow:**

**Step 1:**  Dockerize the application

* Open Command Prompt and navigate to the project location. As shown in Figure 3.1



[Fig 3.1]

* Use double quotes (““) for the path in Command Prompt.
* Use the docker build command to build the docker image.

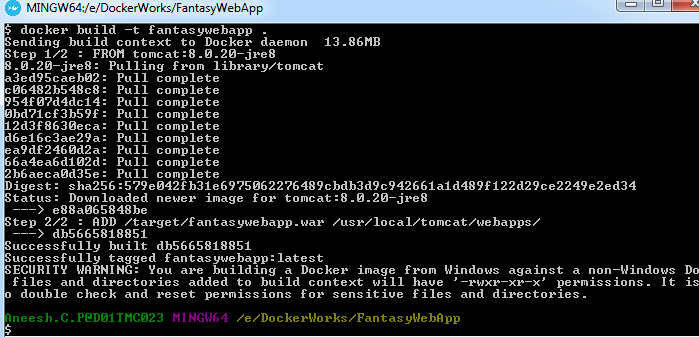
**Solution:**

**Command: docker build –t fantasywebapp .** (Figure 3.2)

* Where **fantasywebapp** is the docker image name. Make sure that the image name is in lowercase. Otherwise docker build command shows error.
* The dot (.) after the image name indicates that the docker file needs to be searched in the current directory.



* The above command searches for the local tomcat image in docker container. It opens the existing image if available or downloads a new image else. You will get a screen as shown in Figure 3.3 on successful built.



[Fig 3.3]

* View the docker images inside the docker container, and verify the existence of previously built docker image. Refer Figure 3.4

**Solution:**

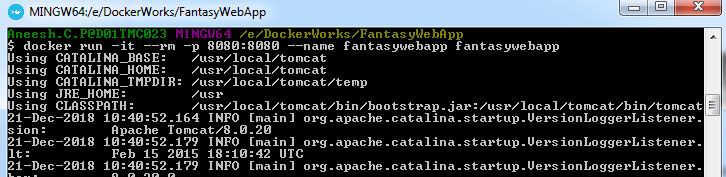
**Command: docker images**



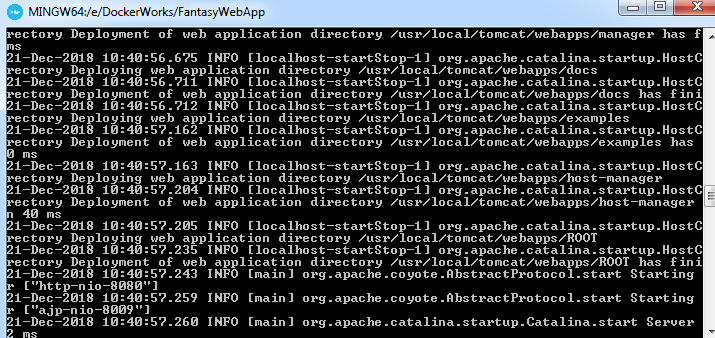
[Fig 3.4]

* To run the docker image, use docker run <<image name >> command.

Refer Figure 3.5 and 3.6

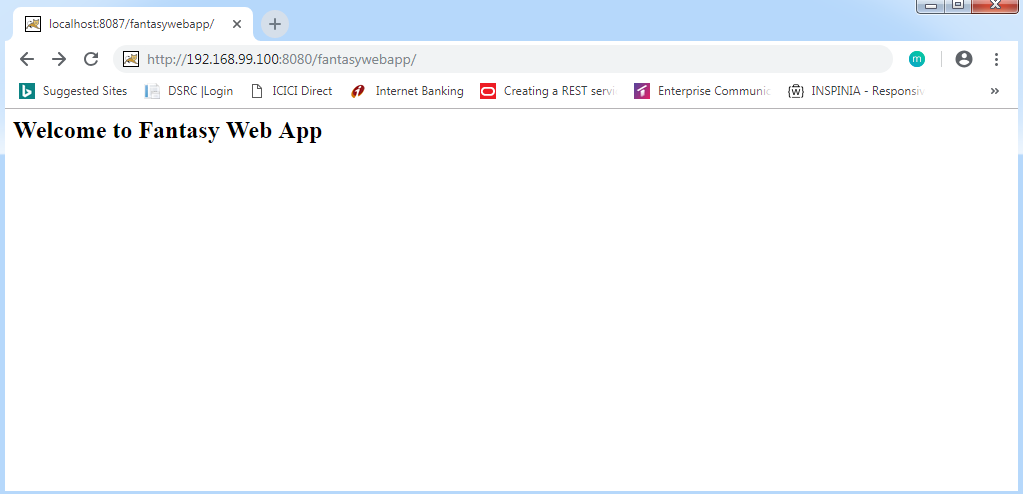


[Fig 3.5]



[Fig 3.6]

* Where 8080 is the exposed port for tomcat in docker container.
* After the application is started successfully in docker. To verify the application use <<docker ip: 8080>> in web browser.
* The default docker ip is 192.168.99.100 .Refer Figure 3.7



[Fig 3.7]

Guided Exercise 4:Push the docker image to cloud repository (Docker Hub)

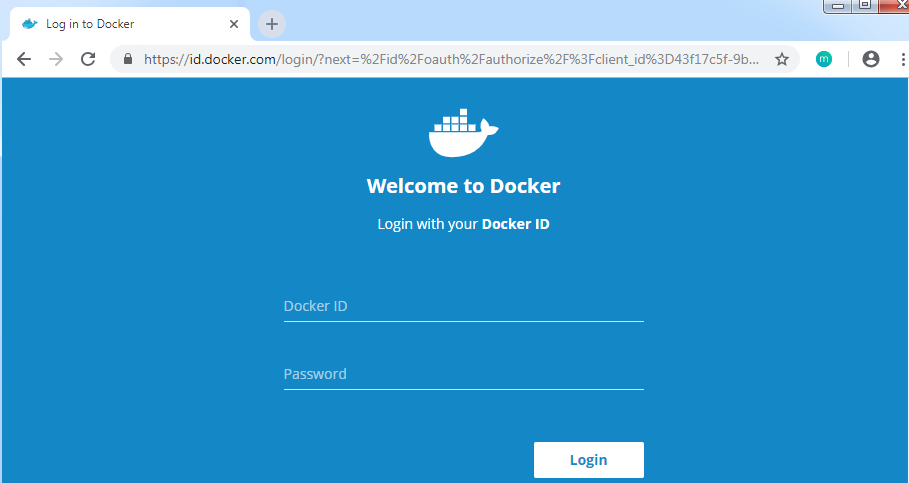
**Estimated Completion Time:** 20 Minutes

**Objective:** Push the created docker image to docker hub.

**Steps to follow:**

**Step 1:**  Login to Docker Hub.

* Open browser and enter the URL <https://hub.docker.com> to login to the docker hub.
* Click on **Sign In** link to display the login page as shown in Figure 4.1

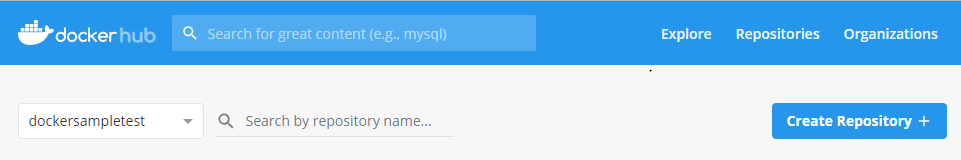


[Fig 4.1]

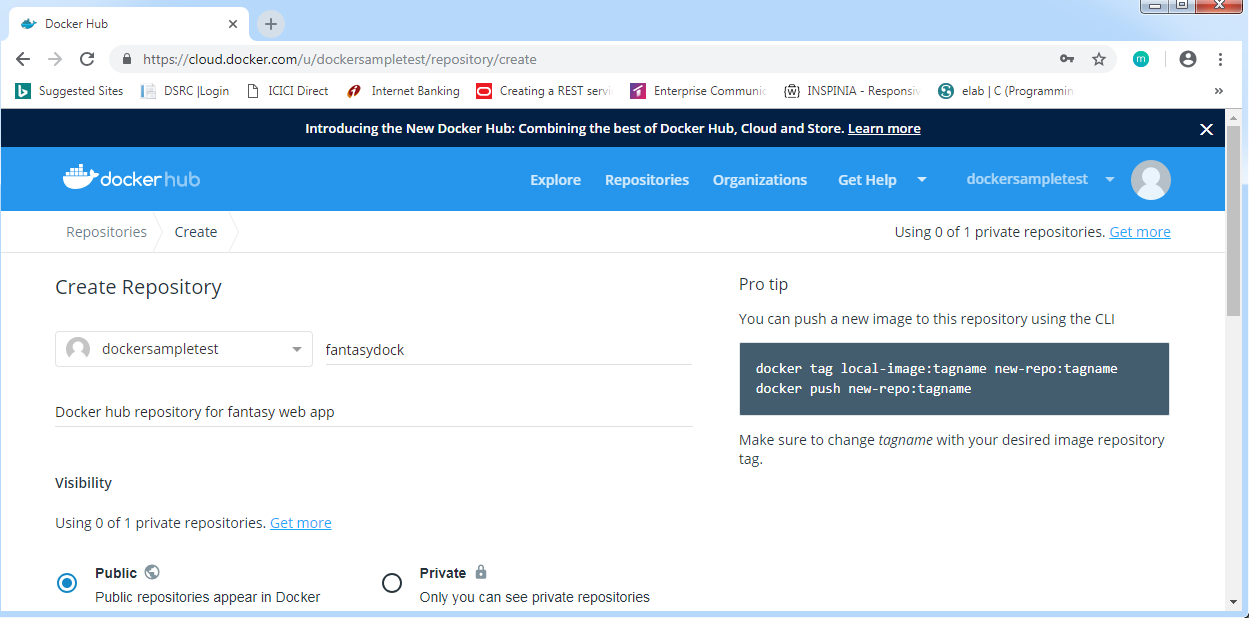
* Login with your docker credentials or sign up if you do not have credentials.

**Step 2:**  Create a repository in Docker Hub.

* To create Docker cloud repository, click on the create repository and enter the name of the repository and choose this repository public or private. You will see the screen as provided in Figure 4.2

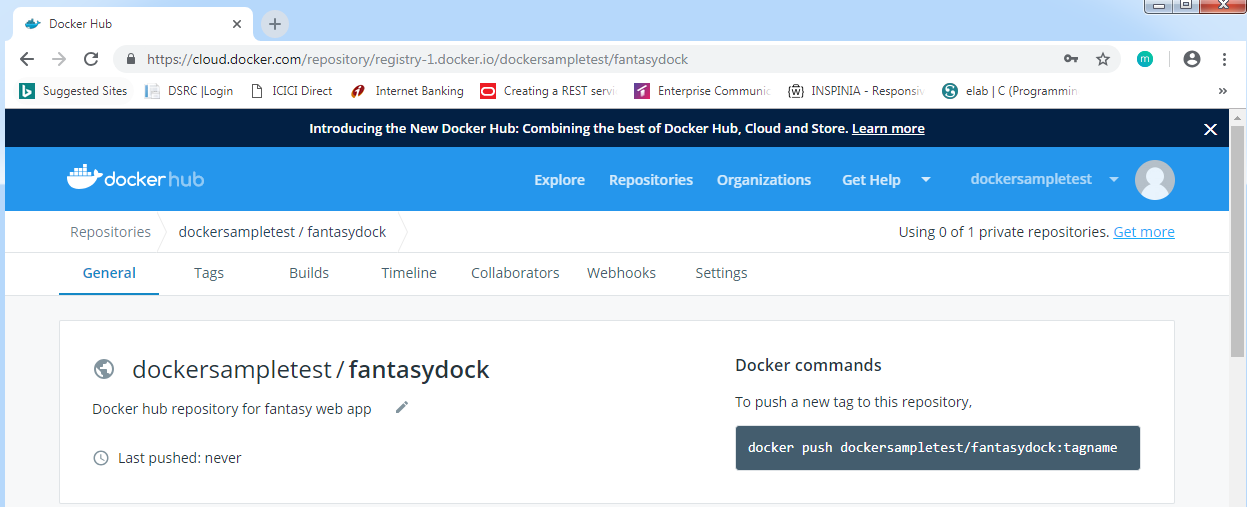
 [Fig 4.2]

* Create a repository named “**fatansyapp**” as shown in Figure 4.3



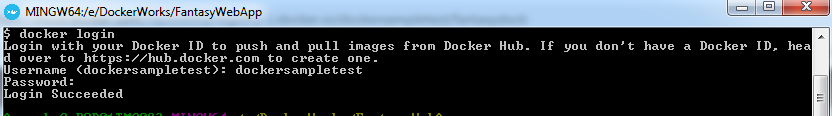
[Fig 4.3]

* The repository named **fatansyapp** will be created as Figure 4.4



[Fig 4.4]

* Open the command prompt and navigate to the project folder.
* Login to Docker hub from command prompt using docker login command as shown in figure 4.5



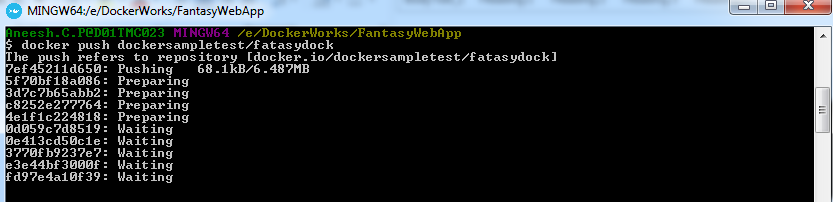
[Fig 4.5]

* Use docker tag command to tag the docker image that you want to push. Refer Figure 4.6



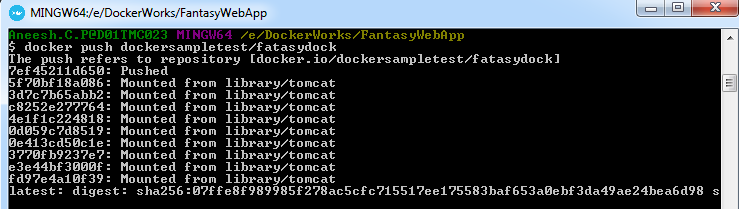
[Fig 4.6]

* Use docker push command to push the image to the docker repository as shown in figure 4.7



[Fig 4.7]

* The following screen shows the final console once the push is complete. Refer Figure 4.8



[Fig 4.8]

**Summary:**

You have learned to containerize application using Docker.