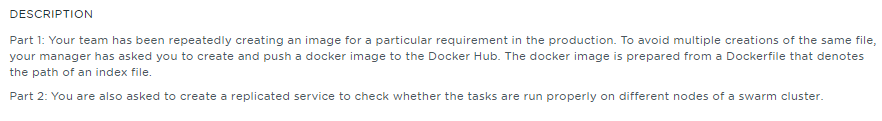
**Creating a Docker Image and Creating a Replicated Service on Swarm Cluster - Assessment**



**Table of Contents**

[**1.** **Introduction** 1](#_Toc44017916)

[**1.1.** **Create and Push a Docker image to Docker Hub** 1](#_Toc44017917)

[**1.1.** **Create replicated service on Swarm cluster** 1](#_Toc44017918)

[**2.** **Work Environment Setup** 1](#_Toc44017919)

[**2.4.** **Setup a Swarm Cluster with a Master and Worker Nodes** 2](#_Toc44017920)

[**3.** **Create and Push a Docker Image to Docker Hub** 3](#_Toc44017921)

[**3.1** **Create a Dockerfile** 3](#_Toc44017922)

[**3.2** **Build Docker Image** 4](#_Toc44017923)

[**3.3** **Push Docker Image to the Docker Hub** 4](#_Toc44017924)

[**4.** **Deploy Service in a Swarm Cluster** 5](#_Toc44017925)

[**4.1** **Create replicated services** 5](#_Toc44017926)

[**4.2** **Check tasks are running in available nodes** 6](#_Toc44017927)

[**5.** **References** 6](#_Toc44017928)

# **Introduction**

The intent of the project is to create a docker image and create the replicated services to ensure the tasks are running in the available nodes of the swarm cluster. The project is divided into two parts:

## **Create and Push a Docker image to Docker Hub**

The intent of this section is to demonstrate on how to create the Docker image based on the Dockerfile and the base image can be stored in the Docker Hub to avoid creating the images for every new changes.

## **Create replicated service on Swarm cluster**

The intent of this section is to show how to create the replicated services on Swarm cluster and ensure the tasks are running correctly in the available nodes of a Swarm Cluster

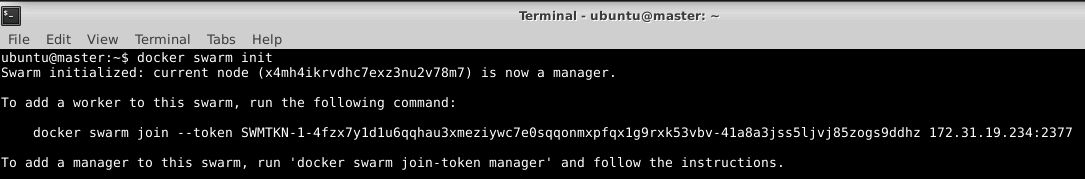
# **Work Environment Setup**

Refer the **Create\_Docker\_image\_Create\_replicated\_service\_swarm\_cluster\_writeup.docx**

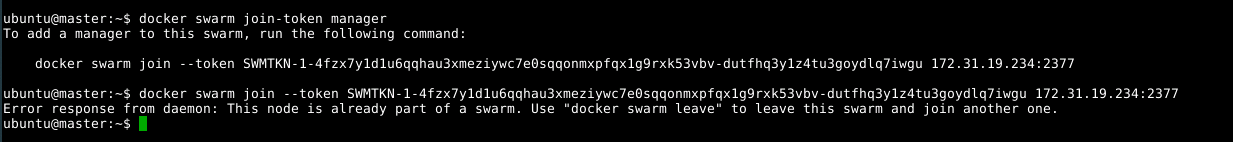
This project is performed using the Practice Lab session meant for the Docker Certified Associate.

## **Setup a Swarm Cluster with a Master and Worker Nodes**

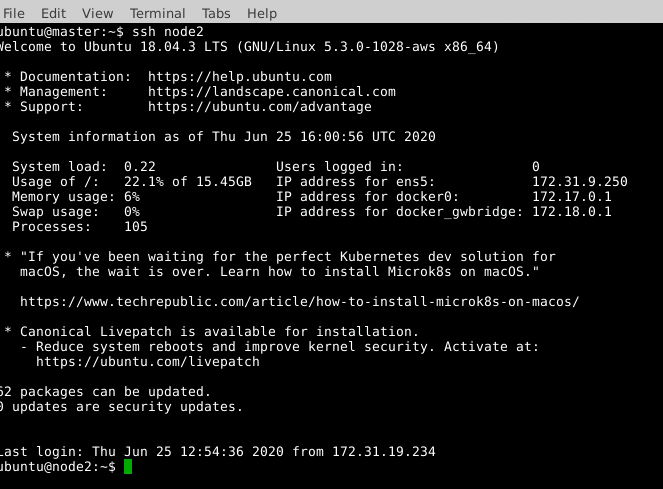
Initialize a swarm. The docker engine targeted by this command becomes a manager in the newly created single-node swarm.



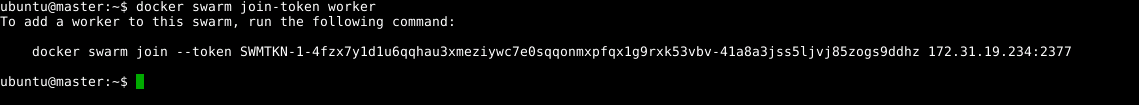
On the master node, run the below command mentioned in the previous screenshot to join it as a swarm manager

Note: In case the master node is already a part of the swarm, you will receive the following message:

Open a new Terminal window and navigate to node2 using the following command:



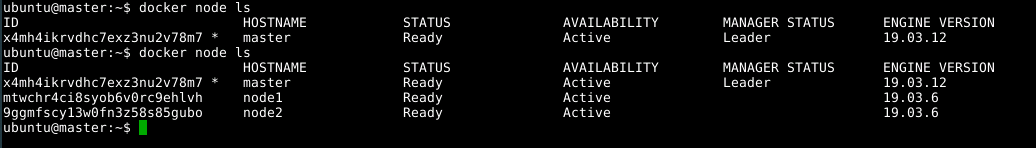
Run the following command on the master node to retrieve the join command for the worker nodes:



On node2, join the swarm as a worker node using the token generated in the previous screenshot

Repeat the same steps to add **node1** as a worker node

In the master node, Run the following command to list all the available nodes



Now both the master and worker nodes are joined in the Swarm cluster

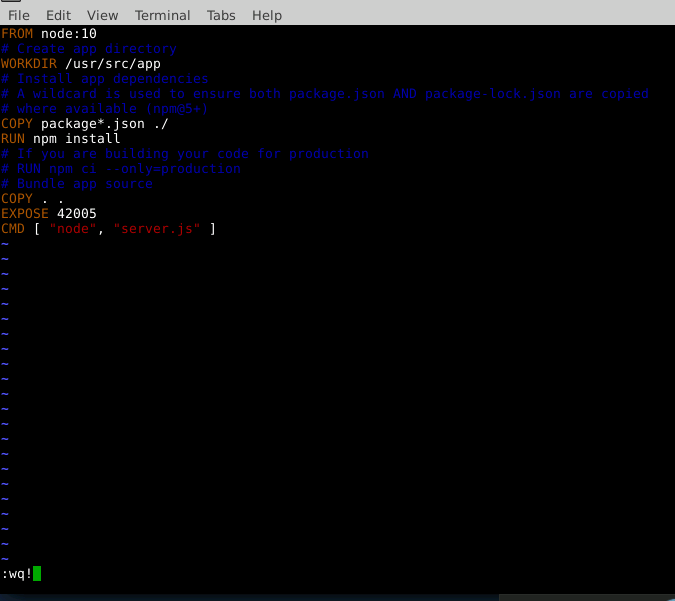
# **Create and Push a Docker Image to Docker Hub**

First step is to create an empty project directory. The directory is the context for our application image. The directory only contains the files to build that image

## **Create a Dockerfile**

Create a new file called ‘Dockerfile’ in our project directory. This file defines an application image content via one or more build commands that configures the image

Add the following contents to the Dockerfile



In this example, the **Dockerfile** starts with the parent image i.e node:10 image

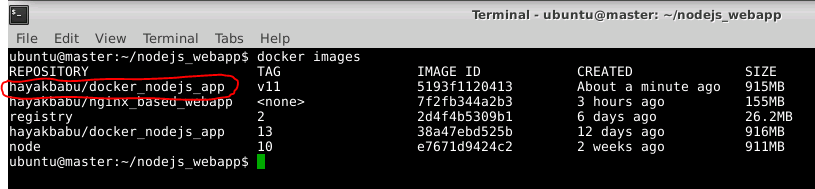
## **Build Docker Image**

The below command is used to build a Docker image



The parameters that we have passed are:

* -t is the Docker image tag. You can give a name to your image and a tag. (image is **hayakbabu/docker\_nodejs\_app** is the docker hub user and the repo name and the tag is **v1**)
* The second parameter (‘.’) specifies the location of the Dockerfile that we created. Since we created the Dockerfile in the same folder in which we are running the docker build, we specified the current directory.

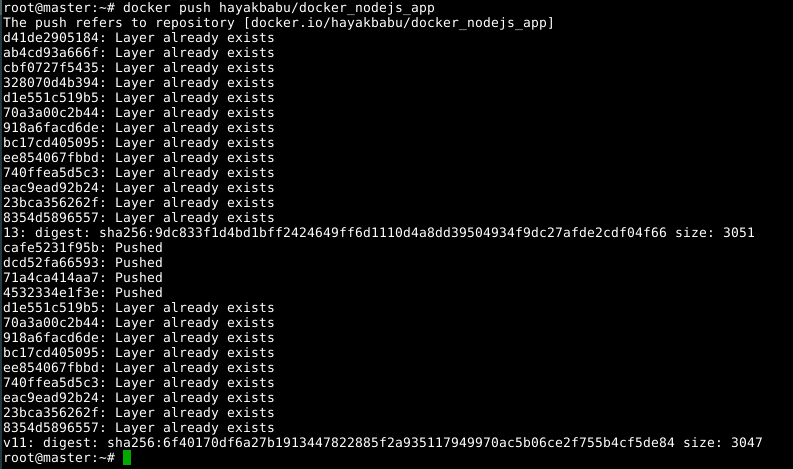


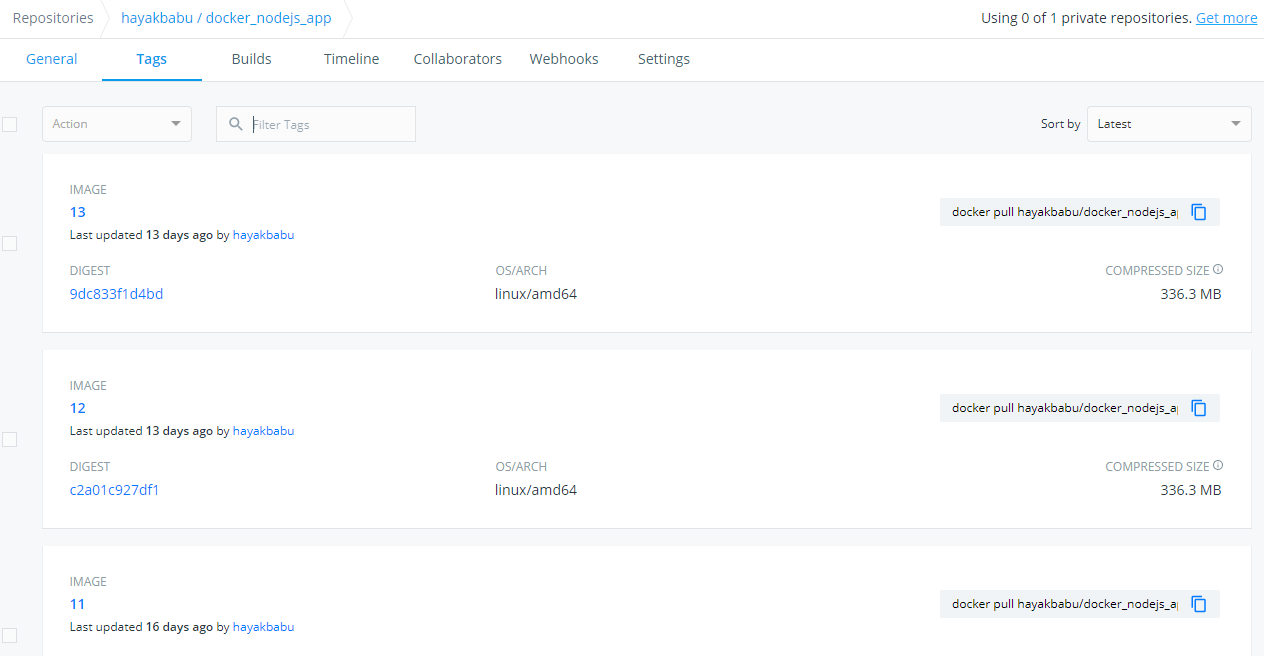
## **Push Docker Image to the Docker Hub**

[Docker Hub](https://hub.docker.com/) is the place where open Docker images are stored. Our objective is to push the image to the Docker Hub. By executing the below command, the created images is pushed to the Docker image

It may ask for the login credentials based on the configuration upon executing this command. After entering the login credentials, the image is being pushed to the Docker Hub repository

The below screenshots shows the image is successfully pushed to the docker hub



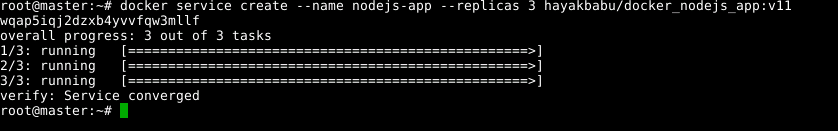


From now on, the developer can pull the image from the docker image for further enhancements and push it back to the docker hub.

# **Deploy Service in a Swarm Cluster**

## **Create replicated services**

The below command creates the replicated services of

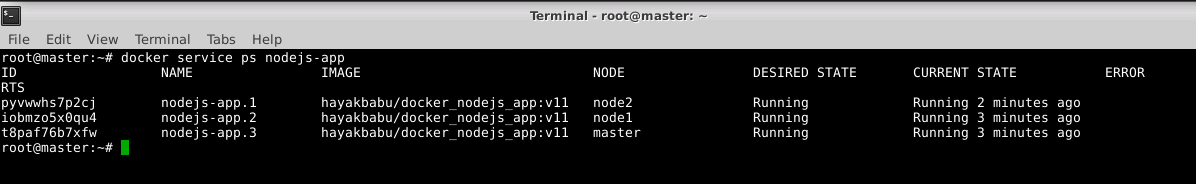


## **Check tasks are running in available nodes**

The below command lists all the docker services of



The below command displays the status of the service hayakbabu/docker\_nodejs\_app:11 which were running in different nodes like node1 and node2



# **References**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Components** | **Reference** |
| 1 | Docker Engine Installation | <https://docs.docker.com/engine/install/ubuntu/> |
| 3 | Writeup Document | Docker\_image\_Replicated\_service\_swarm\_writeup.docx |
| 4 | Sources | <https://github.com/babkannan/docker_image_create_replicated_service_swarm_cluster> |