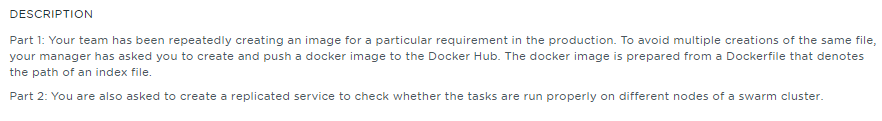
**Creating a Docker Image and Creating a Replicated Service on Swarm Cluster - Assessment**



**Table of Contents**

[**1.** **Introduction** 1](#_Toc44015036)

[**1.1.** **Create and Push a Docker image to Docker Hub** 1](#_Toc44015037)

[**1.1.** **Create replicated service on Swarm cluster** 1](#_Toc44015038)

[**2.** **Work Environment Setup** 2](#_Toc44015039)

[**2.1.** **Three networked host machines** 2](#_Toc44015040)

[**2.2.** **Install Docker Engine** 2](#_Toc44015041)

[**2.3.** **Open Protocols and ports between the Hosts** 2](#_Toc44015042)

[**2.4.** **Setup a Swarm Cluster with a Master and Worker Nodes** 2](#_Toc44015043)

[**3.** **Create and Push a Docker Image to Docker Hub** 3](#_Toc44015044)

[**3.1** **Create a Dockerfile** 3](#_Toc44015045)

[**3.2** **Build Docker Image** 3](#_Toc44015046)

[**3.3** **Push Docker Image to the Docker Hub** 3](#_Toc44015047)

[**4.** **Deploy Service in a Swarm Cluster** 4](#_Toc44015048)

[**4.1** **Create replicated services** 4](#_Toc44015049)

[**4.2** **Check tasks are running in available nodes** 4](#_Toc44015050)

[**5.** **References** 4](#_Toc44015051)

# **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**

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

Here we can list out the Software Pre-requisites required to run this project though all the software’s are pre-built in the Practice lab

## **Three networked host machines**

The practice lab has pre-configured Linux hosts which have Docker installed and can communicate over a network via SSH. However, these can be either physical machines, virtual machines, Amazon EC2 instances. One of the machine is a master node and two of them are worker nodes.

## **Install Docker Engine**

Docker Engine is mandatory to get it installed on all the nodes participating in the Swarm cluster

Refer <https://docs.docker.com/engine/install/#server> for installing the Docker Engine in all the nodes

## **Open Protocols and ports between the Hosts**

The following ports must be available. On some systems, these ports are open by default.

* TCP port 2377 for cluster management communications
* TCP and UDP port 7946 for communication among nodes
* UDP port 4789 for overlay network traffic

## **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.

|  |
| --- |
| $ docker swarm init |

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

|  |
| --- |
| $ docker swarm join –token SWMTKN-1-4fzx7y1d1u6qqhau3xmeziywc7e0sqqonmxpfqx1g9rxk53vbv-dutfhq3y1z4tu3goydlq7iwgu 172.31.19.234:2377 |

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:

|  |
| --- |
| $ ssh node2 |

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

|  |
| --- |
| $ docker swarm join-token worker |

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

|  |
| --- |
| $ docker swarm join --token SWMTKN-1-4fzx7y1d1u6qqhau3xmeziywc7e0sqqonmxpfqx1g9rxk53vbv-41a8a3jss5ljvj85zogs9ddhz 172.31.19.234:2377 |

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

|  |
| --- |
| $ docker nodes ls |

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

|  |
| --- |
| FROM node:10  # Create app directory  WORKDIR /usr/src/app  # Install app dependencies  # A wildcard is used to ensure both package.json AND package-lock.json are copied  # where available (npm@5+)  COPY package\*.json ./  RUN npm install  # If you are building your code for production  # RUN npm ci --only=production  # Bundle app source  COPY . .  EXPOSE 42005  CMD [ "node", "server.js" ] |

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

|  |
| --- |
| $ docker build –t hayakbabu/docker\_nodejs\_app:v11 . |

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/nginx\_based\_webapp** 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

|  |
| --- |
| $ docker push hayakbabu/docker\_nodejs\_app:v11 |

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

Refer the screenshots document which show 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

|  |
| --- |
| $ docker service create –name nodejs-app –replicas 3 hayakbabu/docker\_nodejs\_app:11 |

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

The below command lists all the docker services of

|  |
| --- |
| $ docker service ls |

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

|  |
| --- |
| $ docker service ps hayakbabu/docker\_nodejs\_app:11 |

# **References**

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