Docker Containers Configuration and Deployment

Large Scale Data Management

2018/2019

The main goal of this guide is to understand how to configure and deploy docker containers running different services/applications.

Useful Docker documentation is available at:

- Docker Documentation - https://docs.docker.com
- Step by Step Example - https://docs.docker.com/get-started/

Setup

1. Install Docker from binaries as described in https://docs.docker.com/install/linux/docker-ce/binaries/
2. Install Docker machine as described in https://docs.docker.com/machine/install-machine/
3. Download the sample.war file at https://tomcat.apache.org/tomcat-6.0-doc/appdev/sample/ to a folder called myapps, while this folder should be placed inside the docker env folder. This war file will be used to test the tomcat service that will be deployed during this lab guide.

Steps

Deploying a single Docker Container

1. Setup a Docker configuration file (Dockerfile) that will deploy apache tomcat in a Ubuntu 16.04 image. The commands for installing and starting tomcat in ubuntu are:

```
apt - get update && apt - get -y upgrade
apt - get -y install software - properties - common curl
apt - get -y install default - jdk
```

```
curl -O http :// archive. apache . org / dist / tomcat /
tomcat -7/ v7 .0.55/ bin / apache - tomcat -7.0.55. tar .gz
tar xzf apache - tomcat -7.0.55. tar . gz
apache - tomcat -7.0.55/ bin / startup . sh && tail -f
apache - tomcat -7.0.55/ logs / catalina . out
```

2. The Dockerfile should specify a command to copy the myapps folder to the path /apache-tomcat-7.0.55/webapps/ at the Docker container.
3. Port 8080 should be exposed to the host.
4. For the previous configurations see the documentation of FROM, RUN, EXPOSE, CMD and COPY Dockerfile commands.
5. Build the Docker image using the docker build command. The image should be named my/tomcat.
6. Deploy the container with the command docker run. In order to be able to reach tomcat from the host machine use the -p 8080:8080 flag. To run the container in background use the -d flag.
7. Understand the usage of the commands docker ps, exec, stop, kill. E.g., understand and explore the docker exec -ti ”container id” /bin/bash command.
8. Access the tomcat service from the host machine browser (localhost:8080/sample).

Docker-Swarm and Compose Utilities

1. Setup a Swarm cluster as described in https://docs.docker.com/get-started/part4/#create-a-cluster
2. Understand the usage of the command docker node to check if the nodes have successfully joined the Swarm cluster.
3. At VM1, create a docker compose YAML file that specifies a service called web that will use two replicas and will export port 8080 to the host (https://docs.docker.com/get-started/part3/#docker-composeyml)
4. At VM1, the master node, launch the docker stack with the command docker stack deploy. Note that the compose configuration file only needs to be present at the master node. However, since we are not using a docker registry for the tomcat docker image, this image must be present (built) on both VMs
5. Check that the docker containers are running on both VMs (swarm nodes) with the docker ps command.
6. Access the tomcat service from the hosts (vm1:8080/sample) and (vm2:8080/sample)

Learning outcomes

Experiment linux Docker containers configuration and deployment. Experiment Docker cluster deployment and configuration. Revise Docker configuration parameters and deployment/management commands.