Build Your First Docker Image

Any valid Dockerfile must have FROM as the first noncomment instruction. The argument to FROM defines the base image upon which subsequent instructions in the Dockerfile are executed, such as add packages or install JDK. This base image could be for an operating system such as ubuntu for the Ubuntu operating system, or centos for the CentOS operating system. There are base images available for different operating systems at the Docker website. Additional packages and software can then be installed on these images.

Our First Dockerfile

Let’s create our first Dockerfile:

1. Create a new directory.

This directory will contain the Dockerfile and any other artifacts that need to be included in the image.

2. In this directory, create a new text file and name it Dockerfile. In this file, enter the following code:

```
FROM openjdk
CMD ["java", "-version"]
```

Here’s a breakdown of the image definition:

• This Dockerfile uses openjdk as the base image. This is a prebuilt image on Docker Hub and can generally be used as the base image for all images that need the Java runtime.

• The CMD instruction defines the command that needs to run. The command in this case is simply printing the version of the Java interpreter.

Any other dependencies or libraries, such as JAR files, can be included in this image using the COPY instruction. Then a Java command using that JAR file can be invoked by setting the appropriate classpath.

Build Your First Docker Image

```
docker build -t hello-java .
```

- The docker build command builds the image. -t provides a name for the image. hello-java is the name of the image. . is the context for the command. This context is used as the base directory for copying any files to the image. No files are copied in this case.

- The openjdk image is downloaded from Docker Hub. It also downloads the complete chain of base images.
- The CMD instruction adds a new layer to the image.

List the Docker Image

List the images available using the *docker images* command.

Run Your First Docker Container

You can run a Docker container by using the docker run command and specifying the image name. Let’s run our image as shown here:

```
docker run hello-java
```

Push Image to Docker Hub

Docker Hub is a software-as-a-service (SaaS) registry service. You can search, manage, push, and pull images to this registry. The images can be manually pushed to the registry using the docker push command. Alternatively, they can be built when changes are pushed to a GitHub or Bitbucket repository. User and team collaboration can be facilitated by creating public and private registries.

The search and pull commands can be invoked without having an account on Docker Hub. However, the push command requires you to have an account on Docker Hub, which can be created using the docker login command or at the Docker Hub website.

The image hello-java does not have a Docker Hub username associated with it, which is needed to push the image to Docker Hub. The IMAGE ID column in the following code block shows a unique identifier assigned to this image. A Docker Hub username can be associated with this image using docker tag and the image ID as shown here:

```
docker tag d20e74664517  roynilanjan1/hello-java:latest
```

The Docker Hub username here is *roynilanjan1* . Make sure to change this to your Docker Hub username. By default, the latest tag is assigned to the image. In this case, an explicit latest tag is assigned.