Build Your Own Docker Image

Sang Shin
JPassion.com
"Code with Passion!"

Topics

- Build and run a simple Java SE application

Build and run Simple Java application as Docker app

Steps to take

- Write Java code
- Write a Dockerfile
  - Dockerfile is a text file that contains all the commands that are used to assemble an image
- Build Docker image from a Dockerfile
  - docker build . -t <image>
- Run the image
  - docker run <image>
- Push the image
  - docker push <image>

Lab:

Exercise 1: Build and run a simple Java SE app
Exercise 2: Update app 1661_docker_build_your_own.zip

Dockerfile Commands

Dockerfile commands

- FROM <name-of-the-base-image>
- MAINTAINER: author of the Dockerfile
- RUN <Shell-cmmand>: Any OS command to build the image
- ENTRYPOINT: Specifies different entry point other than /bin/sh -c
- COPY
- WORKDIR
- CMD: Command to be executed when container is started
- EXPOSE: Expose the specified port to the host machine

FROM <name-of-the-base-image> command

- Defines your base layer
  - Images are created in layers, which means you can use another image as the base image for your own
- Starts the Dockerfile
  - Dockerfile must start with the FROM command
- Example
  - FROM scratch
  - FROM ubuntu
  - FROM sangshin/baseimage:latest

RUN <Shell command> command

- Any shell command used to build up the Image you're creating
  - For each RUN command, Docker will run the shell command then create a new layer of the image
  - This way you can roll back your image to previous states easily
- The shell command after the RUN (e.g., RUN mkdir /user/local/foo) runs in a /bin/sh shell as a default
  - You can define a different shell like this: RUN /bin/bash -c 'mkdir /user/local/foo'
- Examples
  - RUN mkdir /user/local/foo
  - RUN /bin/bash -c 'mkdir /user/local/foo'

ENTRYPOINT command

- Any s

COPY command

- Copies local files into the container
- Examples
  - COPY docker-entrypoint.sh /usr/local/bin/

WORKDIR command

- Sets the working directory for any RUN, CMD, ENTRYPOINT, COPY, and ADD directives subsequent in the Dockerfile
- Examples
  - ???

CMD command

- Defines the commands that will run on the Image at start-up
  - > Unlike a RUN, this does not create a new layer for the Image, but simply runs the command
  - > There can only be one CMD per a Dockerfile/Image
- If you need to run multiple commands, the best way to do that is to have the CMD run a script. CMD requires that you tell it where to run the command, unlike RUN
- Examples

CMD ["python", "./app.py"]

CMD ["/bin/bash", "echo", "Hello World"]

EXPOSE command

- Creates a hint for users of an image which ports provide services
  - It is included in the information which can be retrieved via $ docker inspect <container-id>
  - The EXPOSE command does not actually make any ports accessible to the host! Instead, this requires publishing ports by means of the -p flag when using $ docker run
- Examples
  - EXPOSE 3306

Example Dockerfile

- Defin

```
FROM busybox
ENV foo /bar
WORKDIR ${foo}   # WORKDIR /bar
ADD . $foo       # ADD . /bar
COPY \$foo /quux # COPY $foo /quux
```

Lab:

Exercise 3: Simple Java Maven Project
Exercise 4: Spring Boot app
1661_docker_build_your_own.zip

Simple Web application

Steps to create your own image

- Create Dockerfile
- Build the image

  docker build -t cmd .

- Run the image

- Publish the image

What is Docker?

- https://github.com/jiwhiz/spring-boot-docker-mysql

Lab:

Exercise 5: Web Application with Port Binding 1661_docker_build_your_own.zip

Pushing Your Image to Docker Hub

Lab: Push your image to Docker Hub

```
C:\lab>docker tag myjavaapp sangshin/myfirstjavaapp:latest
C:\lab>docker push sangshin/myfirstjavaapp:latest

C:\lab>docker pull sangshin/myfirstjavaapp

C:\lab>docker images

C:\lab>docker run sangshin/myfirstjavaapp
```

Lab:

Exercise 6: Pushing Image to Docker Hub
1661_docker_build_your_own.zip

Code with Passion!
JPassion.com