QA Consulting

Docker

Dockerfiles

TRAINING MATERIALS - MODULE HANDOUT

Contacts

robert.crutchley@qa.com

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview 1

Usage 1

Basic 1

Use Another Dockerfile 1

Specify an Image Name, Tag & Repository 1

Tasks 2

Overview

We can build our own Docker Images by using Dockerfiles. A Dockerfile contains a list of instructions for creating a new image, effectively a build script for Docker Images. Each instruction in a Dockerfile creates intermediate images and stores them, like a cache. For instance if there are four instructions in a Dockerfile and your build fails on the fourth, when you attempt to build the image again, the build can start on step four, because the previous steps have already been built.

Usage

Basic

The docker **build** command is used to create an image from a Dockerfile. The context, which is the folder in which the Dockerfile will operate in must also be supplied. By default Docker will use the Dockerfile from the given context as well.

```
docker build [CONTEXT]
```

```
docker build .
```

Use Another Dockerfile

You may want to provide a Dockerfile that is in a different location to the context.

```
docker build -f [DOCKERFILE_PATH] [CONTEXT]
```

```
docker build -f /home/bob/Dockerfile .
```

Specify an Image Name, Tag & Repository

The Name, Tag and Repository for an Image in Docker becomes very important when you are managing images. When creating your own image you can set these properties with the **-t** option.

```
docker build -t [REPOSITORY]/[NAME]:[TAG] [CONTEXT]
```

```
docker build -t docker.io/bob/myapp:latest .
```

Tasks

This exercise will get you to take the NGINX Docker Image and change the default index.html file that is served. This change will be packed into your own Docker Image that you can run and view the changes for yourself

- Create a new directory for this exercise: **~/docker/03_dockerfile_exercises**
- Make a Dockerfile in the new directory and enter the contents below

```
FROM nginx:latest
RUN printf "My Custom NGINX Image\n" > /usr/share/nginx/html/index.html
```

- Build the docker image, giving it a suitable name
- Run the image in a container, publishing port 80
- Access the service on localhost in your browser or by using a command line tool such as curl
    - "My Custom NGINX Image" should be displayed
- Stop and remove the container
- Delete the image that you created