QA Consulting

Docker

Registries

TRAINING MATERIALS - MODULE HANDOUT

Contacts

robert.crutchley@qa.com

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview

In Docker, a Registry is a server side application that can store Docker Images for us. This Registries can be accessed remotely, meaning that Docker images can be downloaded from any location. When you run a container with an image that you didn't build, NGINX for instance, the image will be downloaded from a Registry first. The default registry that Docker images will be pulled down from is docker.io, also known as Docker Hub.

Image Naming for Registries

You may have noticed from using Docker in the past, an image name prefixed with **docker.io**, such as **docker.io/nginx:latest**. This is referring to the registry that it has been pulled from and where it would also be pushed to if you wanted to upload the image.

Creating Your Own Registry

Why?

Having your own Registry deployed can contribute really well to your own CI/CD setup. Images can be stored and retrieved quickly as well as having the benefit of not having your images uploaded and stored to someone else's registry. Registries also have the ability to execute Web Hooks which could trigger a deployment for instance when an image has been successfully uploaded to the registry.

How?

There is a prebuilt image that we can use to deploy, so getting a Registry working for a test environment is really as simple as executing a docker run command. The Registry service listens on port 5000.

```
docker run -d -p 5000:5000 --name registry registry
```

Tasks

This exercise will take you through creating your own registry that you can push and pull images to.

Create the Registry

Start by creating the Registry and make sure that the port 5000 has been published, you may use the command shown previously in this document for this.

Upload an Image to the Registry

To upload an image to the Registry, we first need one that has been tagged appropriately. Start by pulling an NGINX down and then retagging it to be **localhost:5000/my/nginx**.

Once you have retagged the image, use the **docker push** command to push the image to the Registry that you have created.

Download an image from the Registry

Delete the NGINX images, both yours (**localhost:5000/my/nginx**) and the official (**nginx**) NGINX image. Now we can run a **docker pull** command to prove that the image has been stored in the Registry that we deployed, remember to pull you docker image as **localhost:5000/my/nginx**.

Integrate into a Build Script

Stop and remove all containers, remove all the images.
Try create a shell script that can pull down a project from GitHub, build the Docker image and push it to the registry.