



Docker Compose

Introduction

TRAINING MATERIALS - MODULE HANDOUT

Contacts

robert.crutchley@qa.com

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

Overview	1
Task	1
Install Docker Compose on Linux	1
Create Your First docker-compose.yaml	2
Run Your Configuration	2
Scale Your Application	2
Destroy the Containers	2

Overview

Compose for Docker is a tool that allows you to define and run multiple Docker containers with a command by using a single configuration file. Compose can improve CI workflows and automation when creating multiple containers, complex multi-tier applications can be built, run and destroyed in Docker with a command.

Task

This exercise covers:

- Installing Docker Compose (if necessary)
Compose is a separate tool from Docker, however it does come with Windows and Mac Docker installations. If you are using Linux then you will need to install it using this guide.
- Trying out Docker Compose
You will be able to create a configuration for Docker Compose which creates an NGINX container and also use Docker Compose to scale the application, making more copies of it.

Install Docker Compose on Linux

If you are on Windows or Mac, skip this section, otherwise use the commands below in your terminal to install Docker Compose. These commands are also available in the Compose documentation:

<https://docs.docker.com/compose/install/>.

```
# download the docker-compose binary
sudo curl -L
"https://github.com/docker/compose/releases/download/1.24.0/docker-compose-
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
# make the binary executable so we can use it
sudo chmod +x /usr/local/bin/docker-compose
```

Docker Compose - Introduction

Create Your First docker-compose.yaml

Create a new directory `~/docker-compose/introduction_exercise`. Docker Compose configurations can be made in a YAML file called `docker-compose.yaml`. YAML (YAML Ain't Markup Language) is mainly used for configurations and has been for many tools such as Docker Compose, one of the main benefits of YAML is its readability over something like JSON and XML. Now let's create the configuration file below, which will create an NGINX container, publishing port 80 of the container to a random high port on the host.

```
~/docker-compose/introduction_exercise/docker-compose.yaml
```

```
version: "3.3"
services:
  nginx:
    image: nginx:latest
    ports:
      - target: 80
        protocol: tcp
```

Run Your Configuration

Run your configuration with the `docker-compose` command.

```
docker-compose up -d
```

If you view the running containers now, the NGINX container should be there. You will be able to access NGINX from the high port if you would like to see it.

Scale Your Application

We currently have one instance of NGINX, now let's scale it to five using Docker Compose.

```
docker-compose up -d --scale nginx=5
```

When you view the running containers again, you will see there five there now.

Destroy the Containers

Now it's time to tear down all of the NGINX containers that are running using Docker Compose.

```
docker-compose down
```