Micro Services Application Deployment to Docker

Compiled and presented by Nguyen Canh Hat

Table of Contents

I. Introduction

I.1. Target

The document aims at helping readers understand how to deploy application on Docker. It covers some major Docker's aspects such as Docker CLI, Docker image, Dockerfile, Docker Compose, etc.

I.2. Demo Overview

In this demo, we use Docker compose to deploy the application to the local Docker environment. This is really important to experience the way of local Docker deployment rather than the application source code itself.

The project used in this demo consists of 3 services:

- database
- backend
- frontend

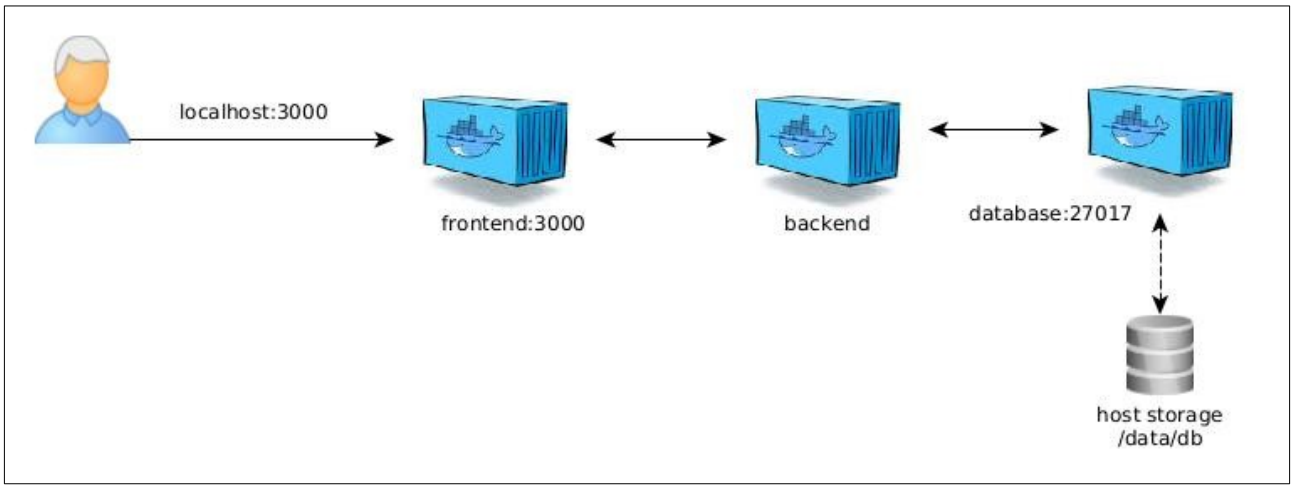

Figure 1: Application Components

The source code can be found in "Demo source code"

I.3. Prerequisite

- Docker is installed

II. Instruction

II.1. General steps

In order to get the application running, following steps are required:

- Build the docker images
- Verify the docker images
- Start the application
- Verify the application

II.2. Execution

- **Build the docker images:**
  - Build frontend docker image: In the root folder, run the following command:
    docker build ./frontend -t frontend

    #The output will be like this:
    docker build ./frontend -t frontend
    Sending build context to Docker daemon 576.5kB
    Step 1/9 : FROM node:lts-buster AS development
    lts-buster: Pulling from library/node
    8eb6dba554cf: Pull complete
    …
    Successfully built 2a76259cf0e7
    Successfully tagged frontend:latest
  - Build backend docker image: In the root folder, run the following command:
    docker build ./backend -t backend

    #The output will be like this:
    docker build ./backend -t backend
    Sending build context to Docker daemon 90.62kB
    Step 1/8 : FROM node:lts-buster-slim AS development
    lts-buster-slim: Pulling from library/node
    90ac1ecaf92c: Pull complete
    …
    Successfully built 609ca47cebb0
    Successfully tagged backend:latest

- **Verify the docker images:**

  # Run the command to verify the docker images
  docker image ls

  #The output will be like this:

```
REPOSITORY   TAG              IMAGE ID       CREATED          SIZE
backend      latest           609ca47cebb0   4 minutes ago    290MB
frontend     latest           2a76259cf0e7   12 minutes ago   1.29GB
node         lts-buster-slim  f653a262fdd3   3 days ago       235MB
node         lts-buster       9a52090cc706   3 days ago       964MB
```

- **Start the application:**

  # In the root folder, Run the command to start the application
  docker-compose up

```
[+] Running 14/14
 ✔ mongo 13 layers [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled
   ✔ 22e816666fd6 Pull complete
   ✔ 079b6d2a1e53 Pull complete
   ✔ 11048ebae908 Pull complete
   ✔ c58094023a2e Pull complete
   ✔ 252003e80cc8 Pull complete
   ✔ 7cb91a976d85 Pull complete
   ✔ 929663192bb1 Pull complete
   ✔ 5af259c6f8d8 Pull complete
   ✔ 44f2ab049616 Pull complete
   ✔ 14f8b9afdeb2 Pull complete
   ✔ 8fd542a9a576 Pull complete
   ✔ d940963cc55f Pull complete
   ✔ 64541ae8fc1a Pull complete
[+] Running 5/5
 ✔ Network src_react-express  Created
 ✔ Network src_express-mongo  Created
 ✔ Container src-mongo-1      Created
 ✔ Container src-backend-1    Created
 ✔ Container src-frontend-1   Created
Attaching to src-backend-1, src-frontend-1, src-mongo-1
src-mongo-1      | 2023-07-08T02:42:43.296+0000 I  CONTROL
```

- **Verify the application:**

  # open browser to access application at URL: localhost:3000

III. Frequently Asked Questions

III.1. Demo source code

- https://github.com/nashtech-garage/kubernetes/tree/master/src

IV. References

- Source code for application: https://github.com/docker/awesome-compose/tree/master/react-express-mongodb