What is docker?

Open-source containerisation platform. It enables developers to package applications into containers. Containers simplify delivery of distributed applications and have become increasingly popular as organisations shift towards cloud native development and hybrid multicloud environments.

How containers work

Containers are made possible by process isolation and virtualisation capabilities built into the linux kernel. These capabilities include control groups (Cgroups) for allocating resources amount processes and namespaces for restricting a processes access or visibility into other resources or areas of the system.

Overall container technology offers all the functionality and benefits of a VM but the advantages are:

* Light weight
* Greater resource efficiency
* Improved developer productivity

6

Why use docker?

Improved and seamless portability

Even lighter weight and more granular updates

Automated container creation

Container versioning

Container reuse

Shared container libraries

Docker tools and terms

DockerFile – every docker container starts with a simple text file containing instructions for how to build the docker image.

Docker images – contain executable application source code as well as all the tools’ libraries and dependencies that the application code needs to run as a container. When you run the docker image it become an instance of the container.

Docker containers – are the live running instances of docker images. While docker images are read-only, containers are live, ephemeral executable content. Users can interact with them, and administrators can adjust their settings and conditions using docker commands.

Docker hub – is the public repository of docker images that calls itself the worlds largest library and community of container images. It holds over 100,000 container images sourced from commercial software vendors, open source projects and individual developers.

All docker hub users can share their images at will. They can also download predefined base images from the docker filesystem to use as a starting point for any containerisation project.

Docker daemon – a service running on your operating system such as Microsoft windows. This service creates and manages your docker images for you using the commands from the client, acting as the control centre of your docker implementation.

Docker registry – Scalable open source storage and distribution system for docker images. The registry enables you to track image versions in repositories, using tagging for identification. This is accomplished using git, a version control tool.

How to create a docker image:

1. Download the App contents
2. Open in your code editor. You should see package.json and two subdirectories (src and spec)
3. Build the app’s container

We will need a Dockerfile

1. Create a file named Dockerfile in the same folder as the file package.json with the following contents:
2. # syntax=docker/dockerfile:1
3. FROM node:12-alpine
4. RUN apk add --no-cache python g++ make
5. WORKDIR /app
6. COPY . .
7. RUN yarn install --production
8. CMD ["node", "src/index.js"]

Make sure that the Dockerfile has no file extension. This is because it can cause problems in the next step.

1. Navigate to the app directory with the Dockerfile in the terminal and run this command:

**Docker build -t getting-started .**

This command used the Dockerfile to build a new container image. It used node:12-alpine image, however, as we didn’t have it on our machine it needed to download the image.

After the image was downloaded, we copied in our application and used yarn to unstall our applications dependencies. The CMD directive specifies the default command to run when starting the container from the image.

-t flag tags our image. This is because so as a human-readable name for the final image. Since we named the image getting-started, we can refer to that image when we run a container.

The . at the end of the docker build command tells that Docker should look for the Dockerfile in the current directory.

1. Start an app container
2. Start the container using docker run command and specify the name of the image we just created:

Docker run -dp 3000:3000 getting started

-dp – means we are running the container in ‘detached’ mode in the background and creating a mapping between the host’s port 3000 to the containers port 3000. Without port mapping we will not be able to access the application.

b) Open your web browser to HTTP://localhost:3000 . you should see the app.

If you go to the docker dashboard, you should be able to see all the containers which you are running.

Database Evaluation

Scores are given by professionals who have used these database platforms before and know what they are doing.

PostgreSQL:

URL: <https://www.postgresql.org/>

**Key data and system requirements:**

Maximum database size: Unlimited

Maximum table size: 32 Tb

Maximum size of data set: 1.6Tb

Maximum field size: 1 Gb

Maximum number of columns: 250 – 1600 (depending on data type)

Maximum number of rows: unlimited

Maximum number of indexes: unlimited

**How does it work:**

Client – server model: the central server is called “postmaster” which manages all the database files and connections that are established for communication. Users only need a suitable client program to establish the connection, whereby the PostgreSQL software package with pSQL already has the native solution integrated for operation via the command line or terminal.

**Advantages and disadvantages:**

Advantages: Disadvantages:

Open source Not available on all hosts by default

Highly expandable Comparatively low reading speed

Cross-platform

Largely compliant with SQL standard

Possible to process complex data types

Flexible full text search

Creation of own functions, triggers, data types

Good language support

* + Python
  + Java
  + Perl
  + PHP
  + C
  + C++
  + Etc
* Supports JSON

**Overall score:**

94

RethinkDB:

URL: <https://rethinkdb.com/>

**Advantages and disadvantages**

Advantages:

Open-source database for web-applications

Json dynamic document is NoSQL

Uses distributed database so easy to scale

Provides high availability database with automatic functions

Disadvantages

Each table requires a minimum of 10Mb disk space

No hard limit for a single

Primary keys are limited

**Usage in RethinkDB**

Feeds real time data into the application

Used for web and mobile applications

Supported web browser to establish the connection using HTTP protocols

**Upon further research** I have found that people are saying that rethinkDB is dead however, it is still being updated. This is however, extremely slow as the last update was in august last year. So due to the score being so high I would not recommend this one.

**Overall Score:**

76

MySQL:

URL: <https://www.mysql.com/>

**Advantages and disadvantages:**

Advantages:

Reduced total cost of ownership

Portability

Seamless connectivity

Rapid development and round the clock uptime

Data security

Disadvantages:

Not very efficient at handling large databases

Doesn’t have a good developing and debugging tool compared to other databases

Prone to data corruption as its inefficient in handling transactions

Does not support SQL check constraints

Refer to excel spread sheet.

PostgreSQL:

Support: PostgreSQL has a wide range of support options available. Ranging from an intensive documentation, community help, IRC chat and slack.

Cost: Free, open source

Scalability: Extensive documentation explaining how to scale up, however, its fairly hard to do by yourself

Migration: Extensive documentation explaining how to migrate databases.

Speed/Security: Has few documented security vulnerabilities per the CVE database.

RethinkDB:

Support: Has had no updates since mid 2020.

Cost: Free

Scalability:

Migration: 1.16 or higher is handled automatically. Documentation on how to do this.

Speed/Security: Has had no updates since mid 2020.

MySQL:

Support: Extensive documentation, 24/7 support line

Cost: Free, open source

Scalability: Real-time in-memory, acid-compliant transactional database, combining 99.999% availability with the low TCO of open-source

Migration: Workbench wizard, which does it in minutes rather that hours.

Speed/security: Robust data security to protect data including secure connections, authentication servers, fine grained authorisation and controls and data encryption.