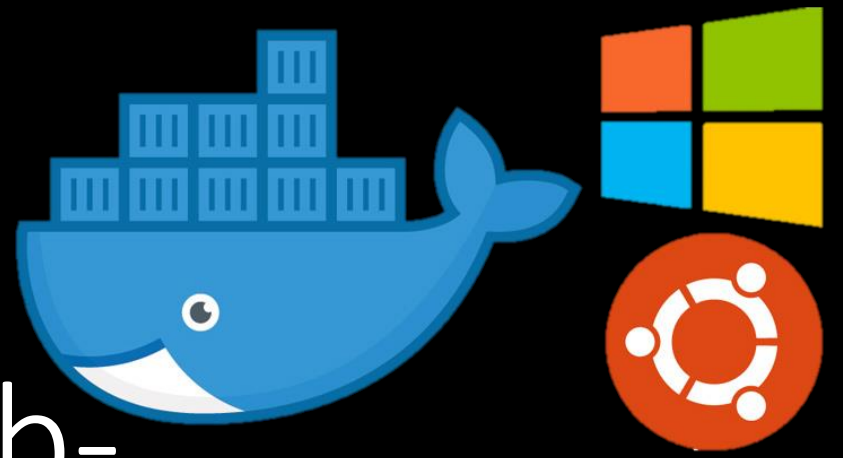

Разработка web-приложений

Docker

Цыгулин Алексей Александрович к.т.н.

История появления

Вам нужен сервер?

- Купить сервер
- Арендовать сервер
- Арендовать виртуальный сервер
- Отправить контейнер в облако

Что такое контейнеры

- Файловая система (chroot)
- Процессы, память, устройства,… (cgroups)
- Управление образами и экземплярами

Docker images

- docker run --interactive --tty alpine:3.10
- docker run -it alpine:3.10

- docker run --detach -it ubuntu:bionic
- docker run -dit ubuntu:bionic

- docker ps

Node.js в Docker

- docker run -it node:12-stretch bash

- docker run node:12-stretch cat /etc/issue

Docker cli

- docker pull node:12
- docker run -it node:12 ls
- docker inspect node
- docker ps
- docker pause <ID or name>
- docker unpause <ID or name>
- docker kill <ID or name>
- docker exec <ID or name> ps aux
- docker top
- docker logs <id from previous command>
- docker search node

Dockerfile

```
FROM node:12-stretch

CMD ["node", "-e", "console.log(\"hi NSTU\")"]
```

```
docker build .
```

```
docker build . --tag my-node-app:1
docker run my-node-app:1
```

Node.js Docker

```
const http = require("http");
http.createServer(function(request, response) {
    console.log("request received");
    response.end(“hi NSTU", "utf-8");
  }).listen(3000);
console.log("server started");
```

```
FROM node:12-stretch
COPY index.js index.js
CMD ["node", "index.js"]
```

```
docker build -t my-node-app .
docker run --init --publish 3000:3000 my-node-app
```

Node.js Docker

```
FROM node:12-stretch

USER node

WORKDIR /home/node/code

COPY --chown=node:node index.js .

CMD ["node", "index.js"]
```

Express

```
const express = require("express");
const app = express();
app.get("/", function(request, response){

  response.send("<h2>Привет Express!</h2>");
});
app.listen(3000);
```

```
npm init -y
npm install express
```

Docjerfile

```
FROM node:12-stretch
USER node
RUN mkdir /home/node/code
WORKDIR /home/node/code
COPY --chown=node:node . .
RUN npm ci
CMD ["node", "index.js"]
```

```
.dockerignore
node_modules/
.git/
```

Alpine Linux

FROM node:12-alpine

```
USER node
RUN mkdir /home/node/code
WORKDIR /home/node/code
COPY --chown=node:node package-lock.json package.json ./
RUN npm ci
COPY --chown=node:node . .
CMD ["node", "index.js"]
```

Размер образа (сравнивая Size поле в docker inspect my-app) уменьшился с 913 МБ до 86 МБ.

Многоступенчатые сборки

```
# build stage
FROM node:12-stretch
WORKDIR /build
COPY package-lock.json package.json ./
RUN npm ci
COPY . .

# runtime stage
FROM node:12-alpine
USER node
RUN mkdir /home/node/code
WORKDIR /home/node/code
COPY --from=0 --chown=node:node /build .
CMD ["node", "index.js"]
```

Статический сайт

```
FROM node:latest
WORKDIR /app
COPY . .
RUN npm ci && npm run build

FROM nginx:latest
COPY --from=0 /app/build /usr/share/nginx/html
```

Теперь, если вы запустите это, оно должно работать:

```
docker build -t static-app .
docker run -p 8080:80 static-app
```

Mount

```
docker run --mount
type=bind,source="$(pwd)"/build,target=/usr/share/nginx/html -p
8080:80 nginx
```

```
docker run --env DATA_PATH=/data/num.txt --mount
type=volume,src=incrementor-data,target=/data incrementor
```

Сеть с Docker

```
docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
xxxxxxxxxxxx        bridge              bridge              local
xxxxxxxxxxxx        host                host                local
xxxxxxxxxxxx        none                null                local
```

```
docker network create --driver=bridge app-net
```

```
docker run -d --network=app-net -p 27017:27017 --name=db --rm mongo:3
```

```
docker run -it --network=app-net --rm mongo:3 mongo --host db
```

Node.js mongo

```
const app = require("express")();
const { MongoClient } = require("mongodb");
async function start() {
  const client = await MongoClient.connect("mongodb://localhost:27017");
  const db = client.db("dockerApp");   const collection = db.collection("count");
  app.get("/", async (req, res) => {
     const count = await collection.count(); res.send(`<h2>${count}</h2>`);});
  app.get( "/add",  async (req, res)=> {
      const dbres = await collection.insertOne({}); res.send(`<h2>+${dbres.insertedCount }</h2>`);});
  app.listen(3000);
}
start()
```

Node.js mongo start

npm install mongodb@3.3 # you need to add mongodb to your project

docker build --tag=my-app-with-mongo .

docker run -p 3000:3000 --network=app-net my-app-with-mongo

Ресурсы

https://dker.ru/

https://www.docker.com/

https://hub.docker.com/

https://btholt.github.io/complete-intro-to-containers