Инструкции

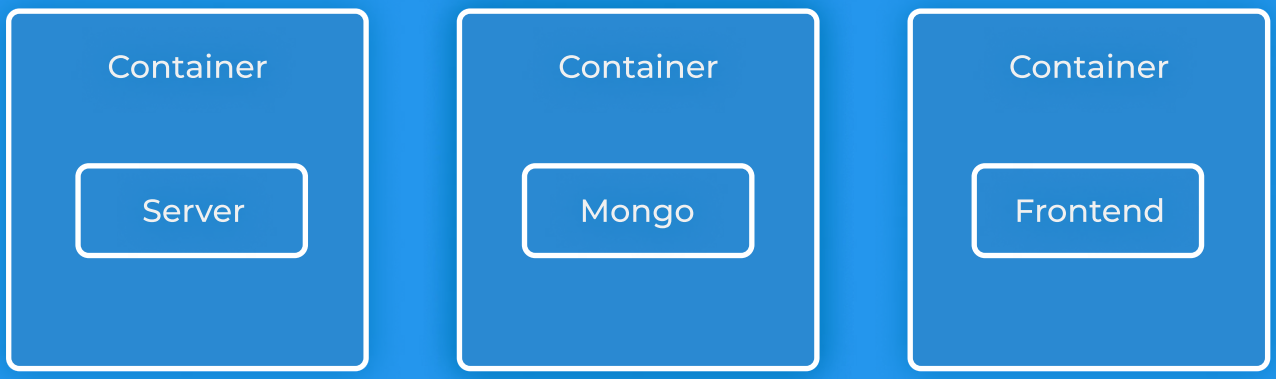
Docker Практикум

Владилен Минин
VLADILEN.RU

Исходный код

https://github.com/vladilenm/docker-mern

Задача



План

- Положить в Docker MongoDB
- Положить в Docker NodeJS (Backend)
- Положить в Docker React (Frontend)
- Создать development режим сборки
- Создать production режим сборки
- Запустить все на VPS

Владилен Минин
VLADILEN.RU

Действия

Mongo

Берём образ БД от сюда : https://hub.docker.com/_/mongo
Запускаем контейнер:

```
docker run -d -p 27017:27017 --rm --name mongodb mongo
```

Backend

Создаем `/server/Dockerfile`

```
FROM node:16
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
ENV PORT 5000
EXPOSE $PORT
CMD ["node", "app.js"]
```

Собираем образ и запускаем контейнер

```
docker build -t notes-backend .
```

```
docker run -d -p 5000:5000 --rm --name notes-backend notes-backend
```

Меняем в app.js Uri → mongodb://host.docker.internal:27017/notes
Пересобираем

```
docker build -t notes-backend .
```

```
docker run -d -p 5000:5000 --rm --name notes-backend notes-backend
```

Стартуем клиента, все работает

Владилен Минин
VLADILEN.RU

Frontend

Создаем `/client/Dockerfile`

```
FROM node:16
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
ENV PORT 3000
EXPOSE $PORT
CMD ["node", "app.js"]
```

Собираем образ и запускаем контейнер

```
docker build -t notes-frontend .
```

```
docker run -d -p 3000:3000 --rm --name notes-frontend notes-frontend
```

Добавляем в одну сеть

Создаем `docker network create notes-net`

Mongo:

1. `docker run -d --network notes-net --rm --name mongodb mongo`
2. `docker run -d -v data:/data/db --network notes-net --rm --name mongodb mongo`

Backend:

1. Меняем MongoUri → mongodb://mongodb:27017/notes
2. Добавляем CORS (npm i cors)
3. `docker build -t notes-backend .`
4. `docker run -d --network notes-net -p 5000:5000 --rm --name backend notes-backend`

Frontend:

1. Добавляем baseURL в axios → http://localhost:5000
2. `docker run -d -p 3000:3000 --rm --name frontend notes-frontend`

Владилен Минин
VLADILEN.RU

Docker Compose

Удаляем все контейнеры и образы

Добавляем docker-compose.yml в корень

```
version: "3.8"
services:
   frontend:
      image: "notes-frontend-dev"
      build: ./client
      ports:
         - "3000:3000"
      volumes:
         - ./client/src:/app/src
   backend:
      image: "notes-backend-dev"
      build:
         context: ./server
         dockerfile: Dockerfile
      ports:
         - "5000:5000"
      volumes:
         - ./server:/app
         - /app/node_modules
      env_file:
         - ./config/server.env
      depends_on:
         - mongodb
   mongodb:
      image: "mongo"
      volumes:
         - data:/data/db
volumes:
   data:
```

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

```
docker-compose down
```

Владилен Минин
VLADILEN.RU

Production сборка

```
version: "3.8"
services:
   mongodb:
      image: "mongo"
      volumes:
         - mongo-prod:/data/db
      env_file:
         - ./config/production.env
   frontend:
      image: "notes-frontend-prod"
      build:
         context: ./client
         dockerfile: Dockerfile.production
      ports:
         - "80:80"
      env_file:
         - ./config/production.env
   backend:
      image: "notes-backend-prod"
      build:
         context: ./server
         dockerfile: Dockerfile.production
      ports:
         - "5000:5000"
      depends_on:
         - mongodb
      env_file:
         - ./config/production.env
volumes:
   mongo-prod:
```

Владилен Минин
VLADILEN.RU

Client

```
FROM node:16-alpine AS builder
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
RUN npm run build
FROM nginx:alpine
COPY nginx.conf /etc/nginx/conf.d/default.conf
COPY --from=builder /app/build /usr/share/nginx/html
CMD ["nginx", "-g", "daemon off;"]
```

```
server {
    listen 80;
    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
        try_files $uri $uri/ /index.html;
    }
}
```

Владилен Минин
VLADILEN.RU

Server

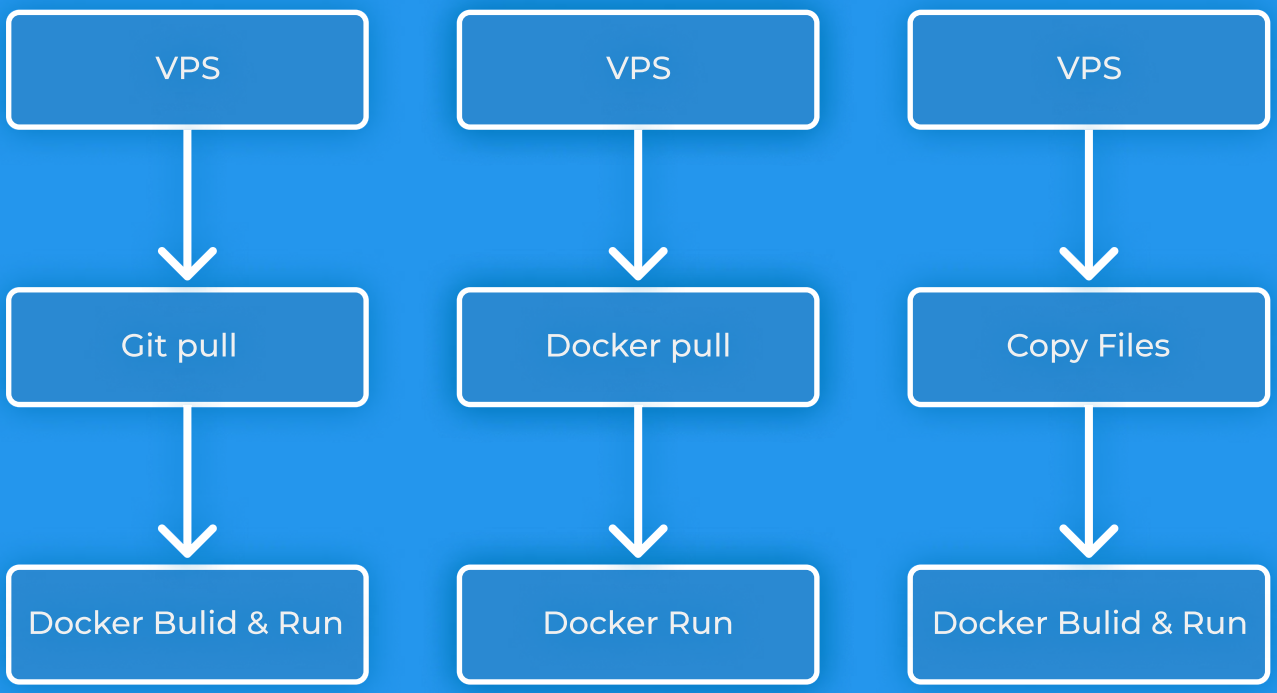
```
FROM node:16-alpine
WORKDIR /app
COPY package.json .
RUN npm install
COPY . .
EXPOSE 5000
CMD ["npm", "start"]
```

Далее скрипт по запуску продашн мода:

```
docker-compose -f docker-compose.production.yml up -d
```

Открываем localhost.

Deploy



Выбираем одну из опций и заливаем на VPS

Как создать SSH ключ

```
ssh-keygen -t rsa
```

```
pbcopy < ~/.ssh/id_rsa.pub
```

Финальное приложение

https://github.com/vladilenm/docker-mern/tree/completed