

# Переменные окружения

Настройка окружения:

```
cd docker_images
mkdir env
cd env
```

Использование флага `--env` для передачи переменной окружения при сборке образа:

```
--env [KEY]=[VALUE]
```

Использование инструкции ENV в Dockerfile:

```
ENV [KEY]=[VALUE]
ENV [KEY] [VALUE]
```

Клонирование `weather-app`:

```
git clone https://github.com/linuxacademy/content-weather-app.git src
```

Создание Dockerfile:

```
vi Dockerfile
```

Содержимое Dockerfile:

```
# Create an image for the weather-app
FROM node
LABEL org.label-schema.version=v1.1
ENV NODE_ENV="development"
ENV PORT 3000

RUN mkdir -p /var/node
ADD src/ /var/node/
WORKDIR /var/node
RUN npm install
EXPOSE $PORT
CMD ./bin/www
```

Создание контейнера `weather-app`:

```
docker image build -t linuxacademy/weather-app:v2 .
```

Инспектирование контейнера для просмотра переменных окружения:

```
docker image inspect linuxacademy/weather-app:v2
```

Развертывание приложения `weather-dev`:

```
docker container run -d --name weather-dev -p 8082:3001 --env PORT=3001  
linuxacademy/weather-app:v2
```

Инспектирование development-контейнера для просмотра переменных окружения:

```
docker container inspect weather-dev
```

Развертывание `weather-app` на production:

```
docker container run -d --name weather-app2 -p 8083:3001 --env PORT=3001 --env  
NODE_ENV=production linuxacademy/weather-app:v2
```

Инспектирование production-контейнера для просмотра переменных окружения:

```
docker container inspect weather-app2
```

Получение логов для `weather-app2`:

```
docker container logs weather-app2
```

```
docker container run -d --name weather-prod -p 8084:3000 --env NODE_ENV=production  
linuxacademy/weather-app:v2
```