Docker, CI, Travis

Patka Zsolt-András

2020.03.16

Sapientia - Computer Science BSc.

What is this presentation about?

- Continous Integration
  - CI CD
- Docker
  - Dockerfile
  - Docker image
  - Docker container
- How to: CI
  - Travis
  - Github Actions
- Live demo
  - Backend
  - Frontend
- Advices

Continous Integration

Continous Integration: CI CD

Continous Integration Continous Delivery

1. Plan
2. Code
3. Build
4. Test
5. Release
6. Deploy
7. Operate
8. Measure
9. Repeat

Docker

Docker: Dockerfile

- How should our image look like?
- Analog: Source code for Windows

Docker: Docker image

- Result of building the Dockerfile
- Contains everything needed to spin up your application
- Analog: CD containing Windows

Docker: Docker container

- Result of starting a docker image
- Is alive, runs your application
- Analog: Your PC after you've installed Windows

How to: CI

How to: CI: Travis

1. Free (has a paid version as well)
2. Can easily be integrated with Github
3. Uses Docker containers internally
4. .travis.yml: contains build steps to be executed on each push

```
language: java
services:
 - docker
install: true
os: linux
dist: trusty
jdk: openjdk8
before_script: cd backend
script:
 - ./gradlew build &&
 ./gradlew test && ./gradlew bootJar
after_success:
 - docker build -t demobackend .
 - docker tag demobackend
 "$DOCKER_USERNAME"/demobackend:latest
 - echo "$DOCKER_PASSWORD" |
 docker login -u "$DOCKER_USERNAME" --password-stdin
 - docker push "$DOCKER_USERNAME"/demobackend
```

How to: CI: Github Actions

1. Free
2. It's trivial to integrate it with Github
3. Uses Docker containers internally
4. Easy to use

```
on:
    push:
    branches:
        - develop
        - master
    pull_request:
    branches:
        - master
jobs:
    build:
    runs-on: windows-latest
    steps:
    - uses: actions/checkout@v1
    - name: Set up Python 3.6.8
        uses: actions/setup-python@v1
        with:
        python-version: 3.6.8
    - name: Install dependencies
        run: |
        python -m pip install --upgrade pip
    - name: Test with pytest
        working-directory: src
        run: |
        pip install pytest
        pytest
```

Live demo

Live demo: Backend

- Spring boot app
- Has two simple endpoints
- Travis automatically (on each push):
  1. Runs the build
  2. Runs the tests
  3. Creates a .jar file
  4. Build the Dockerfile
  5. Pushes the Docker image to DockerHub

Live demo: Frontend

- React
- Can send a request to the backend
- Travis automatically (on each push):
  1. Runs the build
  2. Runs the tests
  3. Build the Dockerfile (two stage build)
  4. Pushes the Docker image to DockerHub

Advices

Advices: Docker commands cheatsheet

You can find a cheat sheet containing commonly used Docker commands:

`https://github.com/andraspatka/Presentations/blob/master/docker/cheatSheet.md`

Thank you for your attention!

Sources

- https://mherman.org/blog/dockerizing-a-react-app/