



AARHUS UNIVERSITET

Microservices and DevOps

DevOps and Container Technology

Docker Exercise

Henrik Bærbak Christensen

Exercise 1

- Go over the exercise at
 - https://github.com/docker-for-data-science/docker-for-data-science-tutorial/tree/master/exercises/02_dockerfile_essentials
- *Create a python 'hello world' program, copy it to the container in folder '/app' and run it upon container startup.*

Exercise 2

- Create a 'Hello (yourname)' web server container.
 - Base it upon base image: henrikbaerbak/jdk8-gradle
 - Make it contain a /root/hello WORKDIR
 - In this, have the following 'build.gradle' file (in /root/hello)
 - And have 'HelloSpark.java' file (in /root/hello/src/main/java/example/)

```
/* Build file hello-spark */

plugins {
    id 'java'
}

repositories {
    mavenCentral()
}

dependencies {
    // Use the SparkJava web server framework
    implementation group: 'com.sparkjava', name: 'spark-core',
        version: '2.9.3'
}

task hello(type: JavaExec) {
    group 'SWEA'
    description 'Dependency management demonstration'

    main = 'example.HelloSpark'
    classpath = sourceSets.main.runtimeClasspath
}
```

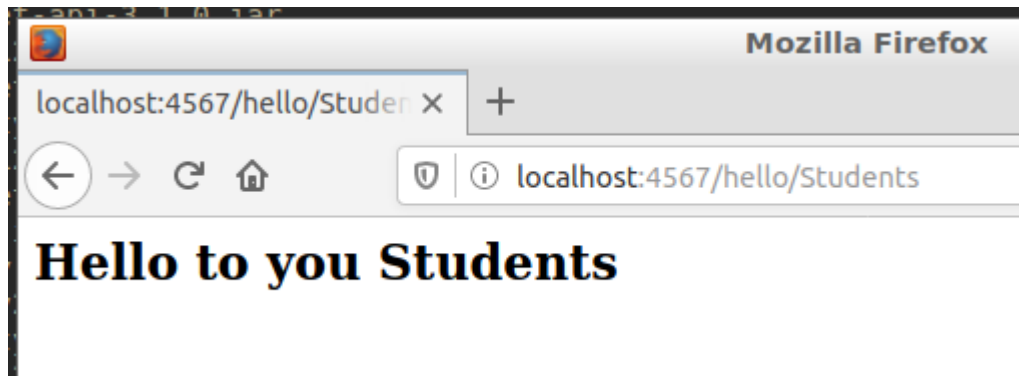
```
package example;

import static spark.Spark.*;

public class HelloSpark {
    public static void main(String[] args) {
        get("/hello/:name",
            (req, res) -> "<H2>Hello to you "+req.params(":name")+"</H2>");
    }
}
```

Exercise 2

- When run, do run 'gradle hello'
- Output required:
 - Open firefox on 'localhost:4567/hello/(name)'





Exercise 3

- Create your own Docker hub account and upload the previous images.
- Let your team mates pull and run your uploaded images.



Exercise 4

- Start on the 'image-dockerfile' mandatory exercise...
- And the 'volume-mount' exercise