EST – IPCB

LICENCIATURA EM ENGENHARIA INFORMÁTICA

Aplicações Distribuídas

Deployment e "*Dockerização*" de Aplicações Spring Boot e Microserviços

3º Ano / 1º Semestre – 2022/2023

Versão 1 dezembro de 2022

Atividade Prática n.º9 – Deployment and Dockerizing Spring Boot Apps e Microserviços

Ao longo desta atividade, pretende-se que ao realizar os passos seguintes experimente o deployment de aplicações Spring Boot usando Contentores Docker.

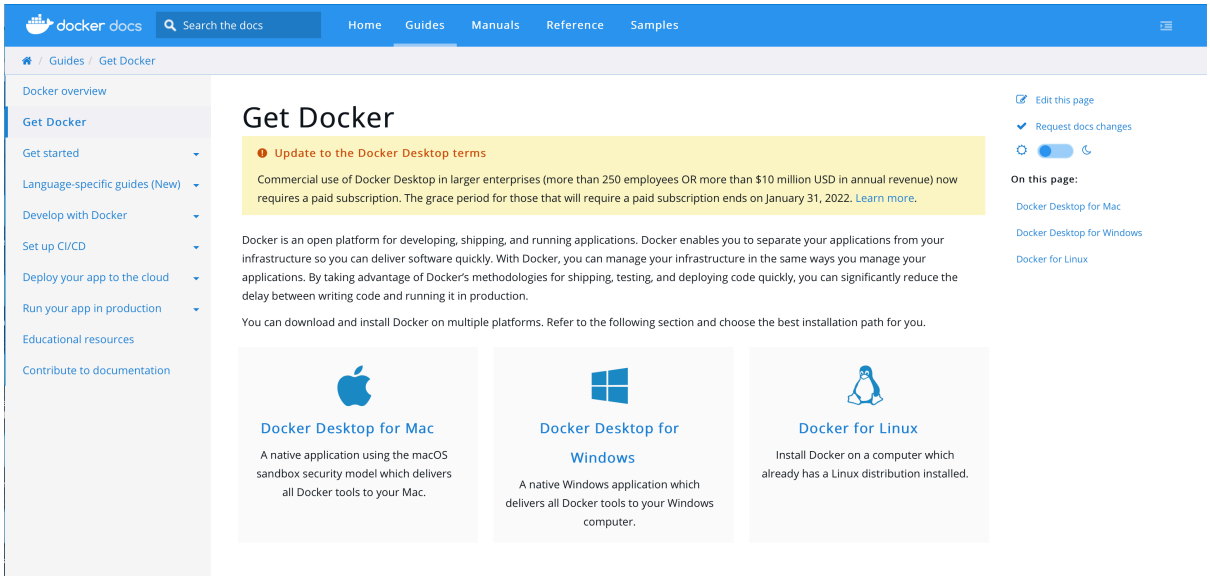
Para melhor compreensão dos conteúdos deste guião sugere-se a consulta do bloco de slides: AD-2023-Modulo-7.1_Docker_e_Dockerizing_Spring_Boot_Apps.pdf.

Durante a realização desta atividade serão implementados ou concluída a implementação de pequenos projetos cada correspondente a um microserviço.

PARTE 1: DEPLOYMENT OF SPRING BOOT APPLICATIONS

Step 1: Install Docker Desktop on Windows, Mac or Linux

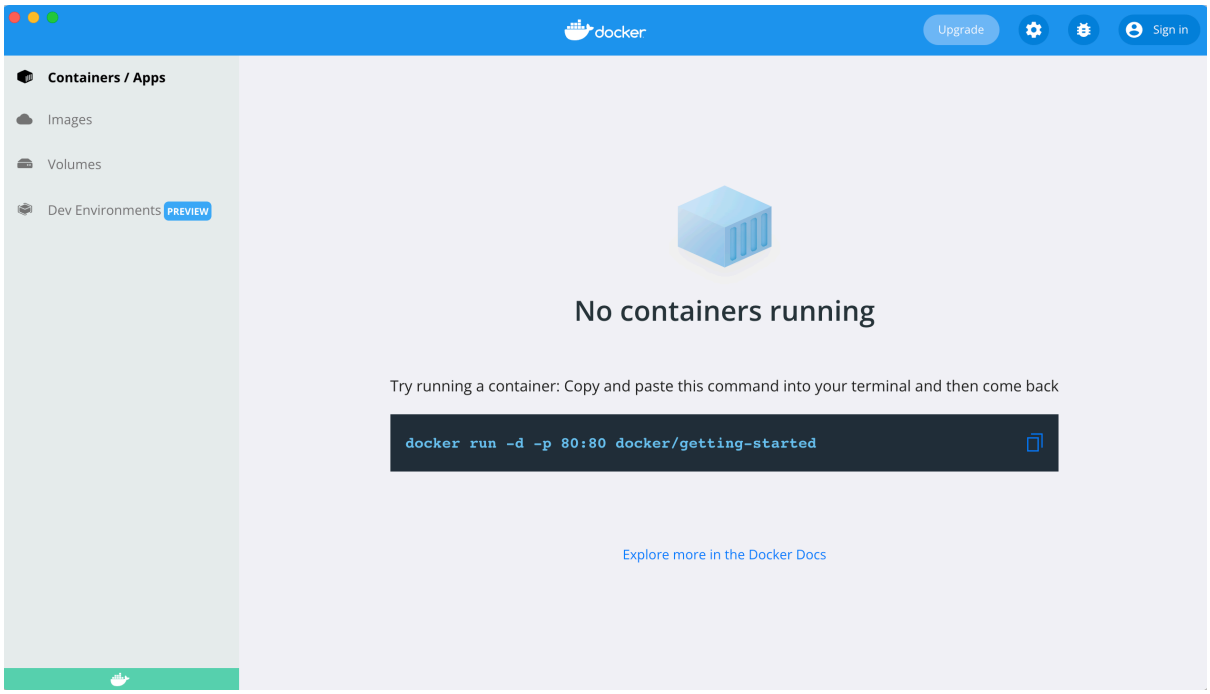
- Go to Docker Documentation searching the Web or docs.docker.com/install
- Select Download and Install



Step 2: Start Docker

- On launch, Docker may ask Admin Password to install network components

"Beginning on August 31, 2021 You must agree to the Docker Subscription Service Agreement to continue using Docker Desktop"



Step 3: Deploying a Spring Boot Application

- Open terminal window, check docker version and deploy a Spring Boot Application as a Container

```
docker --version

docker run -p8080:8080 adfonte/aid-mdssi:latest
```

```
alexandrefonte — com.docker.cli ◂ docker run -p8080:8080 adfonte/aid-mdssi:latest — 114×24

afs-MacBook-Air:~ alexandrefonte$ docker --version
Docker version 20.10.13, build a224086
afs-MacBook-Air:~ alexandrefonte$ docker run -p8080:8080 adfonte/aid-mdssi:latest

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.4)

2022-04-08 11:41:01.637  INFO 1 --- [           main] c.e.h.HelloAidServerApplication          : Starting HelloAid
ServerApplication v0.0.1-SNAPSHOT using Java 1.8.0_322 on 3d24f42c431e with PID 1 (/hello-aid-server.jar started b
y root in /)
2022-04-08 11:41:01.661  INFO 1 --- [           main] c.e.h.HelloAidServerApplication          : No active profile
 set, falling back to 1 default profile: "default"
2022-04-08 11:41:05.556  INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialize
d with port(s): 8080 (http)
2022-04-08 11:41:05.602  INFO 1 --- [           main] o.apache.catalina.core.StandardService   : Starting service
[Tomcat]
2022-04-08 11:41:05.604  INFO 1 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet
engine: [Apache Tomcat/9.0.58]
```

Ctrl+C stop a container

There is an internal docker network:

`docker run -p {HostPort}:{ContainerPort} {docker-image}:{version}`

Step 3: Deploy a Spring Boot Application (Cont.)

- Go to the Browser and Enter the following URI

  http://localhost:8080/messages/students

Recommendations for Windows Users

- If you are using Windows, make sure you are using PowerShell
- If you are using Windows 10 use IP 192.168.99.100 instead of localhost. Use command docker-machine ip (maybe you need to install this command)

Step 4: Go to Docker Registry **https://hub.docker.com**

Docker Registry has many repositories, each contains applications and different versions of applications

- Deploy two Web Services
  - Apache Web Service
  - Tutum

```
docker pull tutum/hello-world
docker run -p80:80 tutum/hello-world:latest
docker pull httpd
docker run -p81:80 httpd:latest
```

Images vs Containers

- Image - a static template with instructions for creating a Container
- Container is the Running version of an image
- Image is like a Class; Container is like an Object!

Step 5: Run in Background two Containers from the same image (e.g., adfonte/aid-mdssi)

- Open a new terminal window
- Use Option -d to run in background

```
alexandrefonte — -bash — 142×24
Last login: Fri Apr  8 12:00:58 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
afs-MacBook-Air:~ alexandrefonte$ docker run -p8080:8080 -d adfonte/aid-mdssi:latest
d035ec1af08da3fbdf0b3c3b3f2c49f334a7853190c107f16471f264bdf77c59
afs-MacBook-Air:~ alexandrefonte$ docker container ls
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS                    NAMES
d035ec1af08d   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   17 seconds ago   Up 15 seconds   0.0.0.0:8080->8080/tcp   strange_cannon
afs-MacBook-Air:~ alexandrefonte$ docker run -p8081:8080 -d adfonte/aid-mdssi:latest
d4e15c186c5ace5b7d2e225a2db64087cf5198641b47b5d52343dd33b063ffc7
afs-MacBook-Air:~ alexandrefonte$ docker container ls
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS                    NAMES
d4e15c186c5a   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   4 seconds ago    Up 2 seconds    0.0.0.0:8081->8080/tcp   gallant_dhawan
d035ec1af08d   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   40 seconds ago   Up 38 seconds   0.0.0.0:8080->8080/tcp   strange_cannon
afs-MacBook-Air:~ alexandrefonte$
```

Step 6: How to Stop a Container

```
docker container stop container id
```

```
alexandrefonte — -bash — 165×24
afs-MacBook-Air:~ alexandrefonte$ docker container ls
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS                    NAMES
d4e15c186c5a   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   3 minutes ago    Up 3 minutes    0.0.0.0:8081->8080/tcp   gallant_dhawan
d035ec1af08d   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   3 minutes ago    Up 3 minutes    0.0.0.0:8080->8080/tcp   strange_cannon
afs-MacBook-Air:~ alexandrefonte$ docker container ls -a
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS                      PORTS                    NAMES
d4e15c186c5a   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   3 minutes ago    Up 3 minutes                0.0.0.0:8081->8080/tcp   gallant_dhawan
d035ec1af08d   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   3 minutes ago    Up 3 minutes                0.0.0.0:8080->8080/tcp   strange_cannon
3d24f42c431e   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   11 minutes ago   Exited (130) 4 minutes ago                           zealous_northcutt
5a309ffd8f59   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   12 minutes ago   Created                                              loving_tereshkova
afs-MacBook-Air:~ alexandrefonte$ docker container stop d4e15
d4e15
afs-MacBook-Air:~ alexandrefonte$ docker container ls -a
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS                      PORTS                    NAMES
d4e15c186c5a   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   3 minutes ago    Exited (143) 3 seconds ago                           gallant_dhawan
d035ec1af08d   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   4 minutes ago    Up 4 minutes                0.0.0.0:8080->8080/tcp   strange_cannon
3d24f42c431e   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   12 minutes ago   Exited (130) 4 minutes ago                           zealous_northcutt
5a309ffd8f59   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   12 minutes ago   Created                                              loving_tereshkova
afs-MacBook-Air:~ alexandrefonte$ docker container ls
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS                    NAMES
d035ec1af08d   adfonte/aid-mdssi:latest  "java -jar hello-aid…"   4 minutes ago    Up 4 minutes    0.0.0.0:8080->8080/tcp   strange_cannon
afs-MacBook-Air:~ alexandrefonte$
```

Step 7: Playing with Docker Images (Commands and Tags)

- Pull **`mysql`** and **`mysql:oracle`** images and Observe Image Tags

`docker pull` *{image}:{tag}*

```
alexandrefonte — com.docker.cli ◂ docker pull mysql — 165×24
afs-MacBook-Air:~ alexandrefonte$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
adfonte/aid-mdssi   latest    437d42c0e475   28 minutes ago   544MB
hello-aid-server    latest    437d42c0e475   28 minutes ago   544MB
message-server      latest    d391b75136db   20 hours ago     544MB
product-server      latest    809354f1ad6f   20 hours ago     544MB
afs-MacBook-Air:~ alexandrefonte$ docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
f003217c5aae: Extracting [=====================>                          ]  12.39MB/27.15MB
65d94f01a09f: Download complete
43d78aaa6078: Download complete
a0f91ffbdf69: Download complete
59ee9e07e12f: Download complete
04d82978082c: Download complete
```

```
alexandrefonte — -bash — 165×24
afs-MacBook-Air:~ alexandrefonte$ docker pull mysql:oracle
oracle: Pulling from library/mysql
e4430e06691f: Pull complete
99677fe408e0: Pull complete
911aa3c638bf: Pull complete
2a50ae0f3675: Pull complete
6e229cecb99f: Pull complete
c1790c22e04b: Pull complete
d99e6dd61db9: Pull complete
45eb6c638025: Pull complete
73dc8a66483d: Pull complete
52190d1e6a3f: Pull complete
Digest: sha256:041a3f0b3d58771afc64762b46289385f2b9126a14ab7b5895e83b87fd26f1bc
Status: Downloaded newer image for mysql:oracle
docker.io/library/mysql:oracle
afs-MacBook-Air:~ alexandrefonte$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED          SIZE
adfonte/aid-mdssi   latest    437d42c0e475   33 minutes ago   544MB
hello-aid-server    latest    437d42c0e475   33 minutes ago   544MB
message-server      latest    d391b75136db   20 hours ago     544MB
product-server      latest    809354f1ad6f   21 hours ago     544MB
mysql               oracle    0ca4d64ca878   8 days ago       435MB
mysql               latest    667ee8fb158e   9 days ago       521MB
afs-MacBook-Air:~ alexandrefonte$
```

Step 8: Playing with Docker Containers (commands)

- Commands related with Containers:

```
afs-MacBook-Air:~ alexandrefonte$ docker container

Usage:  docker container COMMAND

Manage containers

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  inspect     Display detailed information on one or more containers
  kill        Kill one or more running containers
  logs        Fetch the logs of a container
  ls          List containers
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  prune       Remove all stopped containers
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  run         Run a command in a new container
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  wait        Block until one or more containers stop, then print their exit codes

Run 'docker container COMMAND --help' for more information on a command.
```

- Play with the following commands:

`docker container ls`

`docker container run` or **`Docker run`** creates a Container

`docker container pause` `<id container>`

`docker container unpause` `<id container>`

`docker container inspect` `<id container>`

```
alexandrefonte — -bash — 195×29
2022-04-04 13:08:53.000  INFO 1 --- [           main] c.i.r.w.r.RestfulWebServicesApplication  : Started RestfulWebServicesApplication in 10.087 seconds (JVM running for 11.156)
^CADF-iMac:~ alexandrefonte$ dockecontainer unpause b0e48SE
b0e48
ADF-iMac:~ alexandrefonte$ docker container inspect b0e48
[
    {
        "Id": "b0e4882e67c82bdbbe4a418bf8eab935b01276364a04c8b80e97df00c58392a7",
        "Created": "2022-04-04T13:08:41.3399175Z",
        "Path": "sh",
        "Args": [
            "-c",
            "java $JAVA_OPTS -Djava.security.egd=file:/dev/./urandom -jar /app.jar"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 3685,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2022-04-04T13:08:41.8255284Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:f8049a029560296f2c8d98a9668672ed9db1fc85ec0054f9fd9956ae79bf8827",
        "ResolvConfPath": "/var/lib/docker/containers/b0e4882e67c82bdbbe4a418bf8eab935b01276364a04c8b80e97df00c58392a7/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/b0e4882e67c82bdbbe4a418bf8eab935b01276364a04c8b80e97df00c58392a7/hostname",
```

docker container stop => sigterm => Graceful shutdown

```
afs-MacBook-Air:~ alexandrefonte$ docker container ls -a
CONTAINER ID   IMAGE                      COMMAND                  CREATED          STATUS                       PORTS                    NAMES
d4e15c186c5a   adfonte/aid-mdssi:latest   "java -jar hello-aid…"   16 minutes ago   Exited (143) 13 minutes ago                           gallant_dhawan
d035ec1af08d   adfonte/aid-mdssi:latest   "java -jar hello-aid…"   17 minutes ago   Up 17 minutes                0.0.0.0:8080->8080/tcp   strange_cannon
3d24f42c431e   adfonte/aid-mdssi:latest   "java -jar hello-aid…"   25 minutes ago   Exited (130) 17 minutes ago                           zealous_northcutt
5a309ffd8f59   adfonte/aid-mdssi:latest   "java -jar hello-aid…"   25 minutes ago   Created                                               loving_tereshkova
afs-MacBook-Air:~ alexandrefonte$ docker container stop d035
d035
afs-MacBook-Air:~ alexandrefonte$
```

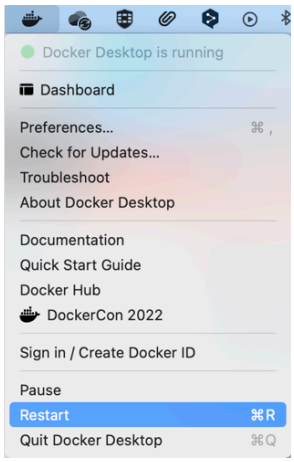
docker container kill

```
afs-MacBook-Air:~ alexandrefonte$ docker container ls
CONTAINER ID   IMAGE                COMMAND                  CREATED          STATUS          PORTS                   NAMES
6cc5cbf98bd4   tutum/hello-world    "/bin/sh -c 'php-fpm…"   18 minutes ago   Up 18 minutes   0.0.0.0:80->80/tcp      hopeful_mcclintock
afs-MacBook-Air:~ alexandrefonte$ docker kill 6cc
6cc
afs-MacBook-Air:~ alexandrefonte$
```

docker container run - p .... --restart=always

```
afs-MacBook-Air:~ alexandrefonte$ docker images
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
docker-aid          latest    81504b630f92   25 hours ago   578MB
adfonte/aid-mdssi   latest    437d42c0e475   33 hours ago   544MB
hello-aid-server    latest    437d42c0e475   33 hours ago   544MB
message-server      latest    d391b75136db   2 days ago     544MB
product-server      latest    809354f1ad6f   2 days ago     544MB
mysql               oracle    0ca4d64ca878   9 days ago     435MB
mysql               latest    667ee8fb158e   11 days ago    521MB
httpd               latest    118b6abfbf55   11 days ago    144MB
tutum/hello-world   latest    31e17b0746e4   6 years ago    17.8MB
afs-MacBook-Air:~ alexandrefonte$ docker container run -p8080:8080 adfonte/aid-mdssi:latest --restart=always

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.4)
```



docker container top <container id>

```
^Cafs-MacBook-Air:~ alexandrefonte$ docker container ls -a
CONTAINER ID   IMAGE                      COMMAND                  CREATED          STATUS                       PORTS                    NAMES
16e24102a347   adfonte/aid-mdssi:latest   "java -jar hello-aid…"   7 minutes ago    Exited (130) 6 minutes ago                            naughty_bell
6cc5cbf98bd4   tutum/hello-world          "/bin/sh -c 'php-fpm…"   12 minutes ago   Up 12 minutes                0.0.0.0:80->80/tcp       hopeful_mcclintock
e4d02a3c2bcd   httpd:latest               "httpd-foreground"       25 minutes ago   Exited (0) 24 minutes ago                             zealous_ptolemy
f773db77544f   tutum/hello-world:latest   "/bin/sh -c 'php-fpm…"   26 minutes ago   Exited (0) 25 minutes ago                             peaceful_bose
afs-MacBook-Air:~ alexandrefonte$ docker top 6cc
UID                 PID                 PPID                C                   STIME               TTY                 TIME                CMD
root                3038                3010                0                   19:53               ?                   00:00:00            nginx: master process
f;
root                3072                3038                0                   19:53               ?                   00:00:00            php-fpm: master proce:
m.conf)
nobody              3073                3072                0                   19:53               ?                   00:00:00            php-fpm: pool www
nobody              3074                3072                0                   19:53               ?                   00:00:00            php-fpm: pool www
root                3076                3038                0                   19:53               ?                   00:00:00            tail -F /var/log/ngin:
root                3077                3038                0                   19:53               ?                   00:00:00            nginx: worker process
afs-MacBook-Air:~ alexandrefonte$
```

docker container stats *<container id>*

```
^Cafs-MacBook-Air:~ alexandrefonte$ docker container ls -a
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS                      PORTS                    NAMES
16e24102a347   adfonte/aid-mdssi:latest "java -jar hello-aid…"   7 minutes ago    Exited (130) 6 minutes ago                           naughty_bell
6cc5cbf98bd4   tutum/hello-world        "/bin/sh -c 'php-fpm…"   12 minutes ago   Up 12 minutes               0.0.0.0:80->80/tcp       hopeful_mcclintock
e4d02a3c2bcd   httpd:latest             "httpd-foreground"       25 minutes ago   Exited (0) 24 minutes ago                            zealous_ptolemy
f773db77544f   tutum/hello-world:latest "/bin/sh -c 'php-fpm…"   26 minutes ago   Exited (0) 25 minutes ago                            peaceful_bose
afs-MacBook-Air:~ alexandrefonte$ docker stats 6cc
```

```
alexandrefonte — com.docker.cli ◂ docker stats 6cc — 180×25
CONTAINER ID   NAME                 CPU %     MEM USAGE / LIMIT     MEM %     NET I/O       BLOCK I/O     PIDS
6cc5cbf98bd4   hopeful_mcclintock   0.03%     4.816MiB / 3.843GiB   0.12%     1.23kB / 0B   0B / 8.19kB   6
```

- Limits to 512MB of RAM and 5% of CPU

```
docker run -p 80:80 -m 521m --cpu-quota 5000 -d --restart=always tutum/hello-world:latest
```

```
alexandrefonte — com.docker.cli ◂ docker stats 90bd — 180×25
CONTAINER ID   NAME         CPU %     MEM USAGE / LIMIT   MEM %     NET I/O     BLOCK I/O     PIDS
90bd09fbdc83   keen_wiles   0.04%     4.883MiB / 521MiB   0.94%     946B / 0B   0B / 8.19kB   6
```

Parte II: How to Dockerize Spring Boot Rest Applications

Dockerize uma Aplicação Spring Boot baseada em microserviços que retorna a previsão da temperatura atual para uma determinada capital de distrito.

A aplicação consiste num API Gateway público, num servidor de nomes Eureka e num microserviço privado do tempo.

O microserviço privado do tempo obtém as previsões do tempo a partir da API aberta do IPMA (consultar https://api.ipma.pt). Mais especificamente consultando um ficheiro Json conforme o id da região.

Exemplo de Castelo Branco:

http://api.ipma.pt/open-data/forecast/meteorology/cities/daily/1050200.json

Passo 1: Descarregue do moodle os projetos.

- Artfact ID: FrontEndMVC-tempo-server
- Artfact ID: naming-server
- Artfact ID: api-gateway-server

Passo 2: Considere o seguinte esquema de endereçamento para as portas de cada serviço e instância.

- FrontEndMVC-tempo-server: server.port=9001 (1.ª instância), server.port=9002 (2.ª instância), etc

- naming-server: server.port=8761 (Porta por omissão)
- api-gateway-server: server.port=8755 (Porta por omissão)

Passo 3: Complete a implementação do projeto FrontEndMVC-tempo-server fazendo a ligação em falta à API do IPMA usando o Open Feign.

- Precisará de criar uma Classe Previsao e uma Classe ItemPrevisao para a API Jackson converter o Json recebido do IPMA para Java.
- Usando o Json retornado pelo IPMA, pode atalhar a criação das classes usando o conversor Json→Java em https://codebeautify.org/json-to-java-converter
- Poderá ser necessário o uso das anotações @JsonProperty("tMin") e @JsonProperty("tMax") antes destes atributos.

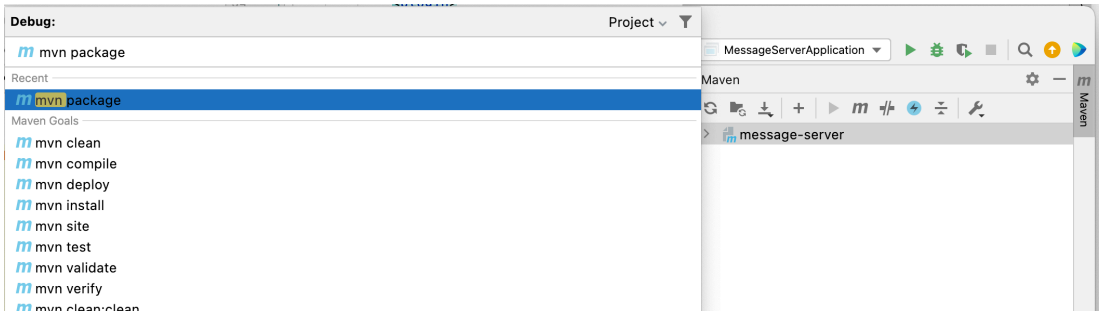
Passo 4: Assegure que o FrontEndMVC-tempo-server se regista no Spring Eureka. Adicione ainda as definições do Actuator

```
server.port=9001
spring.application.name=microservice-frontEnd-server
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
eureka.instance.instance-
id=${spring.application.name}:${random.int(100)}:${server.port}

## Configuring info endpoint for Atuator
info.app.name=Microservice FrontEnd MVC
info.app.description=Serviço Previsões Metereológicas a 5 dias
info.app.version=1.0.0
## Expose all actuator endpoints
management.endpoints.web.exposure.include=*
## Expose info Environment Variable
management.info.env.enabled = true
info.java-vendor = ${java.specification.vendor}
```

Passo 5: Execute e teste se todos os microserviços se executam devidamente, designadamente se ficam registados no servidor de nomes Eureka.

Passo 6: Pare todos os projetos e crie os seus ficheiros .Jar para distribuição, executando o comando maven: **man package**



Passo 7: Em cada Projeto crie um ficheiro **Dockerfile** com a estrutura seguinte (adapte a cada ao caso de cada projeto):

```
FROM openjdk:18
MAINTAINER Alexandre Fonte
COPY target/hello-aid-server-0.0.1-SNAPSHOT.jar hello-aid-server.jar
ENTRYPOINT ["java","-jar","hello-aid-server.jar"]
```

Ou

```
ENTRYPOINT exec java -jar hello-aid-server.jar
```

This file contains the following information:

- FROM: As the base for our image, we'll take the Java-enabled Alpine Linux created in the previous section.
- MAINTAINER: The maintainer of the image.
- COPY: We let Docker copy our jar file into the image.
- ENTRYPOINT: This will be the executable to start when the container is booting. We must define them as JSON-Array because we'll use an ENTRYPOINT in combination with a CMD for some application arguments.

Passo 8: Abra a janela terminal no IntelliJ IDEA e crie a image docker de cada projeto dando-lhes um nome e TAG adequados.

```
docker build -t hello-aid-server:latest .
```

Passo 9: No terminal de cada projecto, execute as images docker como daemons/contendores docker usando o comando docker run:

```
docker run -p 9001:9001 hello-aid-server
```

Passo 10: Deve evitar o *hard-coding* das portas e do endereço do serviço Eureka nos microserviços:

```
server.port=9001
```

```
server.port=${porta:9001}

eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
```

```
eureka.client.serviceUrl.defaultZone=${EUREKA_SERVER_URL:http://loca
lhost:8761/eureka}
```

Em cada Projeto adapte o ENTRYPOINT no Dockerfile com a seguinte estrutura genérica por forma a permitir a passagem de variáveis de ambiente JAVA_OPTS.

```
ENTRYPOINT exec java -jar frontEndMVC-server.jar
```

```
ENTRYPOINT exec java $JAVA_OPTS -jar frontEndMVC-server.jar
```

Passo 11: Abra a janela terminal no IntelliJ IDEA e crie novamente as images docket dos projeto dando-lhes um nome e TAG adequados.

Exemplo:

```
docker build -t frontend-mvc-server:latest .
```

Nota: O docker exige que os nomes das imagens docker sejam sempre em letra minúscula.

Passo 12: Execute o comando docker run para executar o contentor do microserviço **frontend-mvc-server**.

- Registe previamente o IP atribuído à sua máquina.
- Por que razão não serve o endereço **localhost**?

```
docker run -p9001:9001 -e JAVA_OPTS="-
DEUREKA_SERVER_URL=http://10.6.33.116:8761/eureka -Dporta=9001"
frontendmvc-server
```

Nota: como estamos a passar duas ***properties*** Java, precisamos de envolver por "".

- Crie uma segunda instância que usa a porta 9002 do sistema operativo hospedeiro. Mantenha a porta 9001 no contentor.

Seguir para a próxima página ....

Parte III: Playing with Docker Compose

Dockerize os servidores da nossa aplicação (Eureka, API Gateway e Microserviço) e realize o *deployment* da aplicação usando o Docker compose.

Considere os seguintes requisitos:

- Defina uma subrede IP interna para interligar os contentores da aplicação, com o seguinte endereço:
  - Nome da subrede: rede-app-tempo
  - subnet = 172.16.1.0/24
  - gateway = 172.16.1.254

- Defina os seguintes IP estáticos para o serviço Eureka Naming e API Gateway:
  - 172.16.1.10 – **Eureka**
  - 172.16.1.100 – **API Gateway**

- O frontend-mvc usará IP dinâmico (embora no passo 19 terá momentaneamente IP estático 172.16.1.1 – **frontend-mvc**

- O arranque do microserviço **frontend-mvc-server** e do API Gateway dependem do **naming-server.**

Passo 13: Verifique a versão Docker Compose

```
alexandrefonte — -bash — 80×24

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
ADF-iMac:~ alexandrefonte$ docker-compose --version
Docker Compose version v2.3.3
ADF-iMac:~ alexandrefonte$
```

Passo 14: No projeto do microserviço frontend, crie um ficheiro **docker-compose.yaml** com a seguinte estrutura base:

```yaml
version: '3.0'
services:
  frontend-mvc:
    image: frontend-mvc-server:latest
    ports:
      - 9002:9001
    environment:
      JAVA_OPTS: "-
DEUREKA_SERVER_URL=http://192.168.1.108:8761/eureka -
Dporta=9001"
```

Passo 15: Execute a aplicação e inspecione a rede pelo Id.

```
docker compose up -d

docker network ls

docker inspect <id rede>
```

Nota: It is also created a virtual network by default.

Passo 16: Pare a aplicação

```
docker compose down
```

Passo 17: Crie a rede virtual rede-app-tempo. Adicione uma seção networks ao ficheiro docker-compose.yaml e depois associe a rede ao serviço **frontend-mvc**

```
networks:
  rede-app-tempo:
```

Passo 18: Atualize o ficheiro docker-compose.yaml, execute novamente e inspecione a rede.

- Adicione um seção *IP management* à configuração da rede

```
ipam:
      config:
        - subnet: 172.16.1.0/24
          gateway: 172.16.1.254
```

- Atribua o endereço estático ao contentor adicionando uma seção **ipv4_address**

```
networks:
  rede-app-tempo:
    ipv4_address: 172.16.1.1
```

Resumo da configuração atual:

```
version: '3.0'
services:
  frontend-mvc:
    image: frontend-mvc-server:latest
    ports:
      - 9002:9001
    networks:
```

```
      rede-app-tempo:
        ipv4_address: 172.16.1.1
    environment:
      JAVA_OPTS: "-
DEUREKA_SERVER_URL=http://192.168.1.108:8761/eureka -
Dporta=9001"
networks:
  rede-app-tempo:
    ipam:
      config:
        - subnet: 172.16.1.0/24
          gateway: 172.16.1.254
```

Passo 19: Atualize o ficheiro docker-compose.yaml, com as configurações do do serviço Eureka e API gateway execute novamente e inspecione a rede.

Passo 20: Execute 5 replicas do front-end. O Docker compose aparentemente não está a permitir usar gamas de portas definidas pelo utilizador nem a seção deploys replicas, precisa de colocar no serviço **`frontend-mvc`** simplesmente:

```
ports:

  - 9000
```

Execute o comando com o parâmetro scale:

```
docker compose up -d --scale frontend-mvc=5
```

Nota: Esta solução somente funciona bem se existir um API gateway.

FIM

ANEXO – ficheiro docker-compose.yaml completo

```yaml
version: '3.0'
services:
  frontend-mvc:
    image: frontend-mvc-server:latest
    ports:
     # - 9001:9001

      - 9001
    networks:
      rede-app-tempo:
     #   ipv4_address: 172.16.1.1
    environment:
      JAVA_OPTS: "-
DEUREKA_SERVER_URL=http://172.16.1.10:8761/eureka -
Dporta=9001"
    depends_on:
      - naming
      - api-gateway
  naming:
    image: naming-service:latest
    ports:
      - 8761:8761
    networks:
      rede-app-tempo:
        ipv4_address: 172.16.1.10
  api-gateway:
    image: api-gateway:latest
    ports:
      - 8755:8755
    networks:
      rede-app-tempo:
        ipv4_address: 172.16.1.100
    environment:
      JAVA_OPTS: "-
DEUREKA_SERVER_URL=http://172.16.1.10:8761/eureka"
    depends_on:
      - naming
networks:
  rede-app-tempo:
    ipam:
      config:
        - subnet: 172.16.1.0/24
          gateway: 172.16.1.254
```