

### **Comandos auxiliares:**

- `sudo docker ps` - listar os containers que estão rodando
- `sudo docker ps -a` - lista todos os containers

### **1) Entrar no EC2 pelo Amazon Web Service**

### **2) Criar imagem com jdk11 dentro dele:**

2.1) `sudo docker pull adoptopenjdk/openjdk11`

### **3) Criar container:**

- `d` = rodando em background
- `i` - interativo
- `p` = porta - por isso vem as portas dps dele [9010:9010 - isso é um exemplo, podia ser qualquer porta inutilizada]
- `cp` = copiar - usei para copiar o arquivo da area de trabalho e joguei dentro do container na pasta `jar/`
- `adoptopenjdk/openjdk11` - é uma imagem disponivel do jdk

3.1) `sudo docker run -i -p 9010:9010 --name <nomeDesejadoContainer> adoptopenjdk/openjdk11 "/bin/bash"`

3.1.3) `sudo exec -it <nomeContainer> /bin/bash` - para executar o container e abrir o bash dele e possibilitar a criação um diretório `jar/`

3.1.6) `exit` do container

3.2) `sudo docker cp <caminhoDoJar> <nomeContainer>:/<pastaQueDesejaColar("jar/")>`

3.3) `sudo exec -it <nomeContainer> java -jar /jar/Java-CLI.jar`

#### **4) Criar o script de execução do jar**

4.1) vim startProducer.sh

4.2) sudo docker start producer && sudo docker exec -it producer java -jar /jar/Java-CLI.jar

4.3) ESQ + :wq para salvar as alterações no vim e sair

#### **5) commitar nível hard o container criado, init a partir do run :**

5.1) docker commit <idContainer> <nome/version>

5.2) fazer a criação do container baseado na imagem do container commitado com a função "java -jar..." no command