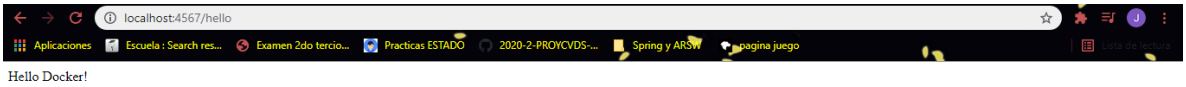
Se crea proyecto Maven y se usa spark , se corre desde consola:

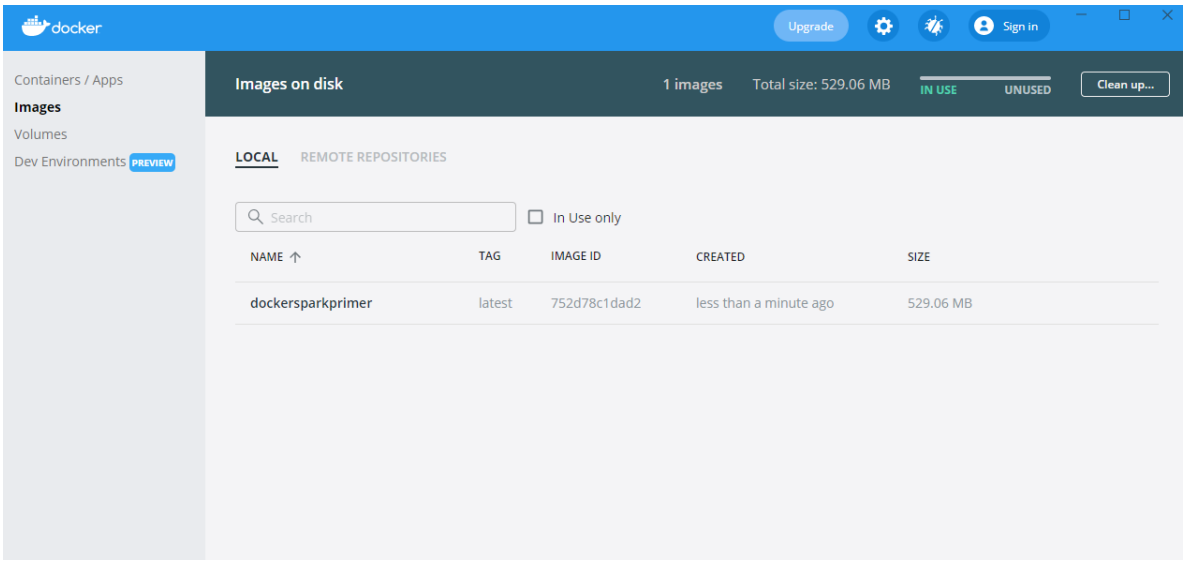
```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>java -cp "target/classes;target/dependency/*" co.edu.escuelaing.virtualiz
ation.SparkWebServer
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

Se abre el explorador y se coloca el localhost



Hello Docker!

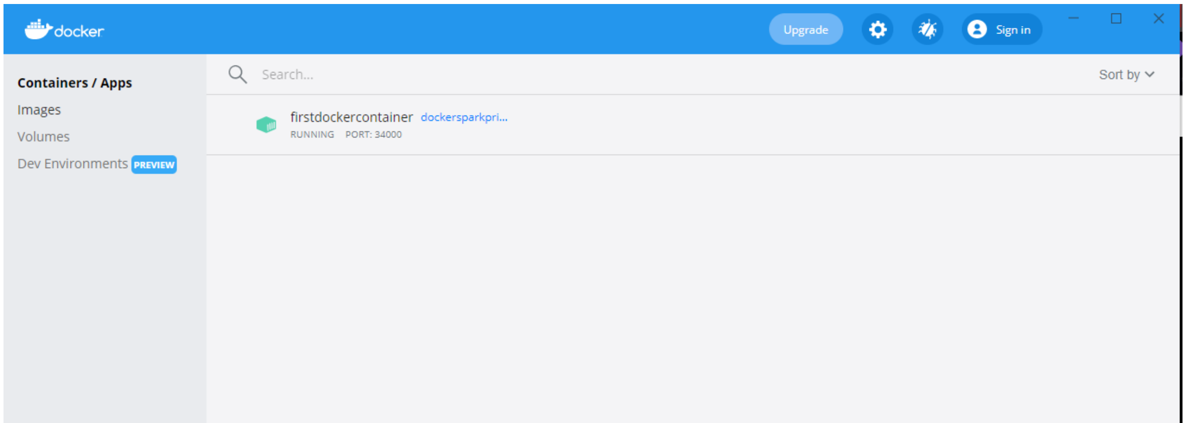
Usando la herramienta de línea de comandos de Docker construya la imagen:
docker build --tag dockersparkprimer .



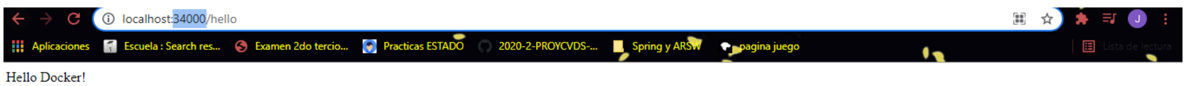
Se crea instancia desde la consola

```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>docker run -d -p 34000:6000 --name firstdockercontainer dockersparkprimer

0ceec53600b31987b4997c303be4a570d5bdb6e6f436c9da75d424a8e4866c06

C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>
```

Se verifica que el contenedor se creara



Se corre nuevamente el localhost pero esta vez en el puerto 34000



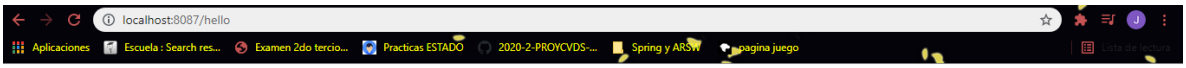
Se ejecuta el docker compose:

```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>docker-compose up -d
[+] Running 11/11
 - db Pulled                                                                                          24.6s
   - c4bb02b17bb4 Pull complete                                                                        7.3s
   - 3f58e3bb3be4 Pull complete                                                                        7.6s
   - a229fb575a6e Pull complete                                                                        8.2s
   - 8f5ddc533743 Pull complete                                                                        8.4s
   - 5e9d2af6e206 Pull complete                                                                        8.5s
   - 3b6c28c0235b Pull complete                                                                        8.6s
   - fd6b165aa317 Pull complete                                                                        8.7s
   - 772467f0b4cd Pull complete                                                                       18.4s
   - a94d919fbb86 Pull complete                                                                       18.6s
   - b0cad17917cd Pull complete                                                                       18.7s
[+] Building 1.9s (9/9) FINISHED
 => [internal] load build definition from Dockerfile                                                   0.1s
 => => transferring dockerfile: 284B                                                                   0.0s
 => [internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                        0.0s
 => [internal] load metadata for docker.io/library/openjdk:8                                           1.5s
 => [1/4] FROM docker.io/library/openjdk:8@sha256:51ab8555b0a8ee08813da2d1d4a71aa8901f4efba23b6c206a3a7d3e8a3a557  0.0s
 => [internal] load build context                                                                      0.1s
 => => transferring context: 3.10MB                                                                    0.1s
 => CACHED [2/4] WORKDIR /usrapp/bin                                                                   0.0s
 => CACHED [3/4] COPY /target/classes /usrapp/bin/classes                                              0.0s
 => CACHED [4/4] COPY /target/dependency /usrapp/bin/dependency                                        0.0s
 => exporting to image                                                                                 0.0s
 => => exporting layers                                                                                0.0s
 => => writing image sha256:9269e4c8d5f012dc19e1667e30ddb18f524d6855e3f72681ec34c6b9bef667a1           0.0s
```

Con el comando docker ps se puede verificar los servicios, se encuentra que se crearon dos nuevos. Uno es web y otro es el de mongo db

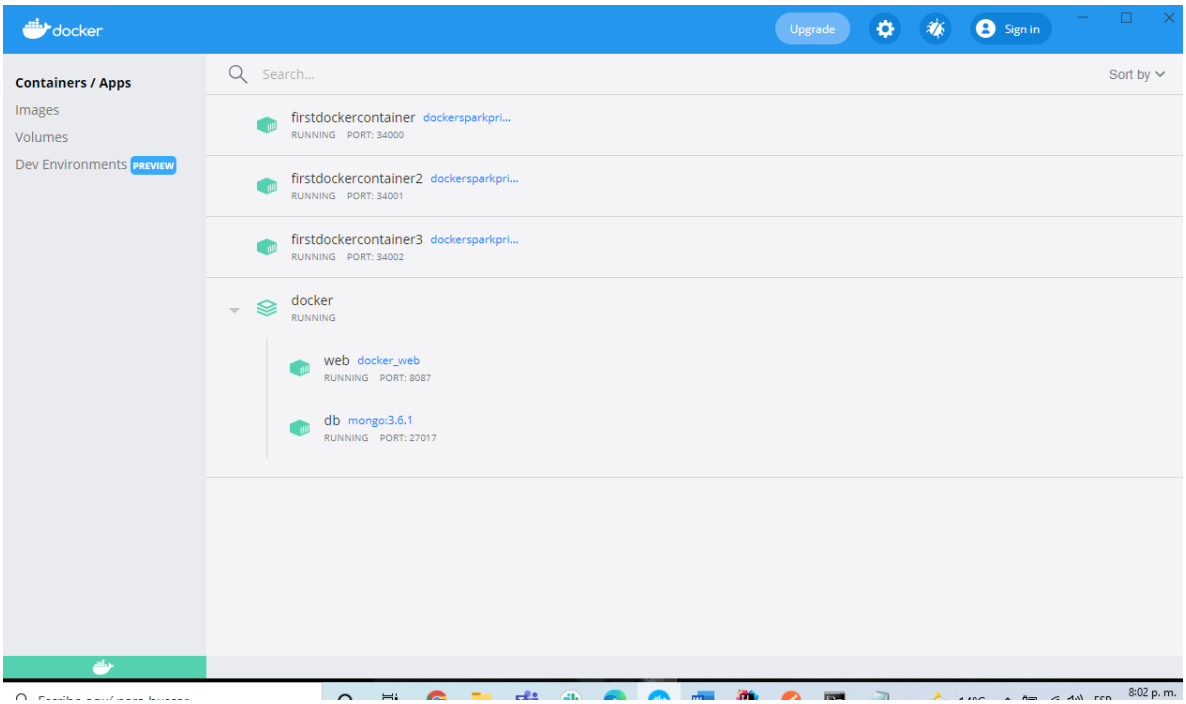
```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED             STATUS             PORTS
                  NAMES
8fe2fa9f065d   docker_web         "java -cp ./classes:…"   6 minutes ago       Up 6 minutes       0.0.0.0:8087->6000/tc
p, :::8087->6000/tcp              web
434b5a5744e2   mongo:3.6.1        "docker-entrypoint.s…"   6 minutes ago       Up 6 minutes       0.0.0.0:27017->27017/
tcp, :::27017->27017/tcp          db
31f530c24d38   dockersparkprimer  "java -cp ./classes:…"   13 minutes ago      Up 13 minutes      0.0.0.0:34002->6000/t
cp, :::34002->6000/tcp            firstdockercontainer3
136e1546839b   dockersparkprimer  "java -cp ./classes:…"   13 minutes ago      Up 13 minutes      0.0.0.0:34001->6000/t
cp, :::34001->6000/tcp            firstdockercontainer2
312b951a1240   dockersparkprimer  "java -cp ./classes:…"   About an hour ago   Up About an hour   0.0.0.0:34000->6000/t
cp, :::34000->6000/tcp            firstdockercontainer

C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>
```

A demás se verifica que se corra el nuevo servicio web en el puerto 8087

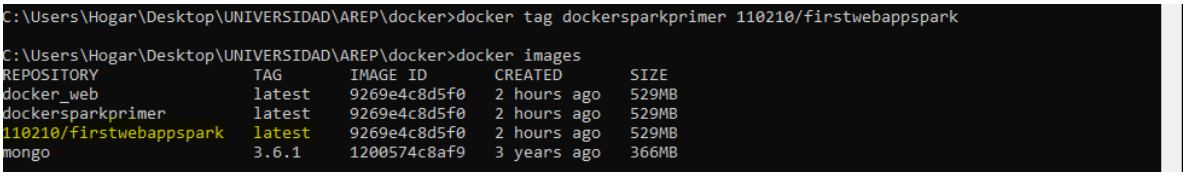


Hello Docker!

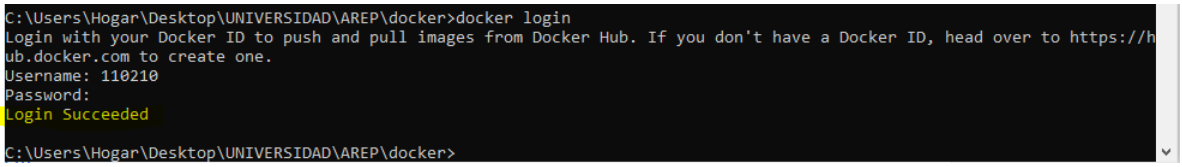
En Docker Desktop verificamos que se encuentre el nuevo servicio



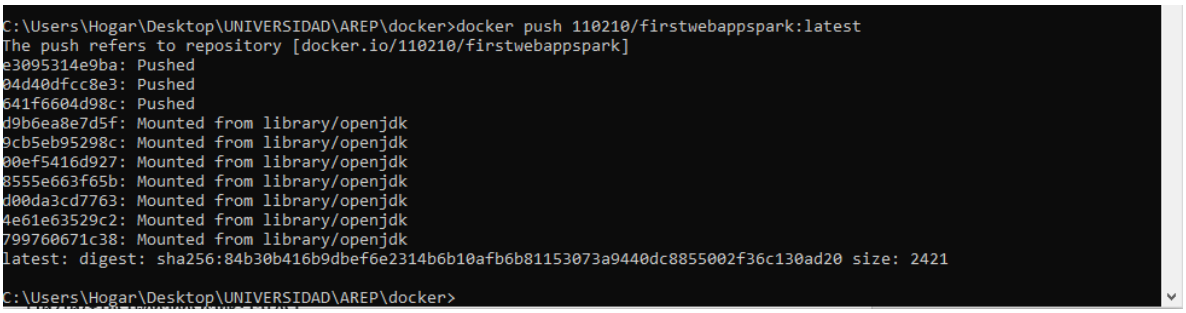
Se crea una referencia a su imagen con el nombre del repositorio a donde desea subirla con el comando **docker tag dockersparkprimer 110210/firstwebappspark** . Al momento de crearlo se puede ver con el comando **docker images** que se creo el servicio con el TAG latest



Luego procedemos a loguearnos con el comando **docker login**

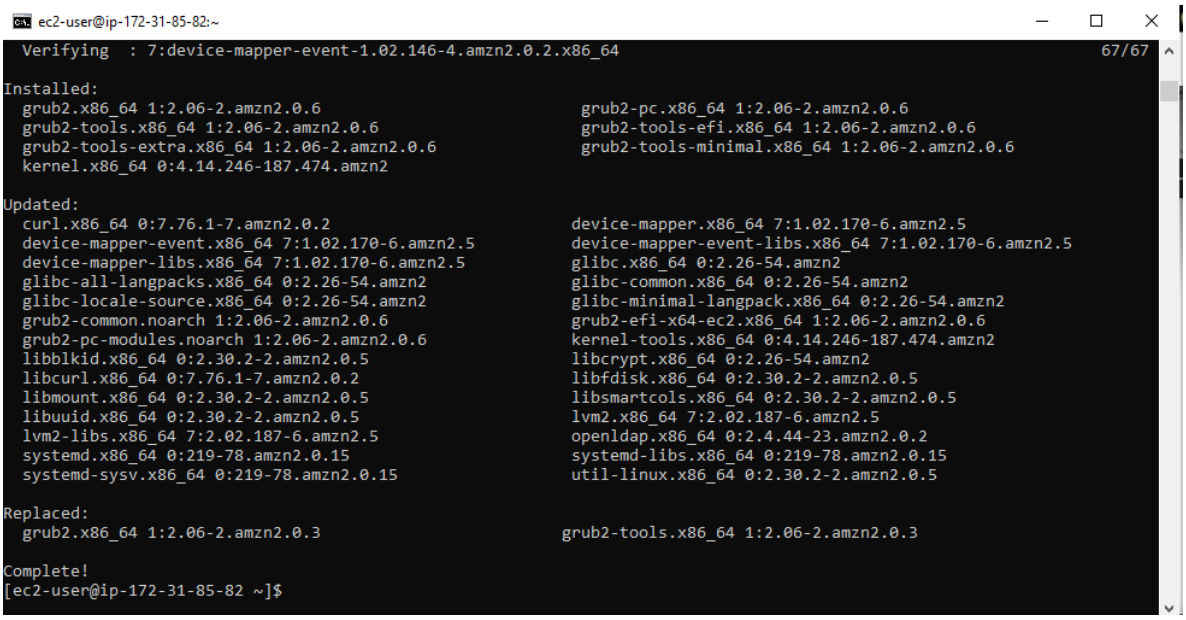


Empujamos la imagen al repositorio en DockerHub con el comando **docker push 110210/firstwebappspark:latest**

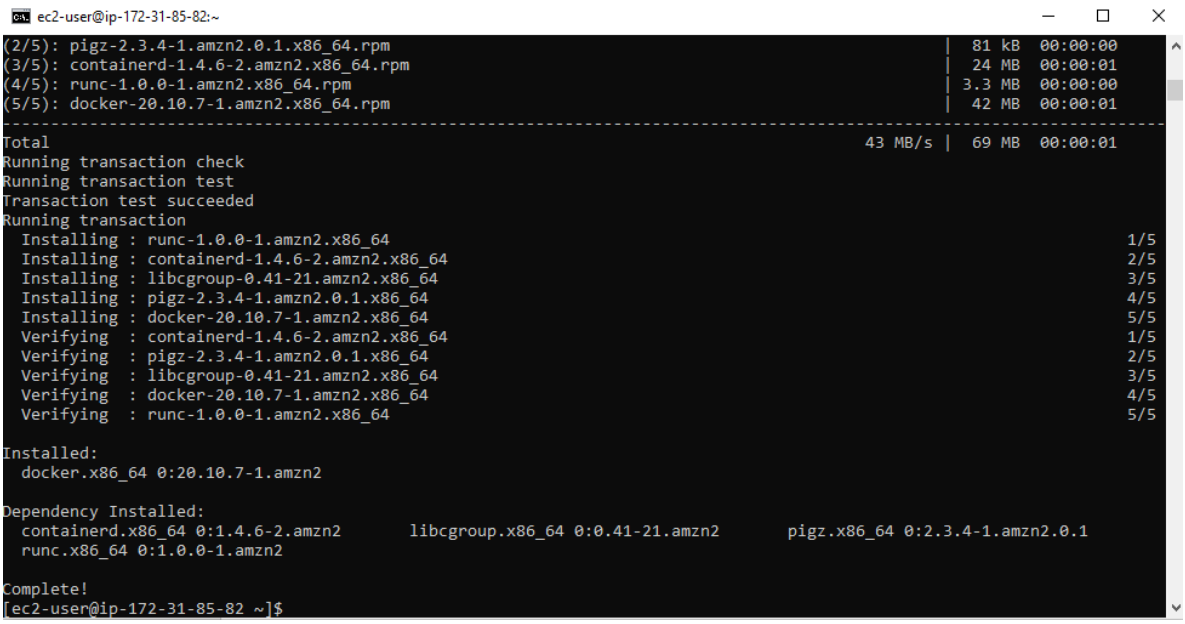


Cuarta parte

Se actualiza el sistema operativo con el comando **sudo yum update -y**

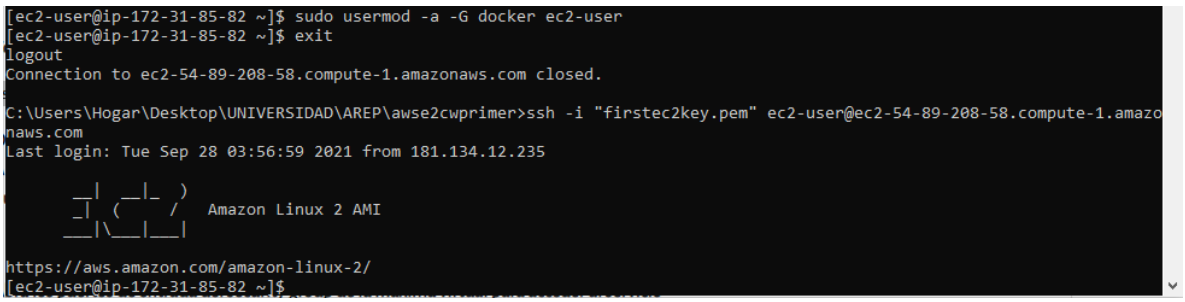


Se instala docker con el comando **sudo yum install docker**

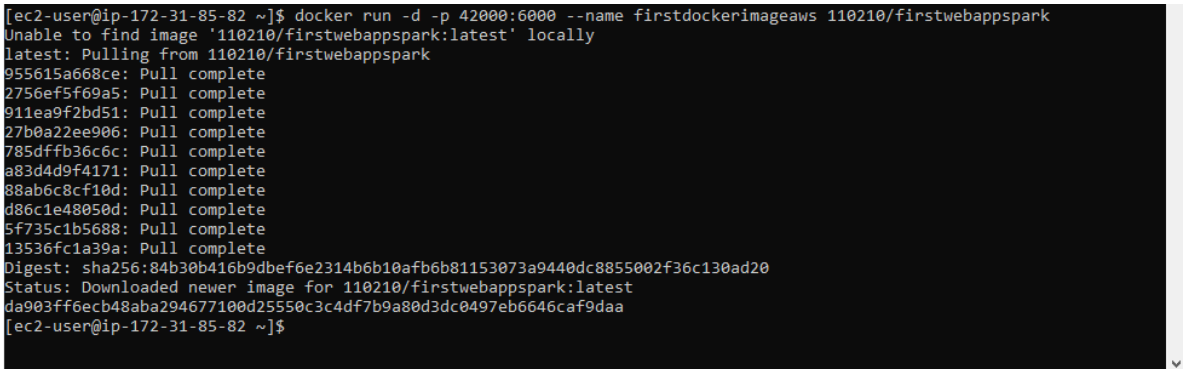


Se inicia el servicio de Docker con el comando **sudo service docker start**

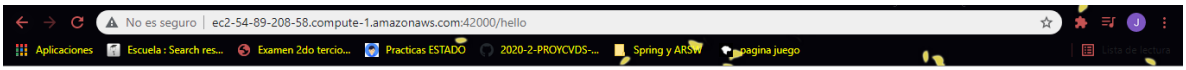
Se configura su usuario en el grupo de docker para no tener que ingresar "sudo" cada vez que invoca un comando con el comando **sudo usermod -a -G docker ec2-user**



A partir de la imagen creada en Dockerhub cree una instancia de un contenedor docker independiente de la consola (opción "-d") y con el puerto 6000 enlazado a un puerto físico de su máquina (opción -p) con el comando **docker run -d -p 42000:6000 --name firstdockerimageaws 110210/firstwebappspark**



Verificamos que pueda acceder en una url similar a esta (la url específica depende de los valores de su maquina virtual EC2)



Hello Docker!