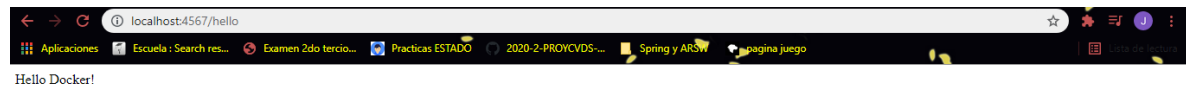


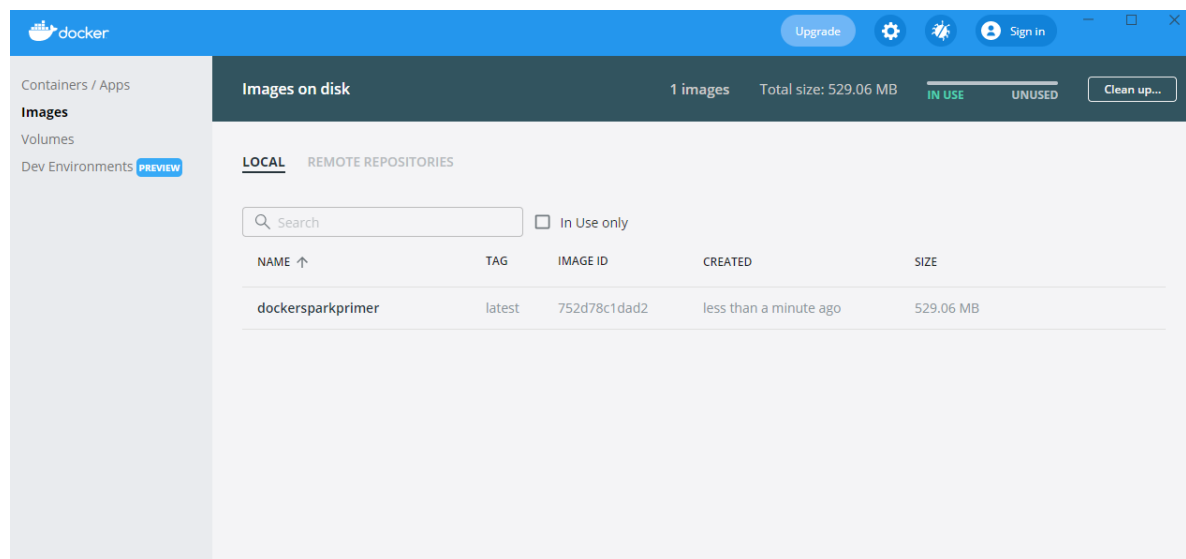
Se crea proyecto Maven y se usa spark , se corre desde consola:

```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>java -cp "target/classes;target/dependency/*" co.edu.escuelaing.virtualization.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



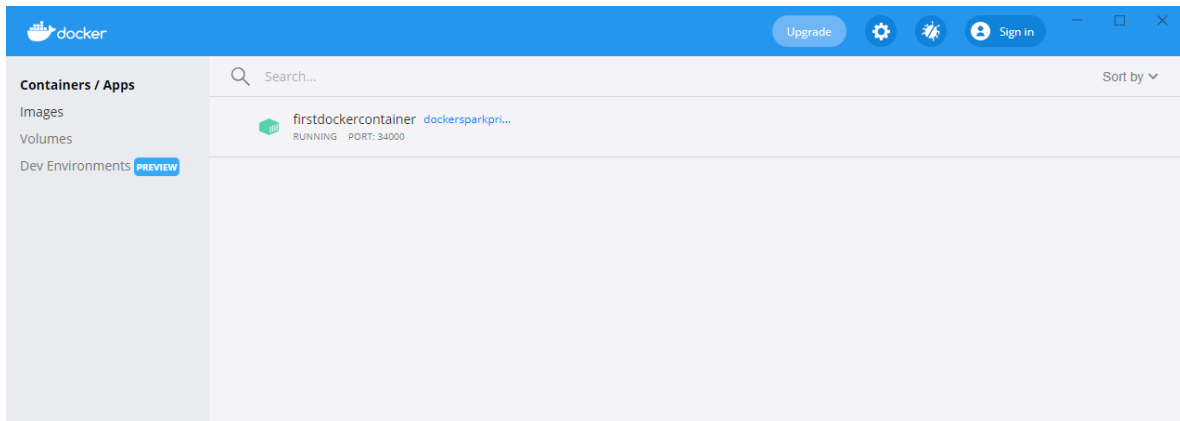
Usando la herramienta de línea de comandos de Docker construya la imagen:
`docker build --tag dockersparkprimer .`



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
```

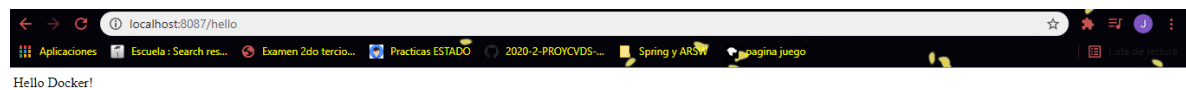
Con el comando docker ps se puede verificar los servicios, se encuentra que se crearon dos nuevos. Uno es web y otro es el de mongo db

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

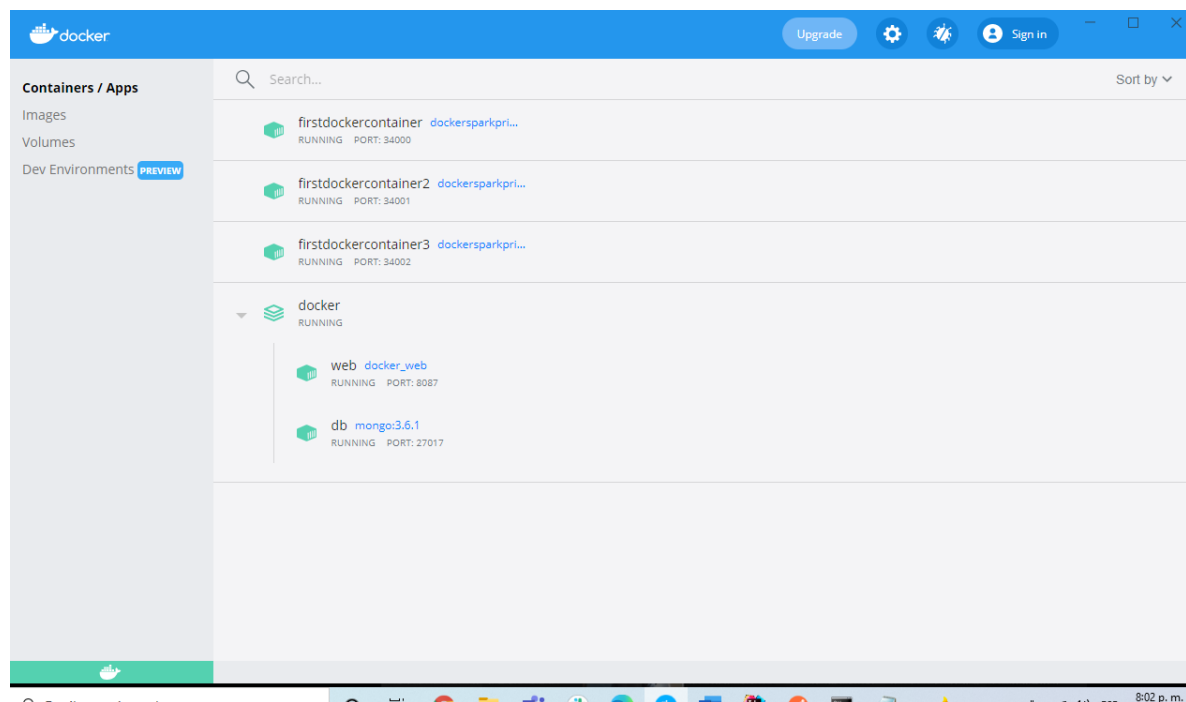
CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
8fe2fa9f065d	docker_web	web	"java -cp ./classes:..."	6 minutes ago	Up 6 minutes	0.0.0.0:8087->6000/tcp
434b5a5744e2	mongo:3.6.1	db	"docker-entrypoint.s..."	6 minutes ago	Up 6 minutes	0.0.0.0:27017->27017/tcp
31f530c24d38	dockersparkspring	firstdockercontainer3	"java -cp ./classes:..."	13 minutes ago	Up 13 minutes	0.0.0.0:34002->6000/tcp
136e1546839b	dockersparkspring	firstdockercontainer2	"java -cp ./classes:..."	13 minutes ago	Up 13 minutes	0.0.0.0:34001->6000/tcp
312b951a1240	dockersparkspring	firstdockercontainer	"java -cp ./classes:..."	About an hour ago	Up About an hour	0.0.0.0:34000->6000/tcp

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

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



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

```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>docker tag dockersparkprimer 110210/firstwebappspark

C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
docker_web          latest      9269e4c8d5f0  2 hours ago  529MB
dockersparkprimer   latest      9269e4c8d5f0  2 hours ago  529MB
110210/firstwebappspark latest      9269e4c8d5f0  2 hours ago  529MB
mongo               3.6.1      1200574c8af9  3 years ago  366MB
```

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

```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: 110210
Password:
Login Succeeded

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

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

```
C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\docker>docker push 110210/firstwebappspark:latest
The push refers to repository [docker.io/110210/firstwebappspark]
e3095314e9ba: Pushed
04d40dfcc8e3: Pushed
641f6604d98c: Pushed
d9b6ea8e7d5f: Mounted from library/openjdk
9cb5eb95298c: Mounted from library/openjdk
00ef5416d927: Mounted from library/openjdk
8555e663f65b: Mounted from library/openjdk
d00da3cd7763: Mounted from library/openjdk
4e61e63529c2: Mounted from library/openjdk
799760671c38: Mounted from library/openjdk
latest: digest: sha256:84b30b416b9dbef6e2314b6b10afb6b81153073a9440dc8855002f36c130ad20 size: 2421

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

Cuarta parte

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

```
ec2-user@ip-172-31-85-82:~  
Verifying : 7:device-mapper-event-1.02.146-4.amzn2.0.2.x86_64 67/67  
Installed:  
grub2.x86_64 1:2.06-2.amzn2.0.6  
grub2-tools.x86_64 1:2.06-2.amzn2.0.6  
grub2-tools-extra.x86_64 1:2.06-2.amzn2.0.6  
kernel.x86_64 0:4.14.246-187.474.amzn2  
grub2-pc.x86_64 1:2.06-2.amzn2.0.6  
grub2-tools-efi.x86_64 1:2.06-2.amzn2.0.6  
grub2-tools-minimal.x86_64 1:2.06-2.amzn2.0.6  
Updated:  
curl.x86_64 0:7.76.1-7.amzn2.0.2  
device-mapper-event.x86_64 7:1.02.170-6.amzn2.5  
device-mapper-event-libs.x86_64 7:1.02.170-6.amzn2.5  
device-mapper-libs.x86_64 7:1.02.170-6.amzn2.5  
glibc-all-langpacks.x86_64 0:2.26-54.amzn2  
glibc-locale-source.x86_64 0:2.26-54.amzn2  
grub2-common.noarch 1:2.06-2.amzn2.0.6  
grub2-pc-modules.noarch 1:2.06-2.amzn2.0.6  
libblkid.x86_64 0:2.30.2-2.amzn2.0.5  
libcurl.x86_64 0:7.76.1-7.amzn2.0.2  
libmount.x86_64 0:2.30.2-2.amzn2.0.5  
libuuid.x86_64 0:2.30.2-2.amzn2.0.5  
lvm2-libs.x86_64 7:2.02.187-6.amzn2.5  
systemd.x86_64 0:219-78.amzn2.0.15  
systemd-sysv.x86_64 0:219-78.amzn2.0.15  
device-mapper.x86_64 7:1.02.170-6.amzn2.5  
device-mapper-event-libs.x86_64 7:1.02.170-6.amzn2.5  
glibc.x86_64 0:2.26-54.amzn2  
glibc-common.x86_64 0:2.26-54.amzn2  
glibc-minimal-langpack.x86_64 0:2.26-54.amzn2  
grub2-efi-x64-ec2.x86_64 1:2.06-2.amzn2.0.6  
kernel-tools.x86_64 0:4.14.246-187.474.amzn2  
libcrypt.x86_64 0:2.26-54.amzn2  
libfdisk.x86_64 0:2.30.2-2.amzn2.0.5  
libsmartcols.x86_64 0:2.30.2-2.amzn2.0.5  
lvm2.x86_64 7:2.02.187-6.amzn2.5  
openldap.x86_64 0:2.4.44-23.amzn2.0.2  
systemd-libs.x86_64 0:219-78.amzn2.0.15  
util-linux.x86_64 0:2.30.2-2.amzn2.0.5  
Replaced:  
grub2.x86_64 1:2.06-2.amzn2.0.3  
grub2-tools.x86_64 1:2.06-2.amzn2.0.3  
Complete!  
[ec2-user@ip-172-31-85-82 ~]$
```

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

```
ec2-user@ip-172-31-85-82:~  
(2/5): pigz-2.3.4-1.amzn2.0.1.x86_64.rpm | 81 kB 00:00:00  
(3/5): containerd-1.4.6-2.amzn2.x86_64.rpm | 24 MB 00:00:01  
(4/5): runc-1.0.0-1.amzn2.x86_64.rpm | 3.3 MB 00:00:00  
(5/5): docker-20.10.7-1.amzn2.x86_64.rpm | 42 MB 00:00:01  
-----  
Total | 43 MB/s | 69 MB 00:00:01  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
Installing : runc-1.0.0-1.amzn2.x86_64 1/5  
Installing : containerd-1.4.6-2.amzn2.x86_64 2/5  
Installing : libcgrouper-0.41-21.amzn2.x86_64 3/5  
Installing : pigz-2.3.4-1.amzn2.0.1.x86_64 4/5  
Installing : docker-20.10.7-1.amzn2.x86_64 5/5  
Verifying : containerd-1.4.6-2.amzn2.x86_64 1/5  
Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64 2/5  
Verifying : libcgrouper-0.41-21.amzn2.x86_64 3/5  
Verifying : docker-20.10.7-1.amzn2.x86_64 4/5  
Verifying : runc-1.0.0-1.amzn2.x86_64 5/5  
Installed:  
docker.x86_64 0:20.10.7-1.amzn2  
Dependency Installed:  
containerd.x86_64 0:1.4.6-2.amzn2 libcgrouper.x86_64 0:0.41-21.amzn2 pigz.x86_64 0:2.3.4-1.amzn2.0.1  
runc.x86_64 0:1.0.0-1.amzn2  
Complete!  
[ec2-user@ip-172-31-85-82 ~]$
```

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

```
Complete!
[ec2-user@ip-172-31-85-82 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-85-82 ~]$
```

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**

```
[ec2-user@ip-172-31-85-82 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-85-82 ~]$ exit
logout
Connection to ec2-54-89-208-58.compute-1.amazonaws.com closed.

C:\Users\Hogar\Desktop\UNIVERSIDAD\AREP\awse2cwprimer>ssh -i "firstec2key.pem" ec2-user@ec2-54-89-208-58.compute-1.amazonaws.com
Last login: Tue Sep 28 03:56:59 2021 from 181.134.12.235

  _ | _ | _ |
  _ | ( _ | _ |
  _ | \ _ | _ |

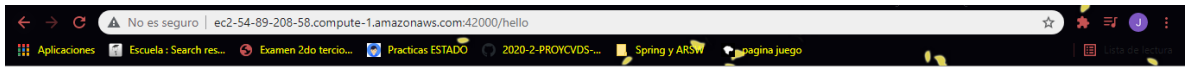
Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-85-82 ~]$
```

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**

```
[ec2-user@ip-172-31-85-82 ~]$ docker run -d -p 42000:6000 --name firstdockerimageaws 110210/firstwebappspark
Unable to find image '110210/firstwebappspark:latest' locally
latest: Pulling from 110210/firstwebappspark
955615a668ce: Pull complete
2756ef5f69a5: Pull complete
911ea9f2bd51: Pull complete
27b0a22ee906: Pull complete
785dfffb36c6c: Pull complete
a83d4d9f4171: Pull complete
88ab6c8cf10d: Pull complete
d86c1e48050d: Pull complete
5f735c1b5688: Pull complete
13536fc1a39a: Pull complete
Digest: sha256:84b30b416b9dbef6e2314b6b10afb6b81153073a9440dc8855002f36c130ad20
Status: Downloaded newer image for 110210/firstwebappspark:latest
da903ff6ecb48aba294677100d25550c3c4df7b9a80d3dc0497eb6646caf9daa
[ec2-user@ip-172-31-85-82 ~]$
```

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



Hello Docker!