DOCKER
PROJECT HANDSON

HandsOn video-
https://www.youtube.com/watch?v=5FfpNuk0sBg&t=4s

Step1 - Take the EC2 Ubuntu instance from AWS with below configurations

Type - Ubuntu
Size - T2.Micro
GB - 8

Step 2 Install tools like

Maven -

sudo apt update -y
sudo apt install maven -y
mvn -version

Docker -

sudo apt update -y

sudo apt install apt-transport-https ca-certificates curl software-properties-common -y

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable" -y

sudo apt update -y

apt-cache policy docker-ce -y

sudo apt install docker-ce -y

#sudo systemctl status docker

sudo chmod 777 /var/run/docker.sock

Git

apt install git

Step 3 - Clone the code

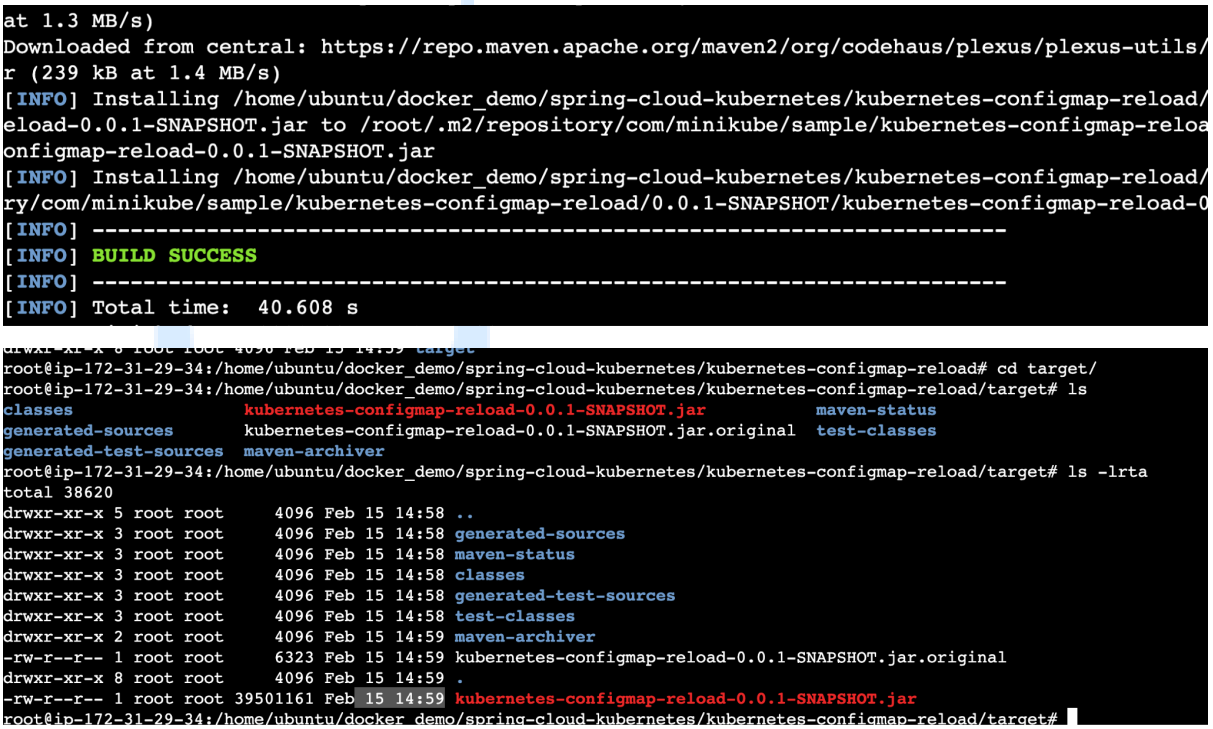
git clone https://github.com/DEVOPS-WITH-WEB-DEV/spring-cloud-kubernetes.git

Step 4 - Go over the docker file in above repo

https://github.com/DEVOPS-WITH-WEB-DEV/spring-cloud-kubernetes/blob/main/kubernetes-configmap-reload/Dockerfile

Step 5 - Build the code with maven

mvn clean install -DskipTests



Step 6 - Go to project folder in Ubuntu where dockerfile is present

Step 7 - Run the below command to create the docker image

docker build -t bahubali/1:latest .

```
root@ip-172-31-29-34:/home/ubuntu/docker_demo/spring-cloud-kubernetes/kubernetes-configmap-reload# docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
root@ip-172-31-29-34:/home/ubuntu/docker_demo/spring-cloud-kubernetes/kubernetes-configmap-reload# docker build -t bahubali
.
[+] Building 5.9s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.1s
 => => transferring dockerfile: 149B                                                                                0.0s
 => [internal] load .dockerignore                                                                                   0.1s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/library/openjdk:8-jdk-alpine                                             1.1s
 => [internal] load build context                                                                                   1.4s
 => => transferring context: 39.51MB                                                                                1.4s
 => [1/2] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283  3.9s
 => => resolve docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283  0.1s
 => => sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 70.73MB / 70.73MB                    1.9s
 => => sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3 1.64kB / 1.64kB                      0.0s
 => => sha256:44b3cea369c947527e266275cee85c71a81f20fc5076f6ebb5a13f19015dce71 947B / 947B                          0.0s
 => => sha256:a3562aa0b991a80cfe8172847c8be6dbf6e46340b759c2b782f8b8be45342717 3.40kB / 3.40kB                      0.0s
 => => sha256:e7c96db7181be991f19a9fb6975cdbbd73c65f4a2681348e63a141a2192a5f10 2.76MB / 2.76MB                      0.2s
 => => sha256:f910a506b6cb1dbec766725d70356f695ae2bf2bea6224dbe8c7c6ad4f3664a2 238B / 238B                          0.2s
 => => extracting sha256:e7c96db7181be991f19a9fb6975cdbbd73c65f4a2681348e63a141a2192a5f10                           0.8s
```

```
 => => naming to docker.io/bahubali/2
root@ip-172-31-29-34:/home/ubuntu/docker_demo/spring-cloud-kubernetes/kubernetes-configmap-reload# docker images
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
bahubali/1   latest    b142872d6397   About a minute ago   144MB
bahubali/2   latest    b142872d6397   About a minute ago   144MB
bahubali     latest    b142872d6397   About a minute ago   144MB
root@ip-172-31-29-34:/home/ubuntu/docker_demo/spring-cloud-kubernetes/kubernetes-configmap-reload#
```

Step 7 - Run the below command to create the image into container

docker run -itd <imageID> –name chirutha

Here chirutha is container name
ImageId is from above screenshot

```
3
07c1cde6de00   b142872d6397   "/bin/sh -c 'exec ja…"   2 minutes ago    Exited (130) About a minute ago            chirutha
2
b88da0ecd040   b142872d6397   "/bin/sh -c 'exec ja…"   3 minutes ago    Exited (130) 2 minutes ago                 chirutha
root@ip-172-31-29-34:/home/ubuntu/docker_demo/spring-cloud-kubernetes/kubernetes-configmap-reload# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                              PORTS     NAMES
6afb57fc2f8f   b142872d6397   "/bin/sh -c 'exec ja…"   12 seconds ago   Up 11 seconds                               chirutha
3
07c1cde6de00   b142872d6397   "/bin/sh -c 'exec ja…"   2 minutes ago    Exited (130) About a minute ago            chirutha
2
b88da0ecd040   b142872d6397   "/bin/sh -c 'exec ja…"   4 minutes ago    Exited (130) 2 minutes ago                 chirutha
root@ip-172-31-29-34:/home/ubuntu/docker_demo/spring-cloud-kubernetes/kubernetes-configmap-reload#
```

Step 8 - Go inside the container and check the data

docker exec -it <containerID> /bin/sh

Step 9 - Check the app.jar file in docker

```
dr-xr-xr-x  176 root     root             0 Feb 15 15:23 proc
drwxr-xr-x    1 root     root          4096 Feb 15 15:23 etc
-rwxr-xr-x    1 root     root             0 Feb 15 15:23 .dockerenv
drwxr-xr-x    1 root     root          4096 Feb 15 15:23 ..
drwxr-xr-x    1 root     root          4096 Feb 15 15:23 .
drwxr-xr-x    5 root     root           360 Feb 15 15:23 dev
drwxrwxrwt    1 root     root          4096 Feb 15 15:23 tmp
drwx------    1 root     root          4096 Feb 15 15:25 root
/ # ps ef | grep -i app.jar
    1 root      0:11 java -jar app.jar --info
   48 root      0:00 grep -i app.jar
/ # exit
```