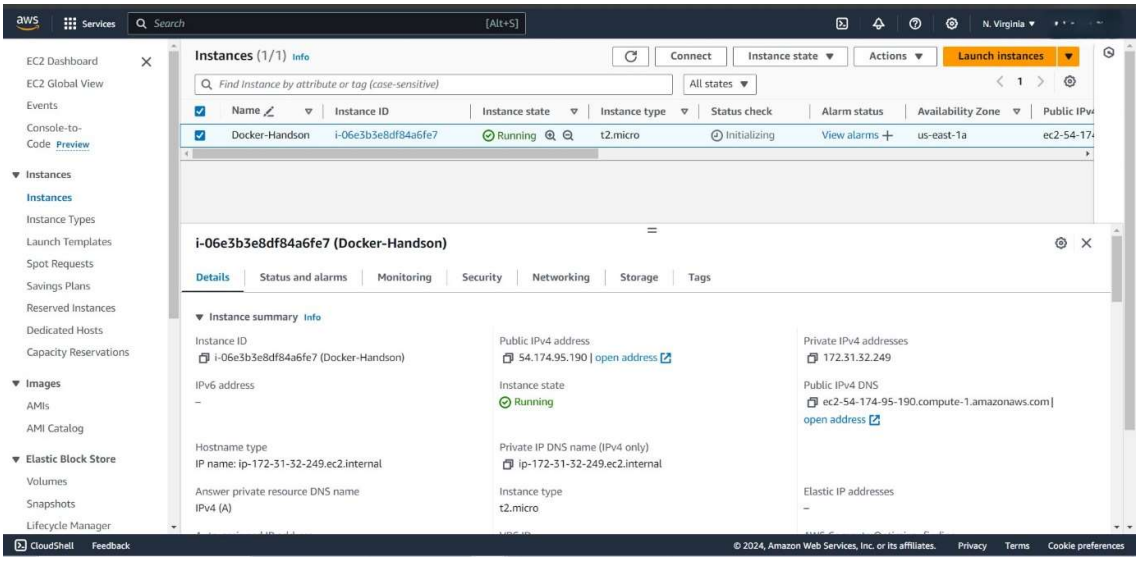
DOCKER PROJECT HANDSON

Step 1: Launch an EC2 Ubuntu Instance

1. **Log in to your AWS Management Console** and navigate to the EC2 dashboard.
2. **Launch a new instance**:
    - Choose an Amazon Machine Image (AMI): Select Ubuntu Server.
    - Choose an Instance Type: Select t2.micro.
    - Configure Instance: Set the storage to 8 GB.
    - Add tags and configure security group as per your requirements.



3. **Connect to your instance** using SSH:

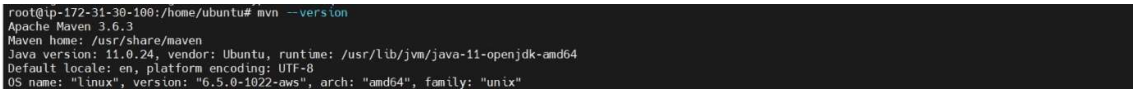
Step 2: Install Necessary Tools

Update Package List

```
sudo apt update -y
```

Install Maven

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



Install Docker

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
sudo apt install docker-ce -y
sudo chmod 777 /var/run/docker.sock

Install Git

sudo apt install git -y

mkdir docker_handson

cd docker handson/

```
root@ip-172-31-30-100:/home/ubuntu# sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.11).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 23 not upgraded.
root@ip-172-31-30-100:/home/ubuntu# mkdir docker_handson
root@ip-172-31-30-100:/home/ubuntu# cd docker_handson
root@ip-172-31-30-100:/home/ubuntu/docker_handson# cd ..
root@ip-172-31-30-100:/home/ubuntu# cd docker_handson/
```

Step 3: Clone the Repository

git clone https://github.com/bhargavdevopsaws/spring-cloud-kubernetes.git

ls

cd spring-cloud-kubernetes/

ls

cd Kubernetes-configmap-reload/

```
root@ip-172-31-30-100:/home/ubuntu/docker_handson# git clone https://github.com/bhargavdevopsaws
/spring-cloud-kubernetes.git
Cloning into 'spring-cloud-kubernetes'...
remote: Enumerating objects: 206, done.
remote: Counting objects: 100% (72/72), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 206 (delta 58), reused 47 (delta 43), pack-reused 134
Receiving objects: 100% (206/206), 99.10 KiB | 6.61 MiB/s, done.
Resolving deltas: 100% (82/82), done.
root@ip-172-31-30-100:/home/ubuntu/docker_handson# ls
spring-cloud-kubernetes
root@ip-172-31-30-100:/home/ubuntu/docker_handson# ^C
root@ip-172-31-30-100:/home/ubuntu/docker_handson# cd spring-cloud-kubernetes/
root@ip-172-31-30-100:/home/ubuntu/docker_handson/spring-cloud-kubernetes# ls
kubernetes-configmap-reload
```

Step 4: Review the Dockerfile

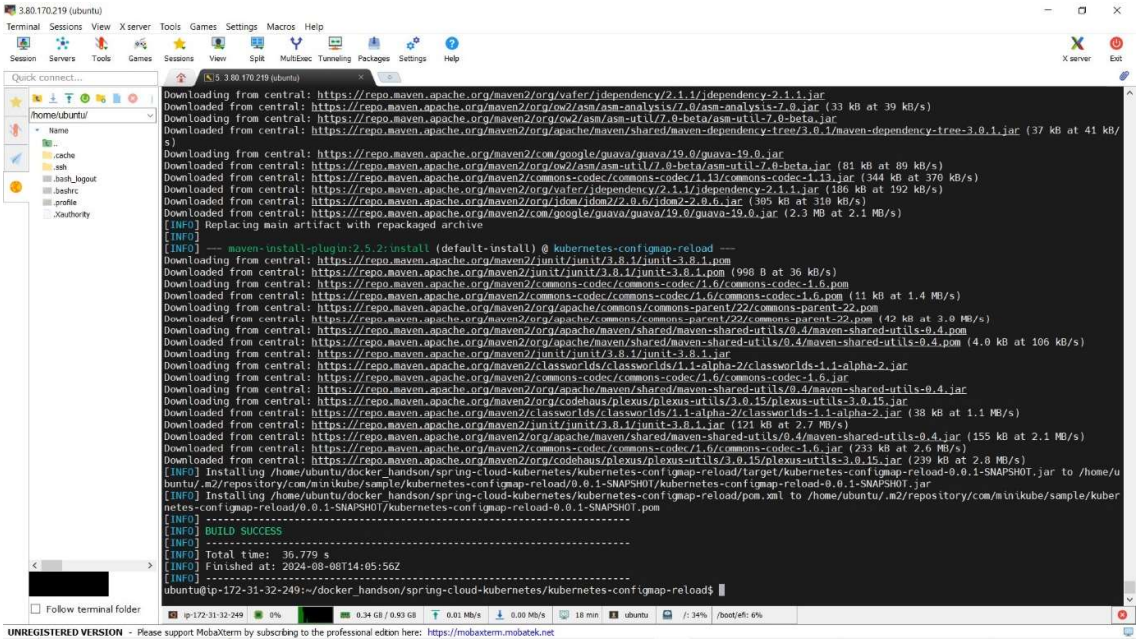
vim Dockerfile

After executing this command review the below script and save and exit from the vim editor. (Save & Exit (:wq))

```
FROM openjdk:8-jdk-alpine
VOLUME /c/Users/eresh.gorant la/
COPY ./target/*.jar app.jar
ENV JAVA_OPTS=""
ENTRYPOINT exec java -jar app.jar –info
```

Step 5: Build the Code with Maven

mvn clean install -DskipTests

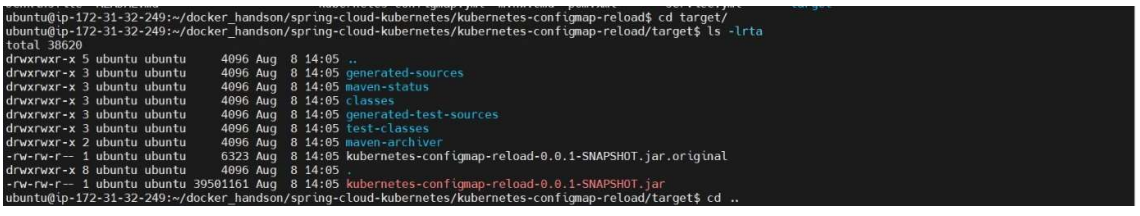


ls

cd target/

ls -lrta

cd ..



Step 6: Build the Docker Image

docker build -t alpha .

```
ubuntu@ip-172-31-32-249:~/docker_handson/spring-cloud-kubernetes/kubernetes-configmap-reload$ docker build -t alpha .
[+] Building 7.4s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.1s
 => => transferring dockerfile: 184B                                                                                0.0s
 => [internal] load .dockerignore                                                                                   0.1s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/library/openjdk:8-jdk-alpine                                             0.6s
 => [internal] load build context                                                                                   2.5s
 => => transferring context: 39.51MB                                                                                2.1s
 => [1/2] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2  4.7s
 => => resolve docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f22  0.1s
 => => sha256:44b3cea369c947527e266275cee85c71a81f20fc5076f6ebb5a13f19015dce71 947B / 947B                          0.0s
 => => sha256:a3562aa0b991a80cfe8172847c8be6dbf6e46340b759c2b782f8b8be45342717 3.40kB / 3.40kB                      0.0s
 => => sha256:e7c96db7181be991f19a9fb6975cdbbd73c65f4a2681348e63a141a2192a5f10 2.76MB / 2.76MB                      0.3s
 => => sha256:f910a506b6cb1dbec766725d70356f695ae2bf2bea6224dbe8c7c6ad4f3664a2 238B / 238B                          0.1s
 => => sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397 70.73MB / 70.73MB                    2.5s
 => => sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2283343547b3 1.64kB / 1.64kB                      0.0s
 => => extracting sha256:e7c96db7181be991f19a9fb6975cdbbd73c65f4a2681348e63a141a2192a5f10                           0.8s
 => => extracting sha256:f910a506b6cb1dbec766725d70356f695ae2bf2bea6224dbe8c7c6ad4f3664a2                           0.0s
 => => extracting sha256:c2274a1a0e2786ee9101b08f76111f9ab8019e368dce1e325d3c284a0ca33397                           1.7s
 => [2/2] COPY ./target/*.jar app.jar                                                                               1.6s
 => exporting to image                                                                                              0.3s
 => => exporting layers                                                                                             0.2s
 => => writing image sha256:f60b1faf57e5de692ee7a0a392d2360d0dbe27b2776fe6418fd4c216fe910286                        0.0s
 => => naming to docker.io/library/alpha                                                                            0.0s
```

docker images

docker build -t alpha1.

```
ubuntu@ip-172-31-32-249:~/docker_handson/spring-cloud-kubernetes/kubernetes-configmap-reload$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
alpha        latest    f60b1faf57e5   23 seconds ago   144MB
ubuntu@ip-172-31-32-249:~/docker_handson/spring-cloud-kubernetes/kubernetes-configmap-reload$ docker build -t alpha1 .
[+] Building 0.3s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 184B                                                                                0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/library/openjdk:8-jdk-alpine                                             0.1s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 103B                                                                                   0.0s
 => [1/2] FROM docker.io/library/openjdk:8-jdk-alpine@sha256:94792824df2df33402f201713f932b58cb9de94a0cd524164a0f2  0.0s
 => CACHED [2/2] COPY ./target/*.jar app.jar                                                                        0.0s
 => exporting to image                                                                                              0.0s
 => => exporting layers                                                                                             0.0s
 => => writing image sha256:f60b1faf57e5de692ee7a0a392d2360d0dbe27b2776fe6418fd4c216fe910286                        0.0s
 => => naming to docker.io/library/alpha1                                                                           0.0s
```

Step 7: Run the Docker Container

Replace <imageID> with the actual image ID you get from the previous step:

docker images

docker run -itd <imageID> -name beta

docker ps -a

```
ubuntu@ip-172-31-32-249:~/docker_handson/spring-cloud-kubernetes/kubernetes-configmap-reload$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
alpha1       latest    f60b1faf57e5   About a minute ago   144MB
alpha        latest    f60b1faf57e5   About a minute ago   144MB
ubuntu@ip-172-31-32-249:~/docker_handson/spring-cloud-kubernetes/kubernetes-configmap-reload$ docker run -itd f60b1faf57e5 –name beta
a814adb7751388878a2980a88168721016a973a65c885fa97c6c8087d5389763
ubuntu@ip-172-31-32-249:~/docker_handson/spring-cloud-kubernetes/kubernetes-configmap-reload$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
a814adb77513   f60b1faf57e5   "/bin/sh -c 'exec ja…"   22 seconds ago   Up 21 seconds             beautiful_thompson
```

Step 8: Access the Running Container

Replace <containerID> with the actual container ID:

docker exec -it <containerID> /bin/sh

ls -l

```
5.3.80.170.219 (ubuntu)
/ # ls -l
total 38640
-rw-rw-r--    1 root     root      39501161 Aug  8 14:05 app.jar
drwxr-xr-x    2 root     root          4096 May  9  2019 bin
drwxr-xr-x    3 root     root          4096 Aug  8 14:14 c
drwxr-xr-x    5 root     root           360 Aug  8 14:14 dev
drwxr-xr-x    1 root     root          4096 Aug  8 14:14 etc
drwxr-xr-x    2 root     root          4096 May  9  2019 home
drwxr-xr-x    2 root     root          4096 Aug  8 14:14 la
drwxr-xr-x    1 root     root          4096 May 11  2019 lib
drwxr-xr-x    5 root     root          4096 May  9  2019 media
drwxr-xr-x    2 root     root          4096 May  9  2019 mnt
drwxr-xr-x    2 root     root          4096 May  9  2019 opt
dr-xr-xr-x  168 root     root             0 Aug  8 14:14 proc
drwx------    1 root     root          4096 Aug  8 14:19 root
drwxr-xr-x    2 root     root          4096 May  9  2019 run
drwxr-xr-x    2 root     root          4096 May  9  2019 sbin
drwxr-xr-x    2 root     root          4096 May  9  2019 srv
dr-xr-xr-x   13 root     root             0 Aug  8 14:14 sys
drwxrwxrwt    1 root     root          4096 Aug  8 14:14 tmp
drwxr-xr-x    1 root     root          4096 May 11  2019 usr
drwxr-xr-x    1 root     root          4096 May  9  2019 var
/ # ls -lh app.jar
-rw-rw-r--    1 root     root       37.7M Aug  8 14:05 app.jar
/ #
```

Step 9: Verify the app.jar File in the Docker Container
Inside the container, navigate to the directory where the app.jar file is supposed to be and verify its presence.

Ps ef | grep -I app.jar