

## Lesson 02 Demo 02

### Running Java App Image as a Container

**Objective:** To run a Java app image as a container, understand the container lifecycle, and perform various actions on the container

**Tool required:** Eclipse IDE and Docker

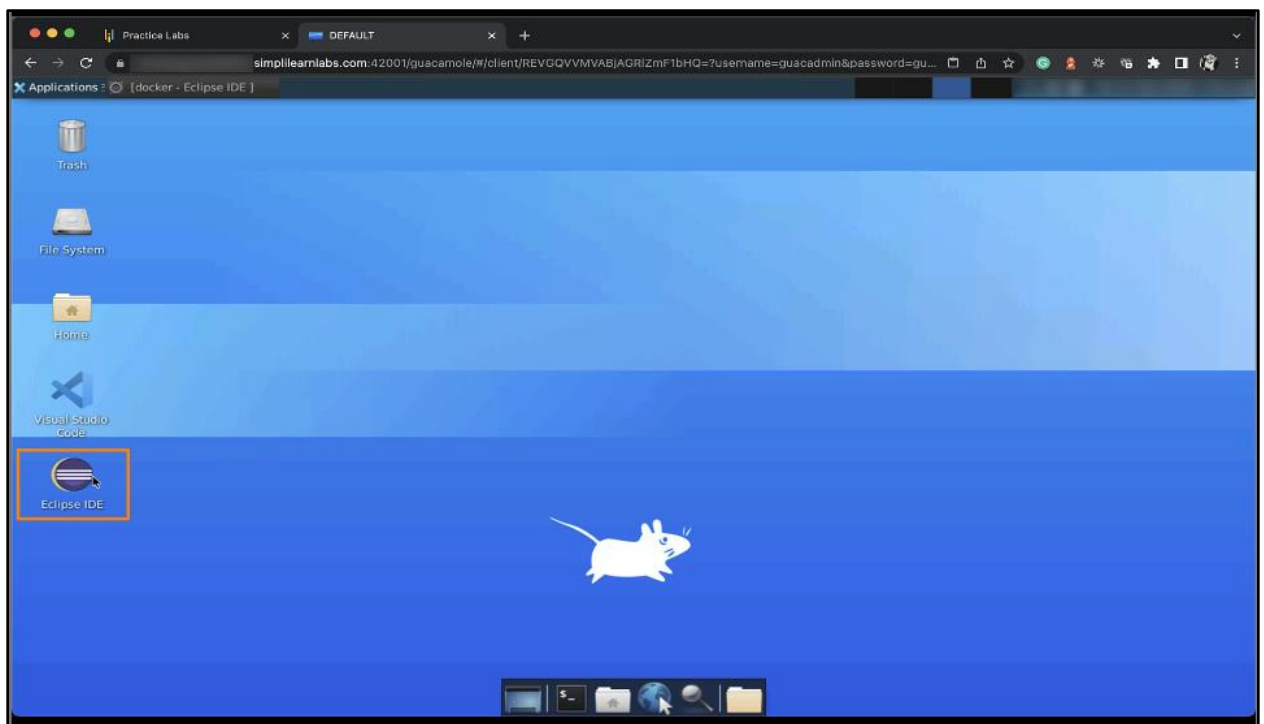
**Prerequisites:** Lesson 02 Demo 01

#### Steps to be followed:

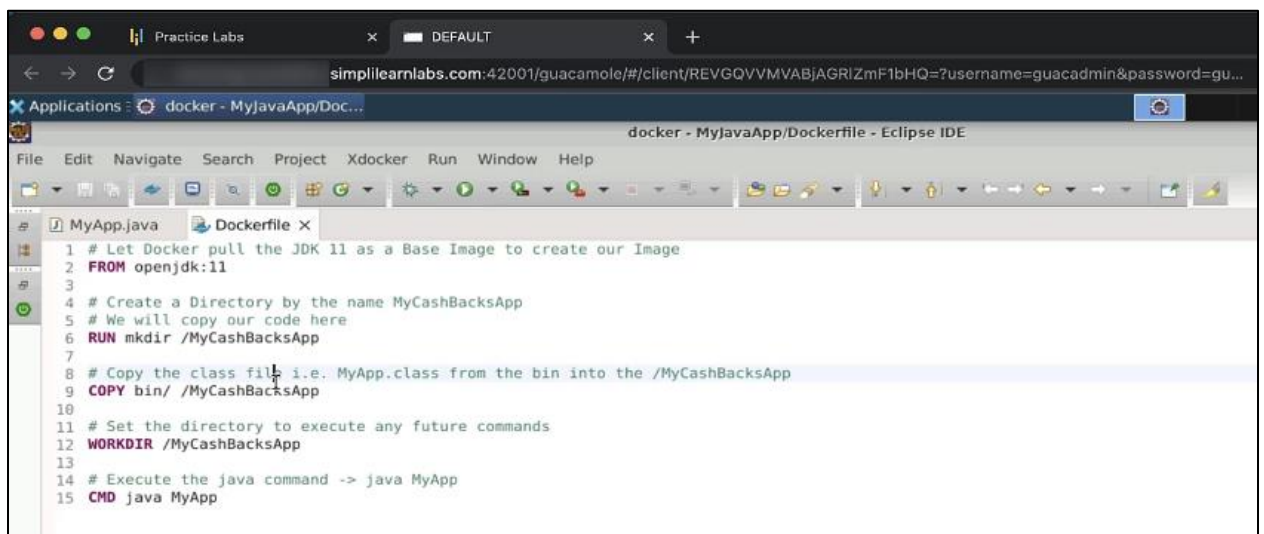
1. Executing the **docker images** command
2. Executing the **docker run** command
3. Restarting the container

#### Step 1: Executing the docker images command

- 1.1 Open **Eclipse IDE**

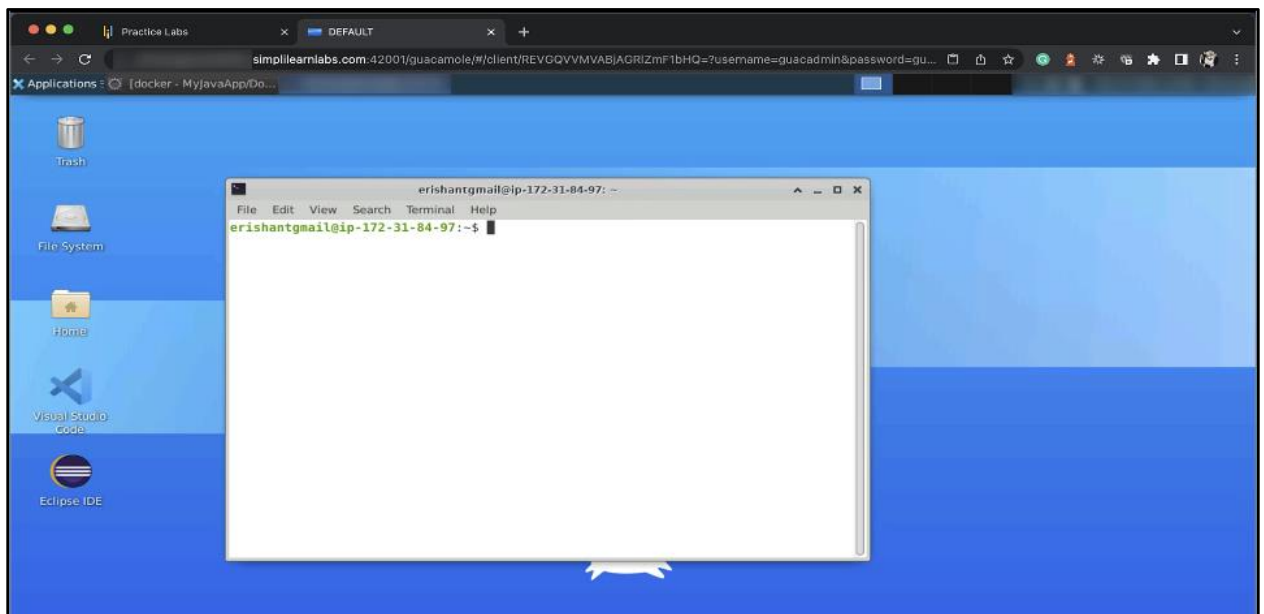


## 1.2 Open **Dockerfile** created in the previous demo

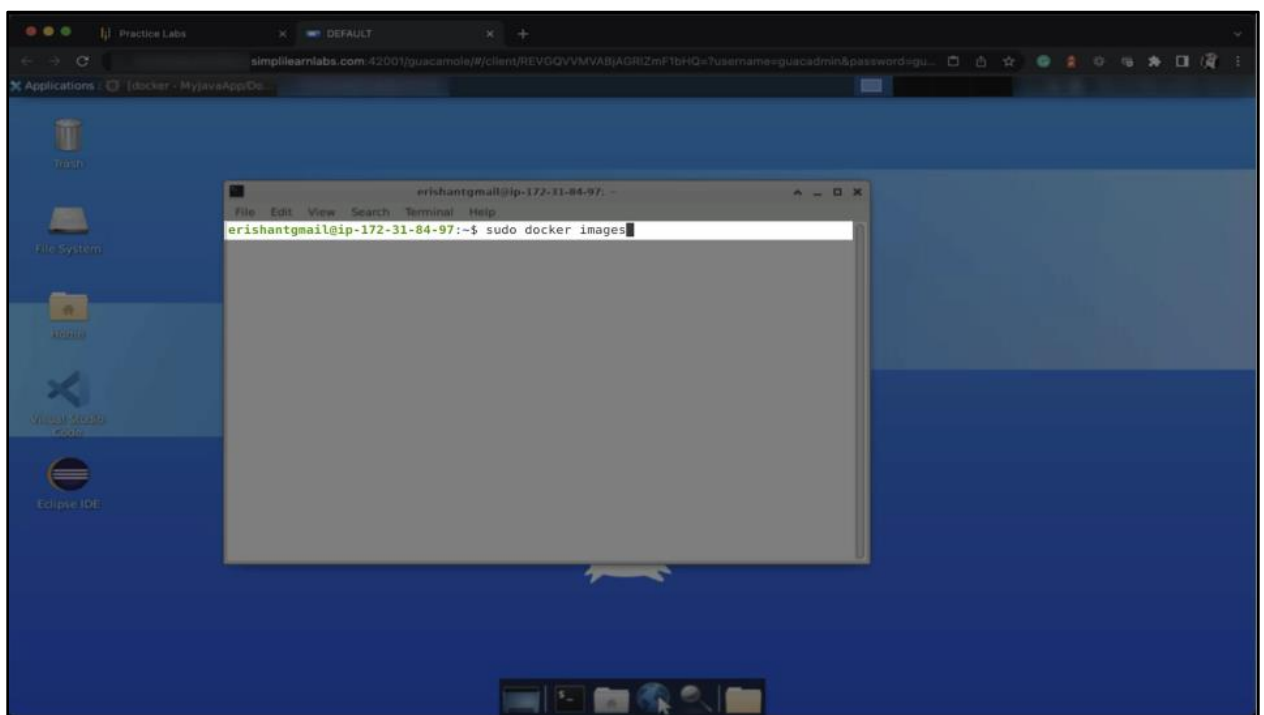


Please note that two Docker images were already created in the previous demo: one for the Java application as a cashback (1.0) and another with an infinite loop running (2.0).

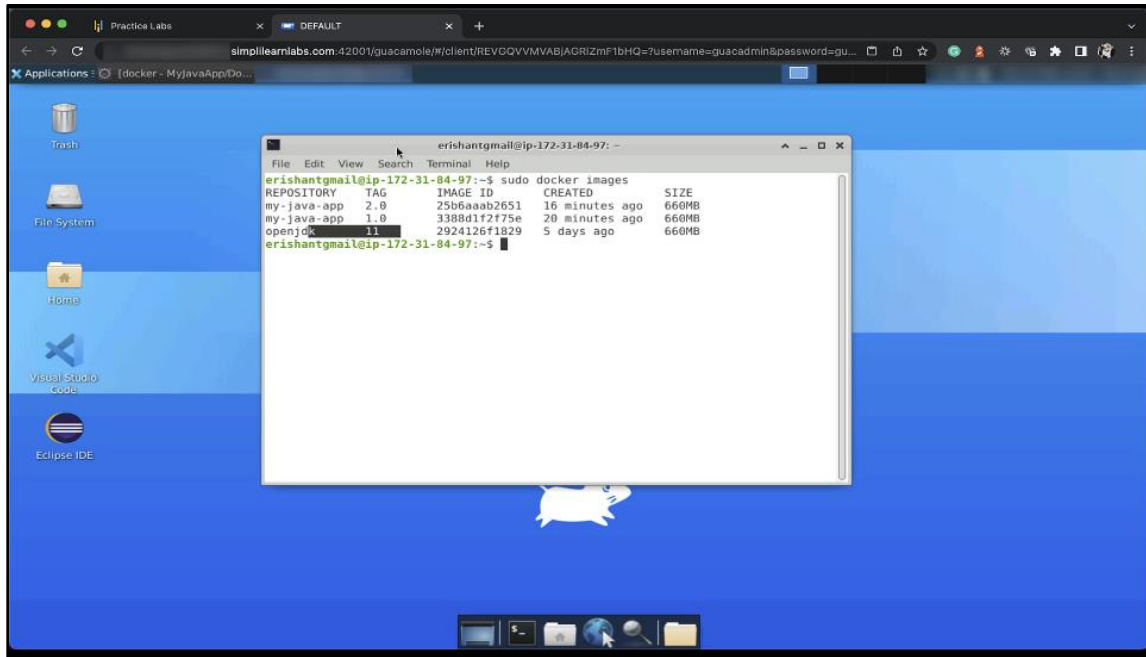
### 1.3 Open the terminal window



### 1.4 Run the **sudo docker images** command to see the available Docker images

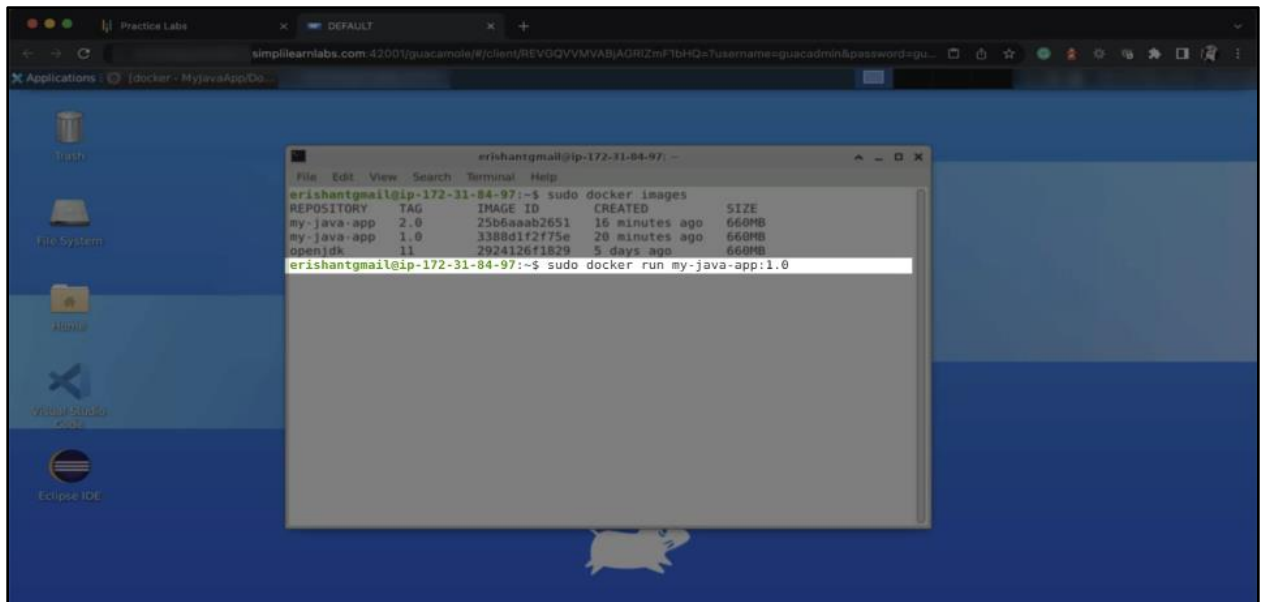


The output should display two images tagged as **1.0** and **2.0**, representing the Java application programs.

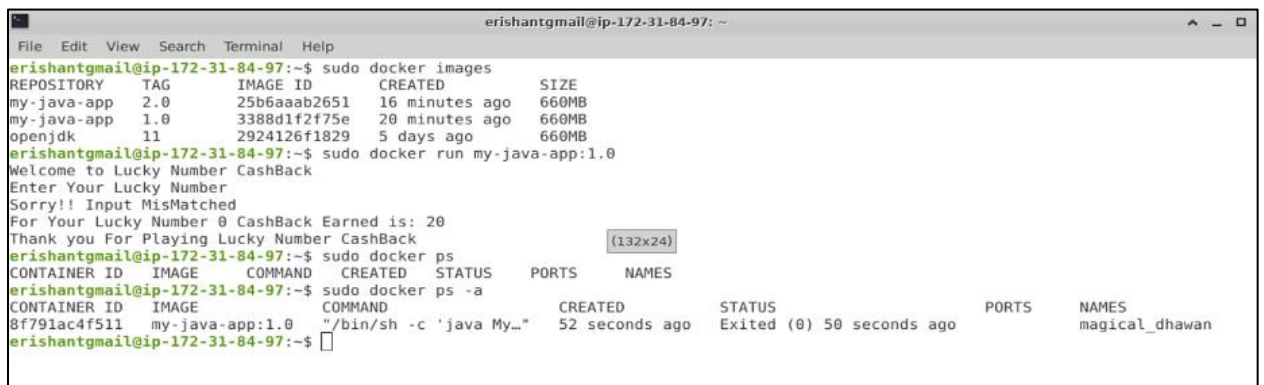


## Step 2: Executing the docker run command

2.1 Run the **sudo docker run my-java-app:1.0** command to run the Java app image as a container



2.2 Run the **sudo docker ps -a** command to see the list of processes



You should find a process named **magical\_dhawan**.

2.3 Remove the container using the **sudo docker rm magical\_dhawan** command

```
erishantgmail@ip-172-31-84-97: ~
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
my-java-app 2.0 25b6aaab2651 16 minutes ago 660MB
my-java-app 1.0 3388d1f2f75e 20 minutes ago 660MB
openjdk 11 2924126f1829 5 days ago 660MB
erishantgmail@ip-172-31-84-97:~$ sudo docker run my-java-app:1.0
Welcome to Lucky Number CashBack
Enter Your Lucky Number
Sorry!! Input MisMatched
For Your Lucky Number 0 CashBack Earned is: 20
Thank you For Playing Lucky Number CashBack
erishantgmail@ip-172-31-84-97:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
erishantgmail@ip-172-31-84-97:~$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8f791ac4f511 my-java-app:1.0 "/bin/sh -c 'java My..." 52 seconds ago Exited (0) 50 seconds ago magical_dhawan
erishantgmail@ip-172-31-84-97:~$ sudo docker start magical_dhawan
magical_dhawan
erishantgmail@ip-172-31-84-97:~$ sudo docker start magical_dhawan
magical_dhawan
erishantgmail@ip-172-31-84-97:~$ sudo docker rm magical_dhawan
magical_dhawan
erishantgmail@ip-172-31-84-97:~$
```

2.4 Run the **sudo docker run -i --name myapp1 my-java-app:1.0** command to create a container named **myapp1** from the image **my-java-app:1.0**

```
erishantgmail@ip-172-31-84-97: ~
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
my-java-app 2.0 25b6aaab2651 19 minutes ago 660MB
my-java-app 1.0 3388d1f2f75e 23 minutes ago 660MB
openjdk 11 2924126f1829 5 days ago 660MB
erishantgmail@ip-172-31-84-97:~$ sudo docker run -i --name myapp1
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
erishantgmail@ip-172-31-84-97:~$ sudo docker run -i --name myapp1 my-java-app:1.0
Welcome to Lucky Number CashBack
Enter Your Lucky Number
```

2.5 Enter the lucky number of **3**. The program should print **Thank you For Playing Lucky Number CashBack**, indicating that your application is encapsulated within the Docker container named **myapp1**.

```

erishantgmail@ip-172-31-84-97: ~
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my-java-app   2.0       25b6aaab2651   19 minutes ago 660MB
my-java-app   1.0       3388d1f2f75e   23 minutes ago 660MB
openjdk       11        2924126f1829   5 days ago    660MB
erishantgmail@ip-172-31-84-97:~$ sudo docker run -i --name myappl
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
erishantgmail@ip-172-31-84-97:~$ sudo docker run -i --name myappl my-java-app:1.0
Welcome to Lucky Number CashBack
Enter Your Lucky Number
3
For Your Lucky Number 3 CashBack Earned is: 500
Thank you For Playing Lucky Number CashBack
erishantgmail@ip-172-31-84-97:~$

```

2.6 Similarly, create an image for **my-java-app 2.0** by running the **sudo docker run -i --name myapp2 my-java-app:2.0** command

```

erishantgmail@ip-172-31-84-97: ~
File Edit View Search Terminal Help
Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
erishantgmail@ip-172-31-84-97:~$ sudo docker run -i --name myappl my-java-app:1.0
Welcome to Lucky Number CashBack
Enter Your Lucky Number
3
For Your Lucky Number 3 CashBack Earned is: 500
Thank you For Playing Lucky Number CashBack
erishantgmail@ip-172-31-84-97:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
3c3067993158   my-java-app:1.0 "/bin/sh -c 'java My..." 31 seconds ago Exited (0) 15 seconds ago
erishantgmail@ip-172-31-84-97:~$ sudo docker start -i myappl
Welcome to Lucky Number CashBack
Enter Your Lucky Number
5
For Your Lucky Number 5 CashBack Earned is: 80
Thank you For Playing Lucky Number CashBack
erishantgmail@ip-172-31-84-97:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my-java-app   2.0       25b6aaab2651   21 minutes ago 660MB
my-java-app   1.0       3388d1f2f75e   25 minutes ago 660MB
openjdk       11        2924126f1829   5 days ago    660MB
erishantgmail@ip-172-31-84-97:~$ sudo docker run -i --name myapp2 my-java-app:2.0

```

2.7 Use the **sudo docker ps -a** command to check the status of the Docker containers

```
erishantgmail@ip-172-31-84-97: ~
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My..."  16 seconds ago  Up 14 seconds
myapp2
erishantgmail@ip-172-31-84-97:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My..."  35 seconds ago  Up 33 seconds
myapp2
3c3067993158   my-java-app:1.0  "/bin/sh -c 'java My..."  2 minutes ago  Exited (0) About a min
ute ago
myapp1
erishantgmail@ip-172-31-84-97:~$
```

You should see two containers: one in an exited state and one running.

2.8 Stop the Docker container **myapp2** by running the **sudo docker stop myapp2** command

```
erishantgmail@ip-172-31-84-97: ~
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My..."  16 seconds ago  Up 14 seconds
myapp2
erishantgmail@ip-172-31-84-97:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My..."  35 seconds ago  Up 33 seconds
myapp2
3c3067993158   my-java-app:1.0  "/bin/sh -c 'java My..."  2 minutes ago  Exited (0) About a minute ago
myapp1
erishantgmail@ip-172-31-84-97:~$ sudo docker pause my-java-app:2.0
Error response from daemon: No such container: my-java-app:2.0
erishantgmail@ip-172-31-84-97:~$ sudo docker pause myapp2
```

## Step 3: Restarting the container

3.1 Restart the container myapp2 using the **sudo docker start myapp2** command



```
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My_..."  16 seconds ago  Up 14 seconds        myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My_..."  35 seconds ago  Up 33 seconds        myapp2
3c3067993158   my-java-app:1.0  "/bin/sh -c 'java My_..."  2 minutes ago   Exited (0) About a minute ago   myapp1
erishantgmail@ip-172-31-84-97: ~$ sudo docker pause my-java-app:2.0
Error response from daemon: No such container: my-java-app:2.0
erishantgmail@ip-172-31-84-97: ~$ sudo docker pause myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker stop myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker start myapp2
```

3.2 Stop the container using the **sudo docker stop myapp2** command

```
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My_..."  16 seconds ago  Up 14 seconds        myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My_..."  35 seconds ago  Up 33 seconds        myapp2
3c3067993158   my-java-app:1.0  "/bin/sh -c 'java My_..."  2 minutes ago   Exited (0) About a minute ago   myapp1
erishantgmail@ip-172-31-84-97: ~$ sudo docker pause my-java-app:2.0
Error response from daemon: No such container: my-java-app:2.0
erishantgmail@ip-172-31-84-97: ~$ sudo docker pause myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker stop myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker start myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker stop myapp2
```

3.3 List all containers using the **sudo docker ps -a** command

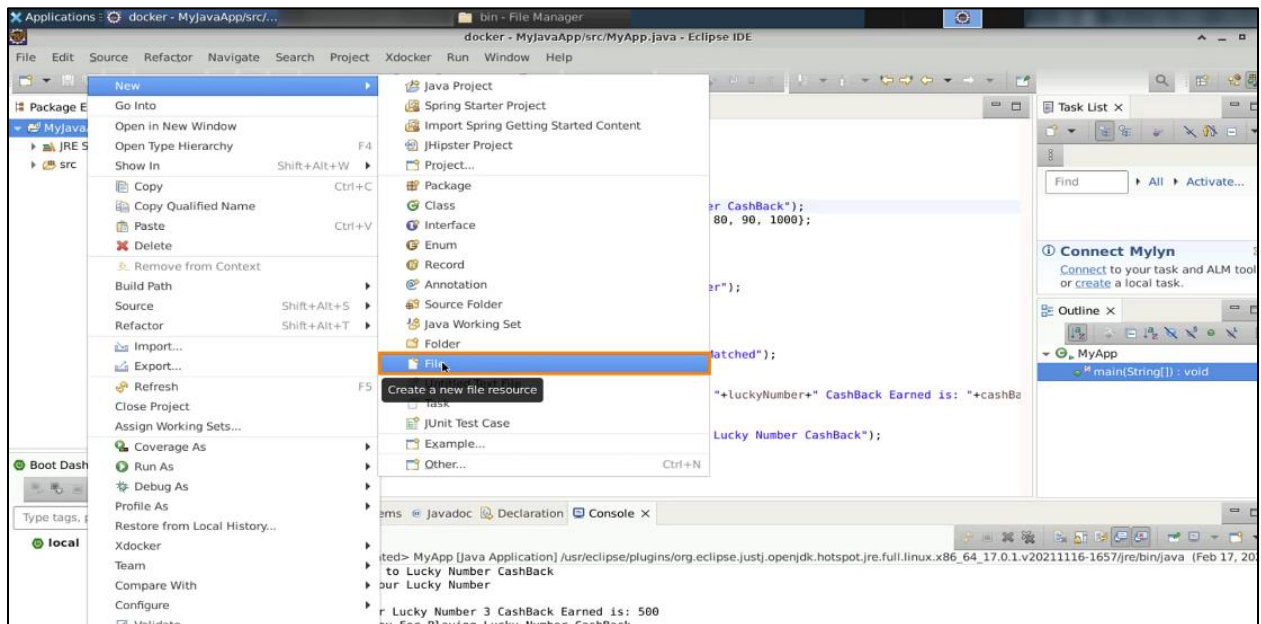
```
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My_..."  16 seconds ago  Up 14 seconds        myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My_..."  35 seconds ago  Up 33 seconds        myapp2
3c3067993158   my-java-app:1.0  "/bin/sh -c 'java My_..."  2 minutes ago   Exited (0) About a minute ago   myapp1
erishantgmail@ip-172-31-84-97: ~$ sudo docker pause my-java-app:2.0
Error response from daemon: No such container: my-java-app:2.0
erishantgmail@ip-172-31-84-97: ~$ sudo docker pause myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker stop myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker start myapp2
myapp2
erishantgmail@ip-172-31-84-97: ~$ sudo docker stop myapp2
^C
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
dcda42980e77   my-java-app:2.0  "/bin/sh -c 'java My_..."  4 minutes ago   Exited (137) 5 seconds ago        myapp2
3c3067993158   my-java-app:1.0  "/bin/sh -c 'java My_..."  5 minutes ago   Exited (0) 4 minutes ago          myapp1
erishantgmail@ip-172-31-84-97: ~$
```

You will see that both containers are now in the exited state.

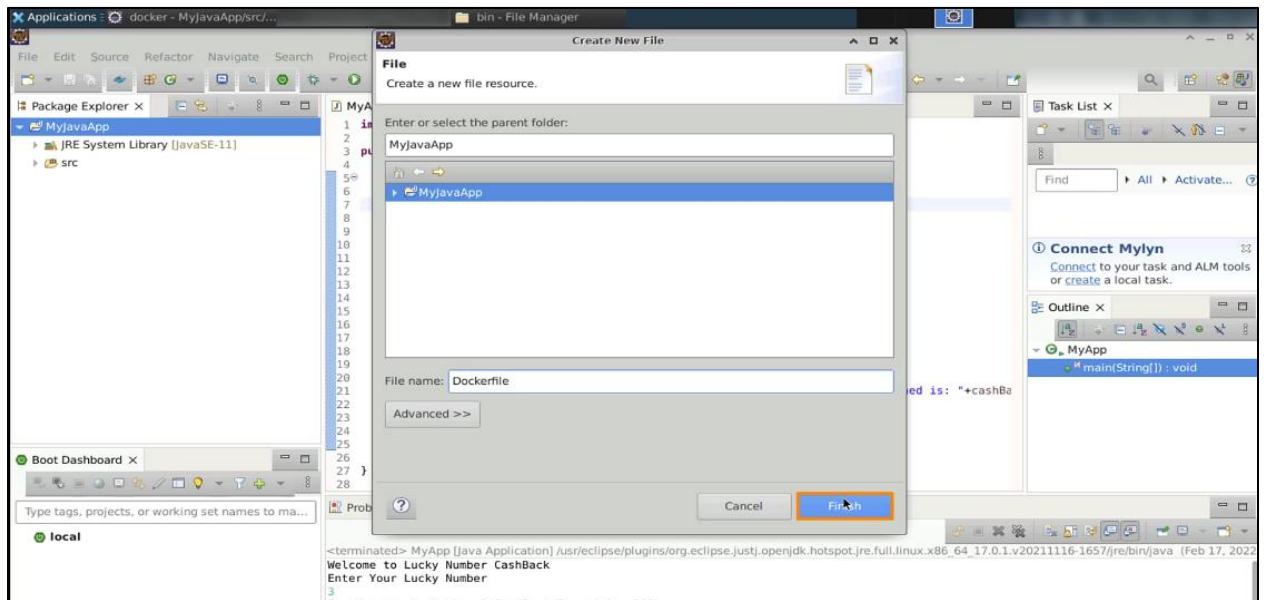
### 3.4 Remove the containers myapp1 and myapp2 using the **sudo docker rm myapp1** and **sudo docker rm myapp2** commands, respectively

```
erishantgmail@ip-172-31-84-97: ~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED      STATUS      PORTS      NAMES
dcda42980e77       my-java-app:2.0    "/bin/sh -c 'java My..." 35 seconds ago Up 33 seconds      myapp2
3c3067993158       my-java-app:1.0    "/bin/sh -c 'java My..." 2 minutes ago Exited (0) About a minute ago myapp1
erishantgmail@ip-172-31-84-97:~$ sudo docker pause my-java-app:2.0
Error response from daemon: No such container: my-java-app:2.0
erishantgmail@ip-172-31-84-97:~$ sudo docker pause myapp2
myapp2
erishantgmail@ip-172-31-84-97:~$ sudo docker stop myapp2
myapp2
erishantgmail@ip-172-31-84-97:~$ sudo docker start myapp2
myapp2
erishantgmail@ip-172-31-84-97:~$ sudo docker stop myapp2
^C
erishantgmail@ip-172-31-84-97:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED      STATUS      PORTS      NAMES
dcda42980e77       my-java-app:2.0    "/bin/sh -c 'java My..." 4 minutes ago Exited (137) 5 seconds ago      myapp2
3c3067993158       my-java-app:1.0    "/bin/sh -c 'java My..." 5 minutes ago Exited (0) 4 minutes ago      myapp1
erishantgmail@ip-172-31-84-97:~$ sudo docker rm myapp2
myapp2
erishantgmail@ip-172-31-84-97:~$
```

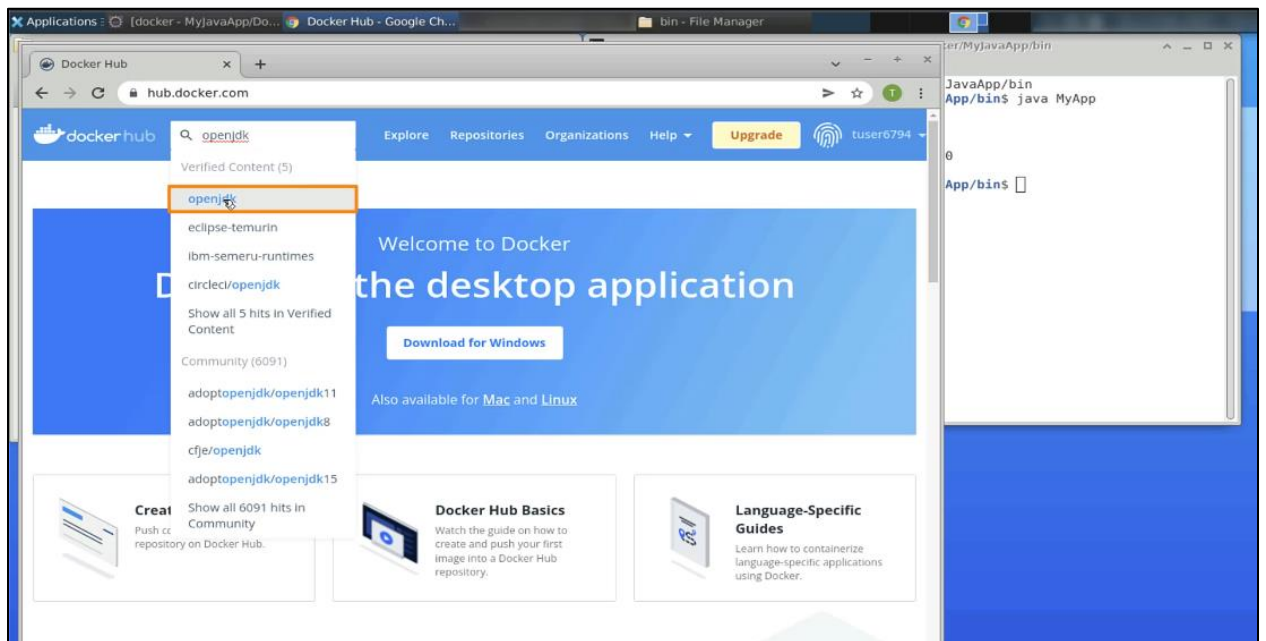
After removing the containers, no containers will be available, but Docker images will still exist. The container lifecycle is straightforward: it is created, moves to the running state, and can be paused, stopped, or removed.



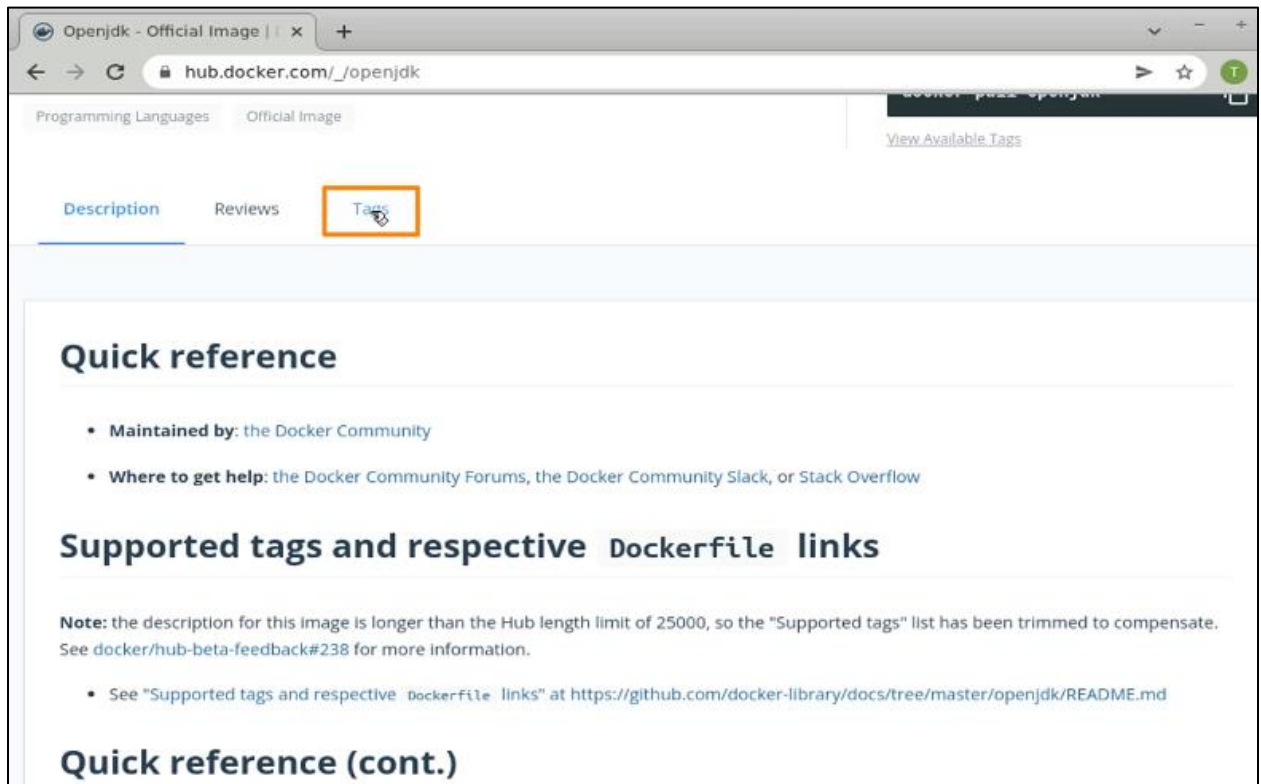
### 3.5 Name the file as **Dockerfile** and click **Finish**



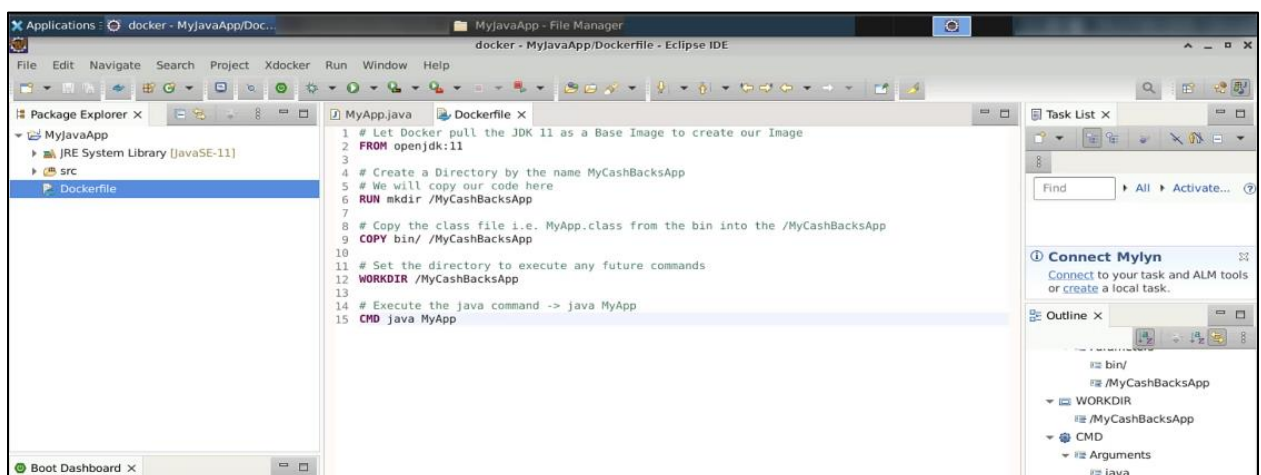
3.6 Open a web browser, navigate to **hub.docker.com**, and search for **openjdk**



3.7 Check for the available tags, specifically for JDK 11



3.8 Go back to **Dockerfile** and write the code to pull JDK 11 as the base image for creating the Java app image. Create a directory named **MyCashBacksApp** and copy the code from the **bin** directory into the **MyCashBacksApp** directory



3.9 Open a new terminal window and navigate to the Docker directory and the project's root folder



```
erishantgmail@ip-172-31-84-97: ~/docker/MyJavaApp
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~$ cd docker/MyJavaApp/
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$
```

3.10 Use the **docker build** command with the tag parameter to create a Docker image. Run the following command:  
**sudo docker build -t my-java-app:1.0 .**



```
erishantgmail@ip-172-31-84-97: ~/docker/MyJavaApp
File Edit View Search Terminal Help
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ ls
Dockerfile bin src
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker build -t my-java-app:1.0
"docker build" requires exactly 1 argument.
See 'docker build --help'.

Usage: docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker build -t my-java-app:1.0 .
```

**Note:** The dot at the end specifies the path to the Dockerfile.



### 3.11 Modify the Java program by adding a while loop to run the code continuously until the user enters 0 to quit the program

```

1  public static void main(String[] args) {
2
3      Scanner scanner = new Scanner(System.in);
4      int choice = -1;
5
6      while(true) {
7
8          System.out.println("Welcome to Lucky Number CashBack");
9          int[] cashBacks = {20, 200, 320, 500, 50, 80, 90, 1000};
10         int luckyNumber = 0;
11
12
13         System.out.println("Enter Your Lucky Number");
14
15         try {
16             luckyNumber = scanner.nextInt();
17         } catch (Exception e) {
18             System.out.println("Sorry!! Input MisMatched");
19         }
20
21         System.out.println("For Your Lucky Number "+luckyNumber+" CashBack Earned is: "+cashBacks[luckyNumber]);
22
23         System.out.println("Enter 0 to quit");
24         choice = scanner.nextInt();
25
26         if(choice == 0) {
27             break;
28         }
29     }
30
31     scanner.close();
32     System.out.println("Thank you For Playing Lucky Number CashBack");
33 }

```

### 3.12 Run the modified Java program and test its functionality by entering various lucky numbers. The program should keep asking for a new number until the user enters 0, upon which it will display the message **Thank you For Playing Lucky Number CashBack**.

```

1  public static void main(String[] args) {
2
3      Scanner scanner = new Scanner(System.in);
4      int choice = -1;
5
6      while(true) {
7
8          System.out.println("Welcome to Lucky Number CashBack");
9          int[] cashBacks = {20, 200, 320, 500, 50, 80, 90, 1000};
10         int luckyNumber = 0;
11
12
13         System.out.println("Enter Your Lucky Number");
14
15         try {
16             luckyNumber = scanner.nextInt();
17         } catch (Exception e) {
18             System.out.println("Sorry!! Input MisMatched");
19         }
20
21         System.out.println("For Your Lucky Number "+luckyNumber+" CashBack Earned is: "+cashBacks[luckyNumber]);
22
23         System.out.println("Enter 0 to quit");
24         choice = scanner.nextInt();
25
26         if(choice == 0) {
27             break;
28         }
29     }
30
31     scanner.close();
32     System.out.println("Thank you For Playing Lucky Number CashBack");
33 }

```

```

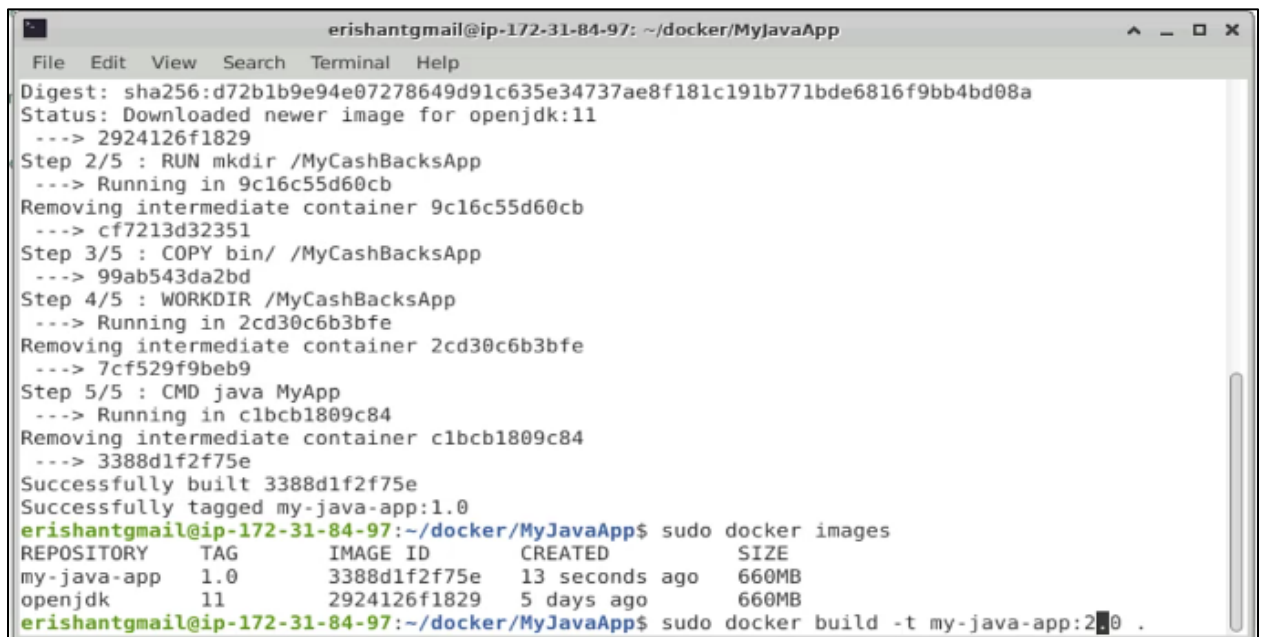
<terminated> MyApp [Java Application] /usr/eclipse/plugins/org.eclipse.justi.openjdk.hotsp
Welcome to Lucky Number CashBack
Enter Your Lucky Number
2
For Your Lucky Number 2 CashBack Earned is: 320
Enter 0 to quit
2
Welcome to Lucky Number CashBack
Enter Your Lucky Number
4
For Your Lucky Number 4 CashBack Earned is: 50
Enter 0 to quit
0
Thank you For Playing Lucky Number CashBack

```

## Step 4: Rebuilding the image

4.1 Go back to the terminal window and rebuild the Docker image using the following command:

```
sudo docker build -t my-java-app:2.0 .
```



```
erishantgmail@ip-172-31-84-97: ~/docker/MyJavaApp
File Edit View Search Terminal Help
Digest: sha256:d72b1b9e94e07278649d91c635e34737ae8f181c191b771bde6816f9bb4bd08a
Status: Downloaded newer image for openjdk:11
---> 2924126f1829
Step 2/5 : RUN mkdir /MyCashBacksApp
---> Running in 9c16c55d60cb
Removing intermediate container 9c16c55d60cb
---> cf7213d32351
Step 3/5 : COPY bin/ /MyCashBacksApp
---> 99ab543da2bd
Step 4/5 : WORKDIR /MyCashBacksApp
---> Running in 2cd30c6b3bfe
Removing intermediate container 2cd30c6b3bfe
---> 7cf529f9beb9
Step 5/5 : CMD java MyApp
---> Running in clbcb1809c84
Removing intermediate container clbcb1809c84
---> 3388d1f2f75e
Successfully built 3388d1f2f75e
Successfully tagged my-java-app:1.0
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
my-java-app         1.0             3388d1f2f75e   13 seconds ago  660MB
openjdk              11              2924126f1829   5 days ago     660MB
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker build -t my-java-app:2.0 .
```

**Note:** This creates a new image with tag **2.0**.

The screenshot shows a web-based IDE with a Dockerfile editor and a terminal. The Dockerfile contains the following instructions:

```

1 # Let Docker pull the JDK 11 as a Base Image to create our Image
2 FROM openjdk:11
3
4 # Create a Directory by the name MyCashBackApp
5 # We will copy our code here
6 RUN mkdir /MyCashBackApp
7
8 # Copy the class file i.e. MyApp.class from
9 COPY bin/ /MyCashBackApp
10
11 # Set the directory to execute any future
12 WORKDIR /MyCashBackApp
13
14 # Execute the java command -> java MyApp
15 CMD java MyApp
  
```

The terminal output shows the following steps:

```

erishantgmail@ip-172-31-84-97: ~/docker/MyJavaApp
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my-java-app   1.0       3388d1f2f75e   13 seconds ago 660MB
openjdk       11        2924126f1829   5 days ago    660MB
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$ sudo docker build -t my-java-app:2.0 .
Sending build context to Docker daemon 10.75kB
Step 1/5 : FROM openjdk:11
--> 2924126f1829
Step 2/5 : RUN mkdir /MyCashBackApp
--> Using cache
--> cf7213d32351
Step 3/5 : COPY bin/ /MyCashBackApp
--> 81e85532659a
Step 4/5 : WORKDIR /MyCashBackApp
--> Running in 9f1ba5647f7d
Removing intermediate container 9f1ba5647f7d
--> ed9b5dc32f02
Step 5/5 : CMD java MyApp
--> Running in 7c0b5f0c5b2a
Removing intermediate container 7c0b5f0c5b2a
--> 25b6aaab2651
Successfully built 25b6aaab2651
Successfully tagged my-java-app:2.0
erishantgmail@ip-172-31-84-97:~/docker/MyJavaApp$
  
```

The output should show **Successfully built** and **Successfully tagged**.

By following these steps, you can create multiple versions of your Java app image. The `-t` parameter is used to add a tag to the image for identification.