


ASSIGNMENT-2

Pulling an image, creating a container and executing it:

```
@d518567295c1/
Microsoft Windows [Version 10.0.25300.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HARSHITA>docker pull centos
Using default tag: latest
latest: Pulling from library/centos
ald0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest

C:\Users\HARSHITA>docker run -it centos
[root@d518567295c1 /]# ls
bin dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
[root@d518567295c1 /]# vi File
[root@d518567295c1 /]# cat File
Hai
Hello
[root@d518567295c1 /]# |
```



The screenshot shows a Windows terminal window with a dark background. The title bar at the top reads '@d518567295c1/'. The terminal content displays 'Welcome to Docker' followed by a series of tilde characters (~) on separate lines. The bottom of the window shows the Windows taskbar with various application icons and a system tray on the right indicating the time as 19:21 on 19-02-2023.

```
@d518567295c1/ x + v
Microsoft Windows [Version 10.0.25300.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HARSHITA>docker pull centos
Using default tag: latest
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest

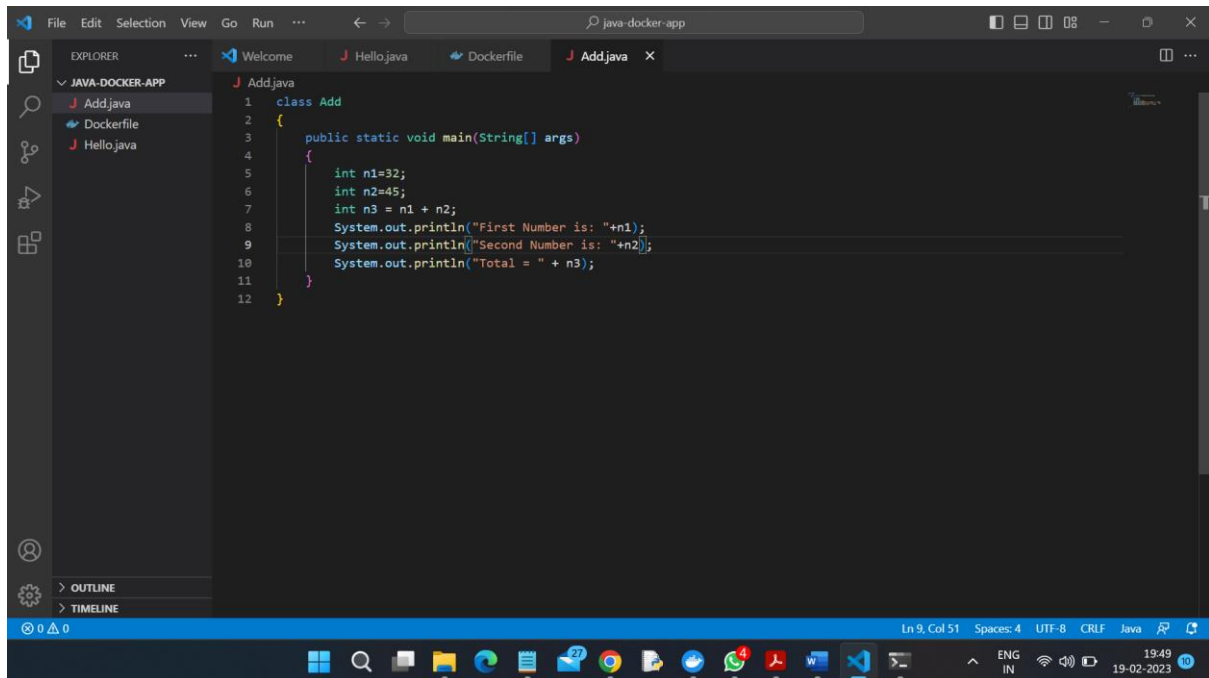
C:\Users\HARSHITA>docker run -it centos
[root@d518567295c1 /]# ls
bin dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
[root@d518567295c1 /]# vi File
[root@d518567295c1 /]# cat File
Hai
Hello
[root@d518567295c1 /]# vi File2
[root@d518567295c1 /]# cat File2
Welcome to Docker
[root@d518567295c1 /]# |
```

```
@d518567295c1/ x + v
[root@d518567295c1 /]# cat File
Hai
Hello
[root@d518567295c1 /]# cp File File3
[root@d518567295c1 /]# cat File3
Hai
Hello
[root@d518567295c1 /]# ls
File File2 File3 bin dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
[root@d518567295c1 /]# rm File2
rm: remove regular file 'File2'? y
[root@d518567295c1 /]# ls
File File3 bin dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
[root@d518567295c1 /]# |
```

Creating Java application and executing it:

I already created a directory called java-docker-app

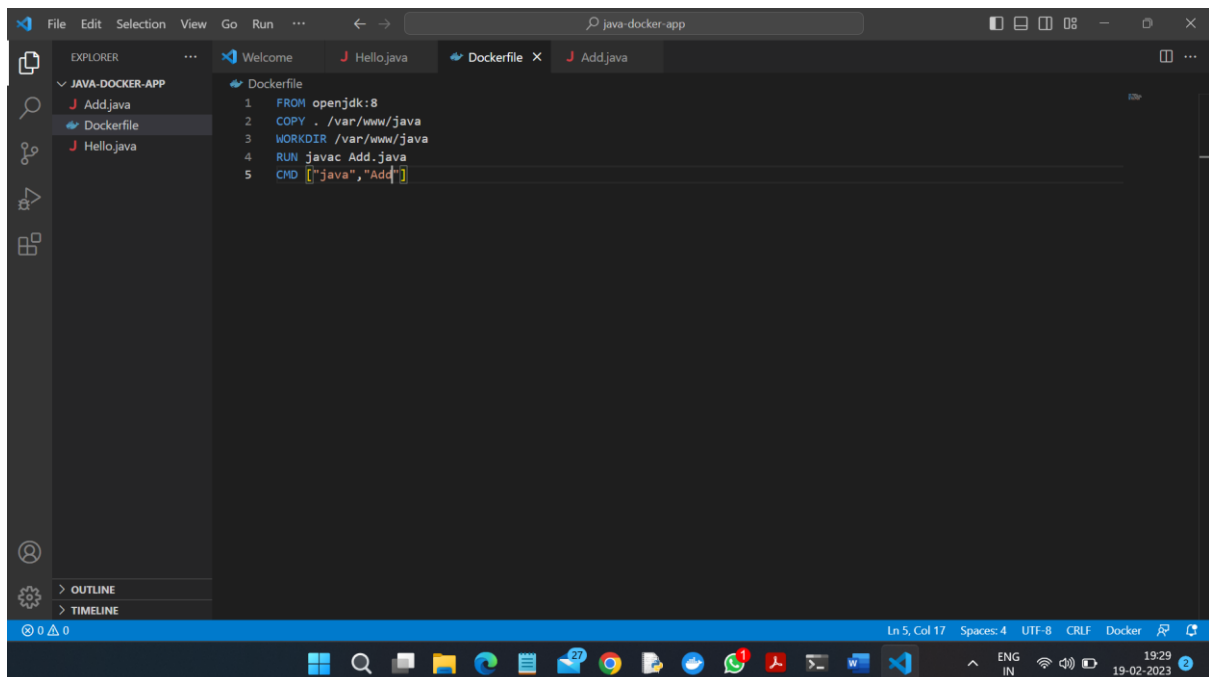
In that app now I am creating a java application and a Dockerfile



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the 'JAVA-DOCKER-APP' directory containing 'Add.java', 'Dockerfile', and 'Hello.java'. The main editor window shows the 'Add.java' file with the following code:

```
1 class Add
2 {
3     public static void main(String[] args)
4     {
5         int n1=32;
6         int n2=45;
7         int n3 = n1 + n2;
8         System.out.println("First Number is: "+n1);
9         System.out.println("Second Number is: "+n2);
10        System.out.println("Total = " + n3);
11    }
12 }
```

The status bar at the bottom indicates 'Ln 9, Col 51', 'Spaces: 4', 'UTF-8', 'CRLF', 'Java', and the system clock shows 19:49 on 19-02-2023.



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the 'JAVA-DOCKER-APP' directory. The main editor window shows the 'Dockerfile' file with the following code:

```
1 FROM openjdk:8
2 COPY . /var/www/java
3 WORKDIR /var/www/java
4 RUN javac Add.java
5 CMD ["java", "-Add"]
```

The status bar at the bottom indicates 'Ln 5, Col 17', 'Spaces: 4', 'UTF-8', 'CRLF', 'Docker', and the system clock shows 19:29 on 19-02-2023.

I changed the directory to the java-docker-app

```
Microsoft Windows [Version 10.0.25300.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HARSHITA>cd java-docker-app

C:\Users\HARSHITA\java-docker-app>docker build -t javaapp .
[+] Building 6.4s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 137B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/openjdk:8                     3.7s
=> [auth] library/openjdk:pull token for registry-1.docker.io                  0.0s
=> [internal] load build context                                                 0.0s
=> => transferring context: 482B                                                0.0s
=> CACHED [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937 0.0s
=> [2/4] COPY . /var/www/java                                                  0.1s
=> [3/4] WORKDIR /var/www/java                                                  0.1s
=> [4/4] RUN javac Add.java                                                    2.3s
=> exporting to image                                                            0.1s
=> => exporting layers                                                            0.1s
=> => writing image sha256:5102f16125cee79ef26350edd9adca883268a6ff3ff62a3c01239f026a3090ec 0.0s
=> => naming to docker.io/library/javaapp                                       0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:\Users\HARSHITA\java-docker-app>
```

```
C:\Users\HARSHITA\java-docker-app>docker run javaapp
First Number is: 32
Second Number is: 45
Total = 77

C:\Users\HARSHITA\java-docker-app>
```