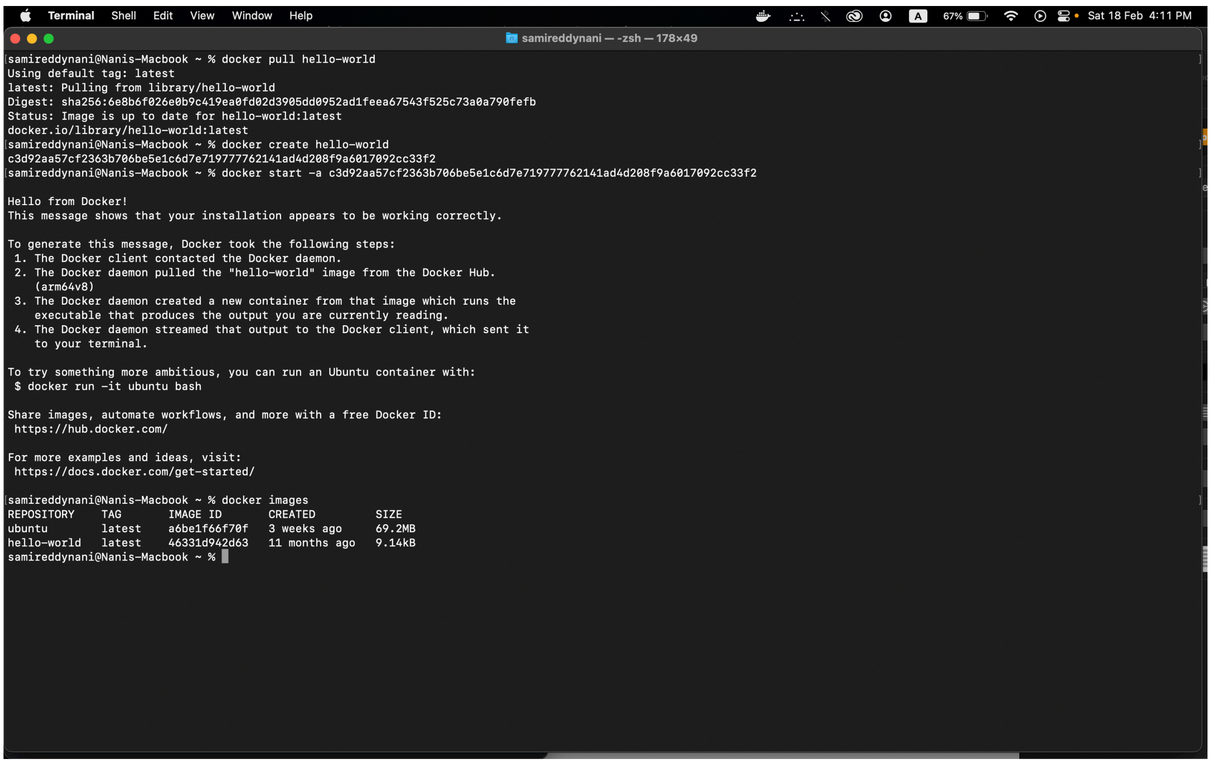
Assignment2

Q1) Pull any image from the docker hub, create its container, and execute it showing the output.

- Docker Hub is a service provided by Docker for finding and sharing container images.
- So, To pull an Image from Docker hub we use the command
    - **docker pull <image_name>**
- So for the simplicity we use the image hello world now
    - **Docker pull hello-world**
- After pulling we have to create a container from the container image using the command
    - **docker create <image_name>**
    - This will give you the container ID we use this ID to start/stop/delete the container
- After creating a container we have to start the container to run it so we use the
    - **docker start <container_ID>**

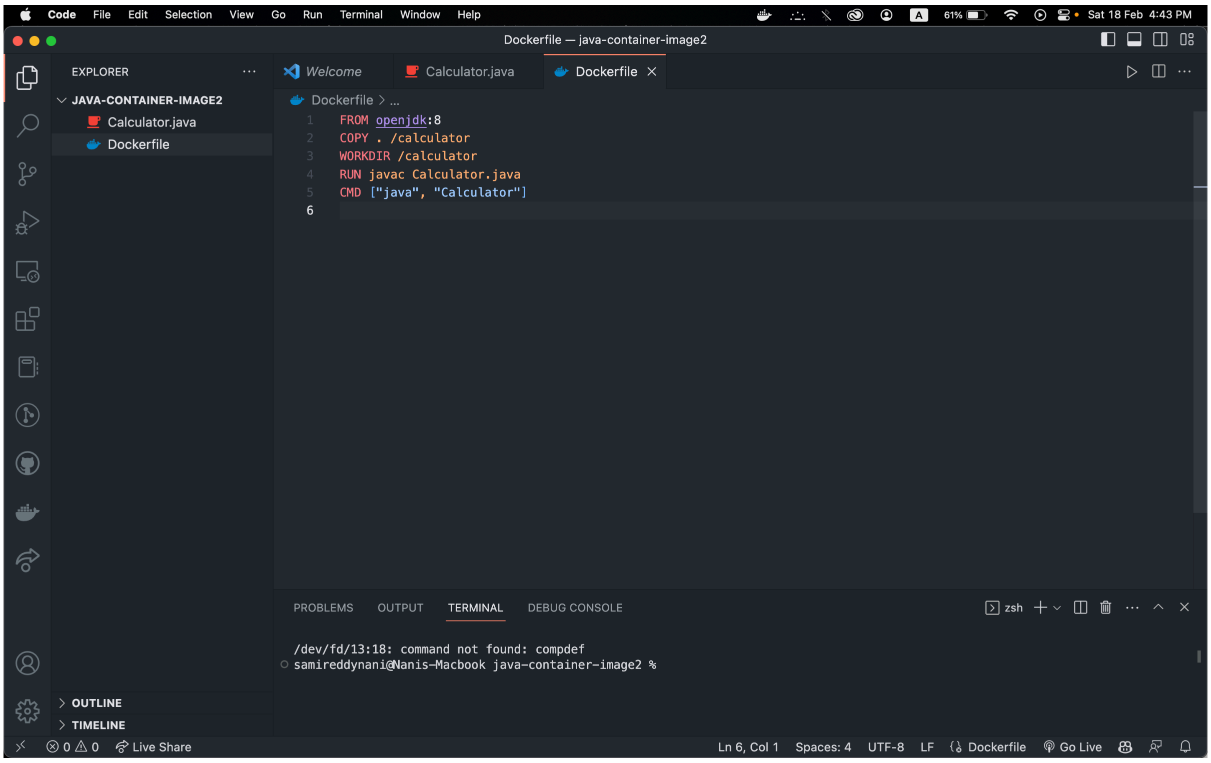


Terminal Shell Edit View Window Help
67%
Sat 18 Feb 4:11 PM
samireddynani — -zsh — 178×49

```
samireddynani@Nanis-Macbook ~ % docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:6e8b6f026e0b9c419ea0fd02d3905dd0952ad1feea67543f525c73a0a790fefb
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
samireddynani@Nanis-Macbook ~ % docker create hello-world
c3d92aa57cf2363b706be5e1c6d7e719777762141ad4d208f9a6017092cc33f2
samireddynani@Nanis-Macbook ~ % docker start -a c3d92aa57cf2363b706be5e1c6d7e719777762141ad4d208f9a6017092cc33f2

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

samireddynani@Nanis-Macbook ~ % docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    a6be1f66f70f   3 weeks ago     69.2MB
hello-world   latest    46331d942d63   11 months ago   9.14kB
samireddynani@Nanis-Macbook ~ % 
```

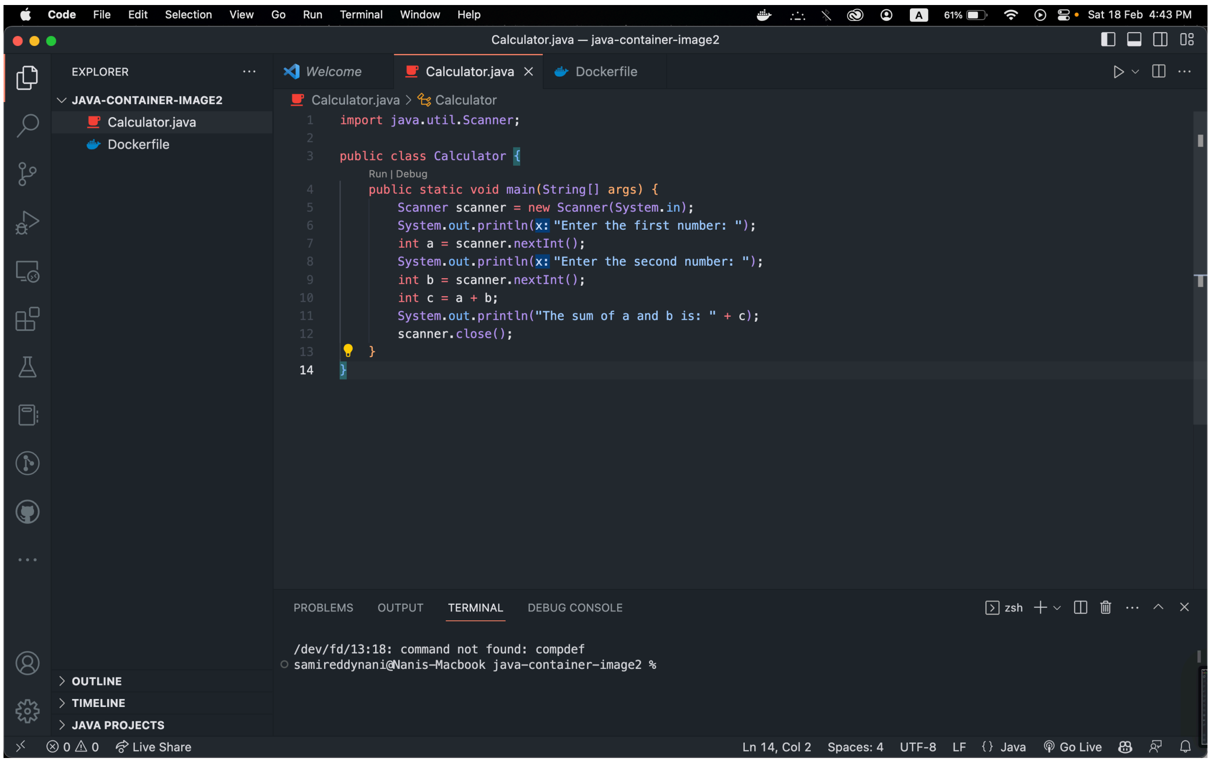
Q2) Create the basic java application, generate its image with necessary files, and execute it with docker.

- A docker container is a light weight environment that contains everything an application needs to run.
- To create a Java container application first we need to create a folder and add two file **Dockerfile, Java program files**



```
FROM openjdk:8
COPY . /calculator
WORKDIR /calculator
RUN javac Calculator.java
CMD ["java", "Calculator"]
```

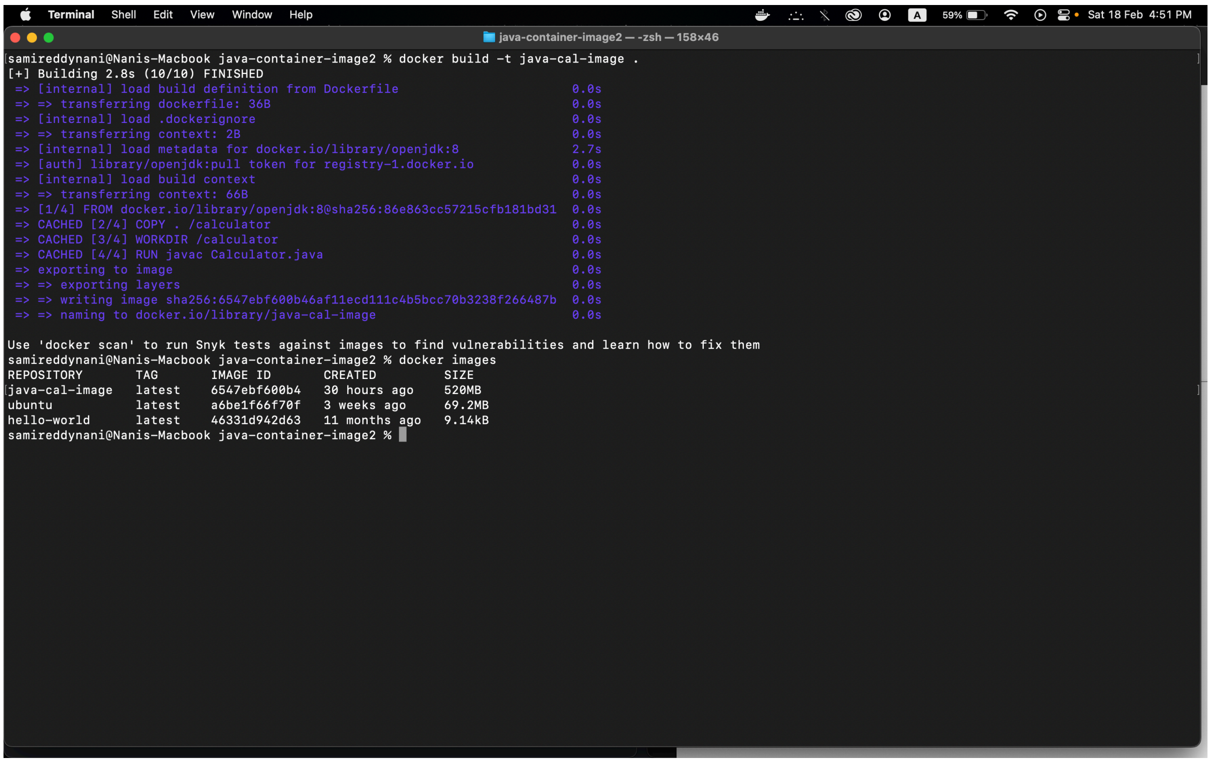
```
/dev/fd/13:18: command not found: compdef
samireddynani@Nanis-Macbook java-container-image2 %
```



```
import java.util.Scanner;

public class Calculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println(x: "Enter the first number: ");
        int a = scanner.nextInt();
        System.out.println(x: "Enter the second number: ");
        int b = scanner.nextInt();
        int c = a + b;
        System.out.println("The sum of a and b is: " + c);
        scanner.close();
    }
}
```

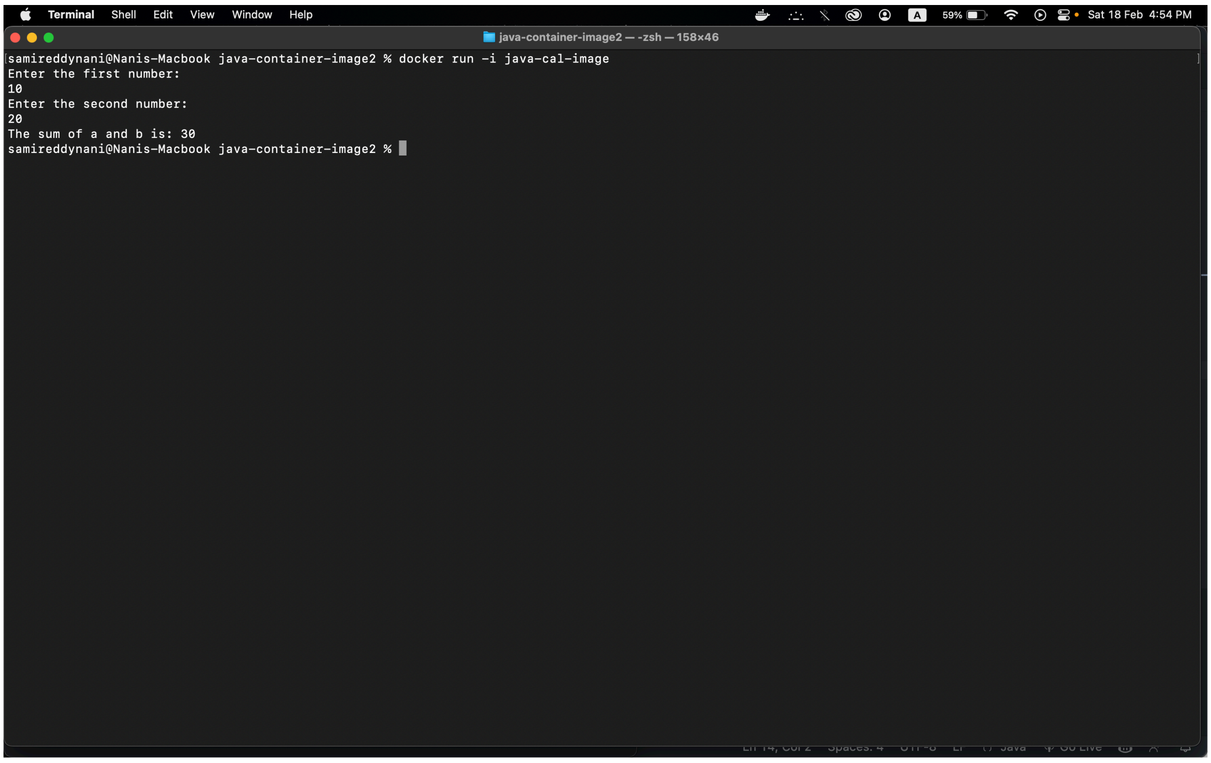
```
/dev/fd/13:18: command not found: compdef
samireddynani@Nanis-Macbook java-container-image2 %
```

- After creating the required files we need to build an **Container Image** from the **Doclerfile** for that we'll use the command **docker build -t <name> <path>**


Terminal Shell Edit View Window Help
59%
Sat 18 Feb 4:51 PM
java-container-image2 — -zsh — 158×46

```
samireddynani@Nanis-Macbook java-container-image2 % docker build -t java-cal-image .
[+] Building 2.8s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                       0.0s
 => => transferring dockerfile: 36B                                        0.0s
 => [internal] load .dockerignore                                          0.0s
 => => transferring context: 2B                                            0.0s
 => [internal] load metadata for docker.io/library/openjdk:8               2.7s
 => [auth] library/openjdk:pull token for registry-1.docker.io             0.0s
 => [internal] load build context                                          0.0s
 => => transferring context: 66B                                           0.0s
 => [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd31  0.0s
 => CACHED [2/4] COPY . /calculator                                        0.0s
 => CACHED [3/4] WORKDIR /calculator                                       0.0s
 => CACHED [4/4] RUN javac Calculator.java                                 0.0s
 => exporting to image                                                     0.0s
 => => exporting layers                                                    0.0s
 => => writing image sha256:6547ebf600b46af11ecd111c4b5bcc70b3238f266487b  0.0s
 => => naming to docker.io/library/java-cal-image                          0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
samireddynani@Nanis-Macbook java-container-image2 % docker images
REPOSITORY       TAG       IMAGE ID       CREATED         SIZE
java-cal-image   latest    6547ebf600b4   30 hours ago    520MB
ubuntu           latest    a6be1f66f70f   3 weeks ago     69.2MB
hello-world      latest    46331d942d63   11 months ago   9.14kB
samireddynani@Nanis-Macbook java-container-image2 %
```

- After building image we have to run the image for that we use **docker run <image>** command


Terminal Shell Edit View Window Help
59%
Sat 18 Feb 4:54 PM
java-container-image2 — -zsh — 158×46

```
samireddynani@Nanis-Macbook java-container-image2 % docker run -i java-cal-image
Enter the first number:
10
Enter the second number:
20
The sum of a and b is: 30
samireddynani@Nanis-Macbook java-container-image2 %
```

-