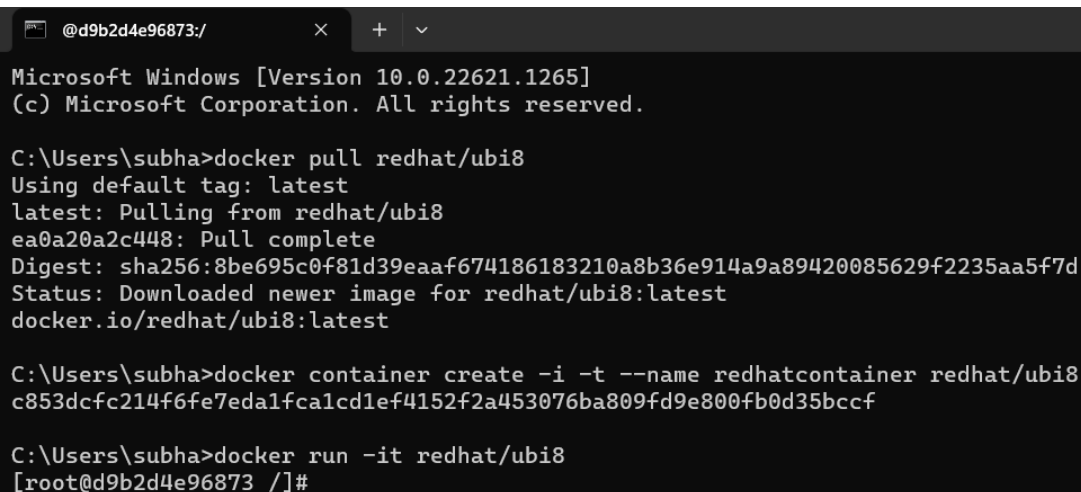


# ASSIGNMENT—2

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

**Answer:**

- To download a particular image, or set of images use : **docker pull**
- The **docker container create** command creates a new container from the specified image, without starting it.
- The **docker run** command first **creates** a writeable container layer over the specified image, and then **starts** it using the specified command.



```
@d9b2d4e96873:/
Microsoft Windows [Version 10.0.22621.1265]
(c) Microsoft Corporation. All rights reserved.

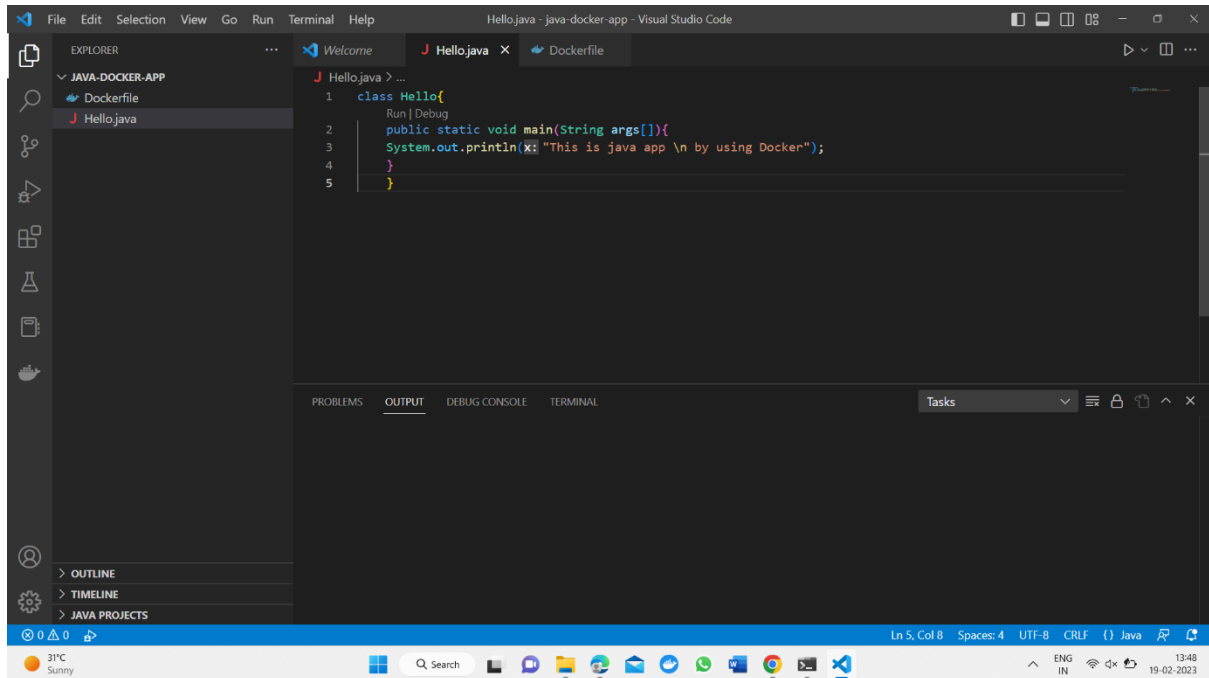
C:\Users\subha>docker pull redhat/ubi8
Using default tag: latest
latest: Pulling from redhat/ubi8
ea0a20a2c448: Pull complete
Digest: sha256:8be695c0f81d39eaf674186183210a8b36e914a9a89420085629f2235aa5f7d
Status: Downloaded newer image for redhat/ubi8:latest
docker.io/redhat/ubi8:latest

C:\Users\subha>docker container create -i -t --name redhatcontainer redhat/ubi8
c853dcfc214f6fe7eda1fca1cd1ef4152f2a453076ba809fd9e800fb0d35bccf

C:\Users\subha>docker run -it redhat/ubi8
[root@d9b2d4e96873 /]#
```

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

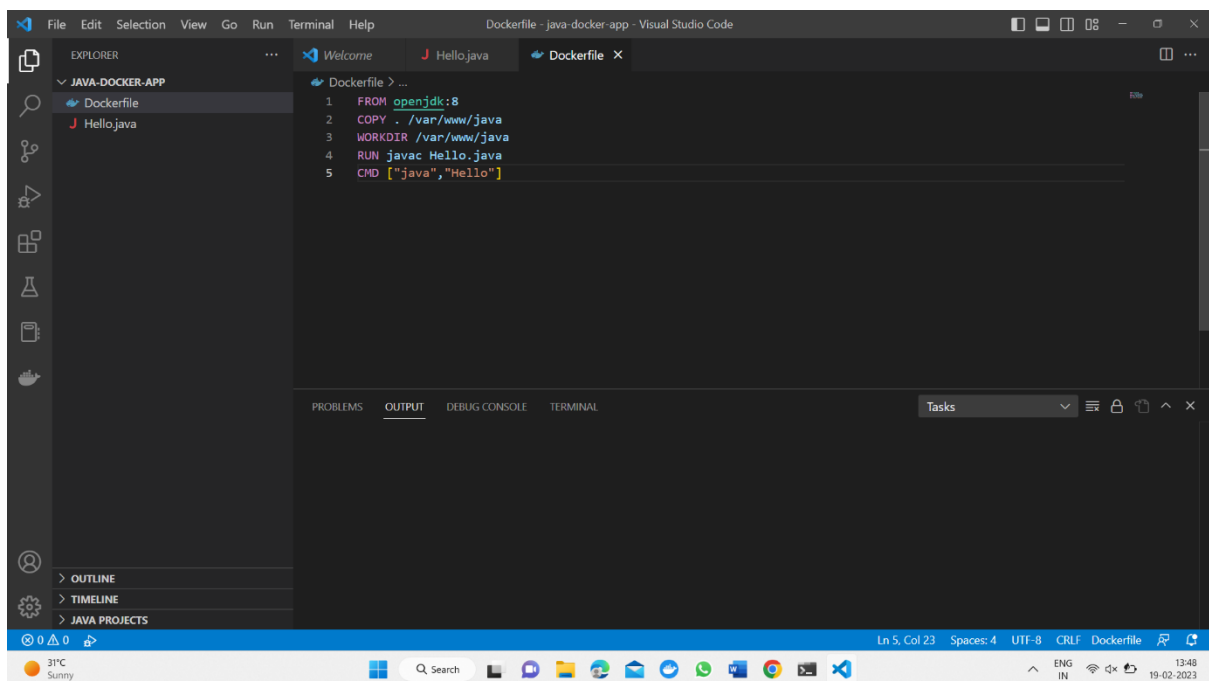
**Answer:**



This screenshot shows the Visual Studio Code editor with the 'Hello.java' file open. The Explorer sidebar on the left shows the project structure with 'JAVA-DOCKER-APP' containing 'Dockerfile' and 'Hello.java'. The main editor area displays the following Java code:

```
1 class Hello{
2     Run | Debug
3     public static void main(String args[]){
4         System.out.println("This is java app \n by using Docker");
5     }
6 }
```

The bottom status bar indicates 'Ln 5, Col 8', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.



This screenshot shows the Visual Studio Code editor with the 'Dockerfile' file open. The Explorer sidebar on the left shows the project structure with 'JAVA-DOCKER-APP' containing 'Dockerfile' and 'Hello.java'. The main editor area displays the following Dockerfile instructions:

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

The bottom status bar indicates 'Ln 5, Col 23', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Dockerfile'.

```
C:\Windows\System32\cmd.exe
C:\Users\subha\OneDrive\Desktop\java-docker-app>docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/gp/access-tokens/

C:\Users\subha\OneDrive\Desktop\java-docker-app>code .

C:\Users\subha\OneDrive\Desktop\java-docker-app>docker build -t java-app .
[+] building 194.4s (18/18) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring Dockerfile: 14KB
=> [internal] load dockerignore
=> transferring context: 2B
=> [internal] load metadata for docker.io/library/openjdk:8
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/library/openjdk:8sha256:86e83cc57215cfb181b319736d8aef625fe0f150577f9eb58bd937f5d52 182.7s
=> resolve docker.io/library/openjdk:8sha256:86e83cc57215cfb181b319736d8aef625fe0f150577f9eb58bd937f5d52cb 0.0s
=> sha256:86e83cc57215cfb181b319736d8aef625fe0f150577f9eb58bd937f5d52cb 1.60kB / 1.60kB
=> sha256:b273080837cc3af285d8e88cfbaa72b93ec7dcb289736c82d8658936f071702 7.81kB / 7.81kB
=> sha256:d96b0869e90457da0918da09173d8c4f8d9bc47b3b4a273cf42f43723d165 5.10kB / 5.10kB
=> sha256:09d7f0ed77c160b3571e47f08093aa8f6da2a653b117651270149c315c216a 10.89kB / 10.89kB
=> sha256:3af2ac8d139765073fc8f1b24fcd8f77996e08218c297fcfa28ca808ff0a217 1.79kB / 1.79kB
=> sha256:08153c26ad57e3b25b43ee0057f6692e5c87d5b982a8a4819ace5c321052 55.00kB / 55.00kB
=> sha256:8aaef778f1509386ef7b3c8b7cda0ff9fa33abb4c3326dc215c7a70ca7116a5 54.53kB / 54.53kB
=> sha256:d83151f15d6683b98f21c3827ac5d5188b180aefb14a1809718eac60926a3d65 5.42kB / 5.42kB
=> sha256:52ab326d386691c0f7e8c8b1b961d0e0824f86a0a4a5d5d0998376d85cc9 218B / 218B
=> sha256:879ada6e0603ba0c1c6ad3110405ba91c7123717b164e6ed07c33e4f0ee 185.32kB / 185.32kB
=> extracting sha256:081c52e26ad57e3b25b439ee0052f6692e5c87d5b982a8a4819ace5c321052 3.0s
=> extracting sha256:d96b086e9e0457da0918da09173d8c4f8d9bc47b3b4a273cf42f43723d165 0.3s
=> extracting sha256:260f70ed77c160b3571e47f08093aa8f6da2a653b117651270149c315c216a 0.3s
=> extracting sha256:9daef720d35893868ef75ac8b7c6eb07fa53abbcb3a26dc218c2ec7bca716a6 1.8s
=> extracting sha256:d83151f15d6683b98f21c3827ac5d5188b180aefb14a1809718eac60926a3d65 0.3s
=> extracting sha256:52ab326d386691c0f7e8c8b1b961d0e0824f86a0a4a5d5d0998376d85cc9 0.0s
=> extracting sha256:879ada6e0603ba0c1c6ad3110405ba91c7123717b164e6ed07c33e4f0ee 2.4s
=> [internal] load build context
=> transferring context: 30B
=> [2/4] COPY ./src/main/java
=> [3/4] WORKDIR /src/main/java
=> [5/4] RUN java hello.java
=> exporting to image
=> exporting layers
=> writing image sha256:f43782263aaa7bd11715d6e98772a381ad93e97c4a61d16c8f385a94ulle705 0.1s
=> naming to docker.io/library/java-app 0.0s

C:\Users\subha\OneDrive\Desktop\java-docker-app>docker run java-app
This is java app
by using Docker

C:\Users\subha\OneDrive\Desktop\java-docker-app>
```