

A simple java application docker image build with maven

Step 1: Create a docker file with “touch Dockerfile”

Add the following contents

```
FROM openjdk:11

WORKDIR /app

COPY HelloWorld.java .
RUN javac HelloWorld.java

CMD ["java", "HelloWorld"]
```

Setp2: built the image

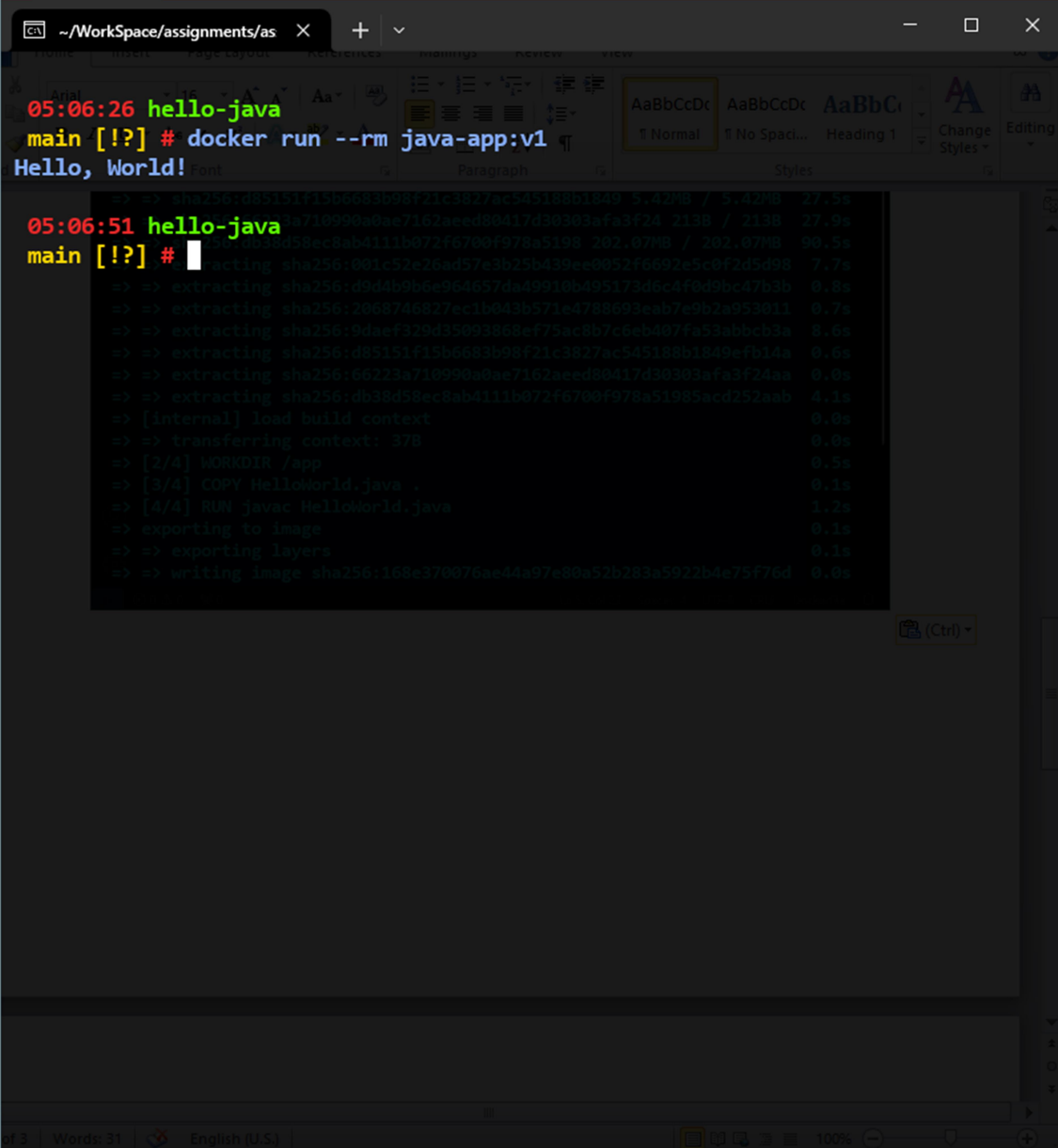
“docker build -t hello-world-app .”



```
05:03:53 hello-java
main [!?] # docker build -t java-app:v1 .
[+] Building 98.9s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 204B                                0.0s
=> [internal] load metadata for docker.io/library/openjdk:11       2.0s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/openjdk:11@sha256:99bac5bf83633e3c7 94.9s
=> => resolve docker.io/library/openjdk:11@sha256:99bac5bf83633e3c73 0.0s
=> => sha256:99bac5bf83633e3c7399aed725c8415e7b569b5 1.04kB / 1.04kB 0.0s
=> => sha256:e81b7f317654b0f26d3993e014b04bcb2925033 1.79kB / 1.79kB 0.0s
=> => sha256:001c52e26ad57e3b25b439ee0052f6692e5c 55.00MB / 55.00MB 38.1s
=> => sha256:47a932d998b743b9b0bcce55aa8ede77de94a6a 6.26kB / 6.26kB 0.0s
=> => sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc 5.16MB / 5.16MB 5.2s
=> => sha256:2068746827ec1b043b571e4788693eab7e9b 10.88MB / 10.88MB 15.1s
=> => sha256:9daef329d35093868ef75ac8b7c6eb407fa5 54.58MB / 54.58MB 34.7s
=> => sha256:d85151f15b6683b98f21c3827ac545188b1849 5.42MB / 5.42MB 27.5s
=> => sha256:66223a710990a0ae7162aead80417d30303afa3f24 213B / 213B 27.9s
=> => sha256:db38d58ec8ab4111b072f6700f978a5198 202.07MB / 202.07MB 90.5s
=> => extracting sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d98 7.7s
=> => extracting sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b 0.8s
=> => extracting sha256:2068746827ec1b043b571e4788693eab7e9b2a953011 0.7s
=> => extracting sha256:9daef329d35093868ef75ac8b7c6eb407fa53abbc3a 8.6s
=> => extracting sha256:d85151f15b6683b98f21c3827ac545188b1849efb14a 0.6s
=> => extracting sha256:66223a710990a0ae7162aead80417d30303afa3f24aa 0.0s
=> => extracting sha256:db38d58ec8ab4111b072f6700f978a51985acd252aab 4.1s
=> [internal] load build context                                  0.0s
=> => transferring context: 37B                                       0.0s
=> [2/4] WORKDIR /app                                           0.5s
=> [3/4] COPY HelloWorld.java .                                0.1s
=> [4/4] RUN javac HelloWorld.java                             1.2s
=> exporting to image                                           0.1s
=> => exporting layers                                              0.1s
=> => writing image sha256:168e370076ae44a97e80a52b283a5922b4e75f76d 0.0s
```

Setp3: run the image in container

“docker run --rm java-app:v1”



The screenshot shows a code editor with a dark theme. The top part of the editor displays a terminal output for a Docker build process. The output shows the build context being loaded and the image being exported. The bottom part of the editor shows a terminal output for a Docker run command. The output shows the container running and printing "Hello, World!".

```
05:06:26 hello-java  
main [!?] # docker run --rm java-app:v1  
Hello, World!  
  
05:06:51 hello-java  
main [!?] #  
=> sha256:d85151f15b6683b98f21c3827ac545188b1849 5.42MB / 5.42MB 27.5s  
=> extracting sha256:d85151f15b6683b98f21c3827ac545188b1849 27.9s  
=> extracting sha256:db38d58ec8ab4111b072f6700f978a5198 202.07MB / 202.07MB 90.5s  
=> extracting sha256:001c52e26ad57e3b25b439ee0852f6692e5c0f2d5d98 7.7s  
=> extracting sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b 0.8s  
=> extracting sha256:2868746827ec1b043b571e4788693eab7e9b2a953011 0.7s  
=> extracting sha256:9daef329d35093868ef75ac8b7c6eb407fa53abbc3a 8.6s  
=> extracting sha256:d85151f15b6683b98f21c3827ac545188b1849efb14a 0.6s  
=> extracting sha256:66223a710990a0ae7162aedd80417d30303afa3f24aa 0.0s  
=> extracting sha256:db38d58ec8ab4111b072f6700f978a51985acd252aab 4.1s  
=> [internal] load build context 0.0s  
=> transferring context: 37B 0.0s  
=> [2/4] WORKDIR /app 0.5s  
=> [3/4] COPY HelloWorld.java . 0.1s  
=> [4/4] RUN javac HelloWorld.java 1.2s  
=> exporting to image 0.1s  
=> exporting layers 0.1s  
=> writing image sha256:168e370076ae44a97e80a52b283a5922b4e75f76d 0.0s
```