

Daniel Gómez Sánchez

DW1E

Docker - Example: Java application

Create a folder called java_app with a single file Main.java whose content is:

```
public class Main { public static void main(String[] args) {  
  
// Prints "Hello, World" to the terminal window.  
  
System.out.println("Hello, World"); } }
```

```
root@daniel-VirtualBox:/home/daniel# nano main.java cp /var/lib/docker/volumes/java-vol/_data  
root@daniel-VirtualBox:/home/daniel# nano main.java cp main.java /var/lib/docker/volumes/java-vol/_data  
root@daniel-VirtualBox:/home/daniel# nano main.java  
root@daniel-VirtualBox:/home/daniel# ls  
Desktop Documents Downloads main.java Music Pictures Public Templates Videos  
root@daniel-VirtualBox:/home/daniel# cp main.java /var/lib/docker/volumes/java-vol/_data  
root@daniel-VirtualBox:/home/daniel# ls /var/lib/docker/volumes/java-vol/_data  
main.java
```

build the image from the current folder java_app:

```
root@daniel-VirtualBox:/home/daniel# docker volume create java-vol  
java-vol  
root@daniel-VirtualBox:/home/daniel# docker inspect java-vol  
[  
  {  
    "CreatedAt": "2021-12-16T16:41:29+01:00",  
    "Driver": "local",  
    "Labels": {},  
    "Mountpoint": "/var/lib/docker/volumes/java-vol/_data",  
    "Name": "java-vol",  
    "Options": {},  
    "Scope": "local"  
  }  
]
```

And then we will start a container to run the Java file.

```
root@daniel-VirtualBox:/home/daniel# docker run --rm --name my-running-app --mount source=java-vol,destination=/usr/src/myapp openjdk javac /usr/src/myapp/Main.java
```

Using the same volume as in part 2, it is possible to run a container with an interactive bash from an openjdk image

```
root@daniel-VirtualBox:/home/daniel# docker run -it --rm --name my-running-app --mount source=java-vol,destination=/usr/src/myapp openjdk bash  
bash-4.4# cd /usr/src/myapp  
bash-4.4# javac Main.java  
bash-4.4# java Main.java  
Hello, World  
bash-4.4#
```