

```

const int N = 1000000;
int a[N], b[N], c[N], d[N], e[N], f[N], g[N], h[N], i[N], j[N], k[N], l[N], m[N], n[N], o[N], p[N], q[N], r[N], s[N], t[N], u[N], v[N], w[N], x[N], y[N], z[N];
// ... (other code)

```

Flight skills are provided —> before —> before project —> ...

© 2007 The McGraw-Hill Companies. All rights reserved. (Make a note: that HIT Week is already booked for your summer plans.)

```

D:\tools\app\727\Tools\727\IndexEditor
Volume Serial Number is 9627-9627

Directory of D:\tools\app\727\Tools\727\IndexEditor

11/26/2024  10:00 AM    <DIR>          .
11/26/2024  10:00 AM    <DIR>          ..
11/26/2024  10:00 AM    <DIR>          D:\tools\app\
11/26/2024  10:00 AM    <DIR>          D:\tools\
11/26/2024  10:00 AM    <DIR>          D:\tools\app\
5 Files(s)      175,060,345,432 bytes free

```

C:\Documents\par\F5 Tools\00344444\3-DeckerBertapp\www clean package

$$E\{Y_k|k, h, c, s, q, p, \hat{\lambda}^k, \hat{\lambda}^{k-1}, \text{and } \hat{\lambda}^{k-2}\} = E\{Y_k|k, h, c, s, q, p, \text{and } \hat{\lambda}^{k-1}\} = E\{Y_k|k, h, c, s, q, p\} = \text{and } E\{Y_k|k, h, c, s, q, p, \hat{\lambda}^k, \hat{\lambda}^{k-1}, \text{and } \hat{\lambda}^{k-2}\} = E\{Y_k|k, h, c, s, q, p\}.$$
[illegible][illegible]

```
D:\docker\apps\70-Test\700-Env\70docker\test\70docker tag testapp:1.0 myname/testapp:1.0
```

The path refers to repository, <https://github.com/returncode256/404-page-generator>

MAking Spring Boot Standalone App (jar file) as the Docker App

step1) Develop spring boot starter App by taking the packaging type "jar file"

//Main class

```
package com.nt;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
```

```
public class DockerSbApp1Application {
```

```
public static void main(String[] args) {
```

```
SpringApplication.run(DockerSbApp1Application.class, args); System.out.println("welcome to Docker Apps");
```

Dockerfile

```
=====
```

(In the Project directory)

```
FROM openjdk
```

```
WORKDIR /app1
```

```
COPY target/*.jar /app1
```

```
CMD ["java","-jar","BootApp1.jar"]
```

in pom.xml

```
=====
```

```
add <finalName>Boot App1 </finalName> under <build> tag
```

```
}
```

```
}
```

step2) Keep the Project in GIT Repository

Right click on project ----> team --->share project --->

step3) start docker desktop

step4) pull the GIT repository content (Make sure that GIT Bash is already installed in your computer)

```
D:\DockerApps\FS-Tools903\Demo3>git init
```

```
Initialized empty Git repository in D:/DockerApps/FS-Tools903/Demo3/.git/
```

```
D:\DockerApps\FS-Tools903\Demo3>git pull
```

```
https://github.com/natarazworld/NTFSTools-903.git master
```

```
D:\DockerApps\FS-Tools903\Demo3>dir
```

```
Volume in drive D has no label.
```

```
Volume Serial Number is 90EF-90E7
```

```
Directory of D:\DockerApps\FS-Tools903\Demo3
```

```
11/29/2024
```

```
10:00 AM
```

<DIR>

11/29/2024 09:58 AM

<DIR>

11/29/2024 10:00 AM

<DIR>

11/29/2024 10:00 AM

<DIR>

11/29/2024 10:00 AM

<DIR>

© File(s) 5 Dir(s)

DockerBootApplication GITProj01 JenkinsWebApp1

0 bytes

175,866,642,432 bytes free

step5) build the app using maven commands

D:\DockerApps\FS-Tools903\Demo3\DockerBootApplication>mvn clean package

step6) build the docker image

D:\DockerApps\FS-Tools903\Demo3\DockerBootApplication>docker build -t bootappl-img

step7) Build docker container from the docker image (running the app in docker style)

D:\DockerApps\FS-Tools903\Demo3\DockerBootApplication>docker run

bootappl-img

\\

Spring

||||(||||))|||_,|///=_/|_/_/_/

Chat

from Ganesh Ugale to everyone: executed successfully sir

1

To: Everyone

Enter chat message here

:: Spring Boot ::

2024-11-29T04:38:29.960Z INFO 1

(v3.4.0)

[DockerBootApplication] [

main] com.nt. DockerBootApplication

tBootApplication v0.0.1-SNAPSHOT using Java 18.0.2.1 with PID 1 (/app2/BootApplication.jar started by

2024-11-29T04:38:29.963Z INFO 1 [DockerBootApplication] [set, falling back to 1 default

```
profile: "default" 2024-11-29T04:38:30.443Z INFO 1 [DockerBootApplication] [ ApplApplication
in 0.838 seconds (process running for 1.218) Welcome to Docker Spring boot App
main] com.nt. DockerBootApplication main] com.nt. DockerBootApplication
```

step8) push the image to docker hub (docker registry)

```
D:\DockerApps\FS-Tools903\Demo3\DockerBootApplication>docker tag boot appl-img
natarazdocker/fs903repo: bootappl-tag
```

username

repo name

tag name

```
D:\DockerApps\FS-Tools903\Demo3\DockerBootApplication>docker push
natarazdocker/fs903repo:bootappl-tag The push refers to repository
[docker.io/natarazdocker/fs903repo]
```

Proecedure make Java App web application as the docker Image / docker App

=====

=====

=====

step1) develop the web application and test it locally

Docker-TestWebApp

> Deployment Descriptor: Archetype Created Web App

>

Java Resources

Deployed Resources

src

main

java

resources

webapp

>

WEB-INF

index.html

test

target

> > Worskpaces

pom.xml

step2) Add Dockerfile to the Project

index.html

=====

<h1 style="color:red;text-align:center"> Welcome to Docker WEB application

In pom.xml (change the package name)

```
=====
<finalName>Docker-TestWebApp</finalName>
```

```
: Star
```

```
</h1>
```

DockerFile

```
=====
```

FROM tomcat:10-jdk17

COPY target/Docker-TestWebApp .war EXPOSE 8080

/usr/local/tomcat/webapps/Docker-TestWebApp.war

(copying war file to Docker Container Tomcat server)

note:: u can create war file for the Project in eclipse itself

to

Right click on the Project ---> run as --> maven build .. ---> goals: mvn clean package

step3) Commit the Project Git hub

step4) pull the Project from GIT

```
D:\DockerApps\FS-Tools903\Demo3\Docker-TestWebApp>git pull
```

```
https://github.com/natarazworld/NTFSTools-903.git master
```

step5) build the Project using maven tool

```
D:\DockerApps\FS-Tools903\Demo3\Docker-TestWebApp>mvn clean package
```

(if we create this war file in Eclipse IDE itself then doing this work here not required)

step6) create the Docker image

```
D:\DockerApps\FS-Tools903\Demo3\Docker-TestWebApp>docker build -t webappl-img .|
```

step7) create docker container performing the port mapping

windows

Docker

Os pott Tomcat port

```
D:\DockerApps\FS-Tools903\Demo3\Docker-TestWebApp>docker run -p 6677:8080 webappl-img
```

step9) test the application

http://localhost:6677/Docker-TestWebApp/

in

What is port mapping docker container creation?

port mapping

Ans) The softwares of Docker containers might be running on some default port number like 8080 for tomcat.. but the same 8080 port number might be busy for other services in the HOST computer, So we need to map docker container software service port number to a free port number of the HOST computer This process is called port mapping

Host computer

Docker container

Tomcat server (running on port:: 8080)

Docker-Test WebApp

Docker engine

port mapping

Host OS (Linux/windows)host port :: 6677

Docker FAQs

=====

Q) What is Containerization?

Q) Why do we need Containerization? (Code + env.. packing)

Q) List out various Containerization tools?

Q) Explain various comps of Docker tool?

a) Docker Client b) Docker Daemon c) Docker Image d) Docker Container e) DockerHub

Q) Is Docker S/w required in every machine for achieving and code and env portability?

Q) What is the role Dockerfile in Docker image creation?

Q) What is the difference b/w Docker image and Docker container?

Q) What is need of port mapping in Docker Container creation?

Q) List out few important readily available images of Docker HUB?

Q) What is role of Docker tool as part of build and deployment automation?

<http://localhost:8080/Docker-TestWebApp> (error)

[http://localhost:66 77/Docker-TestwebApp](http://localhost:6677/Docker-TestwebApp) (success)