

1) Download Docker Desktop latest **Deb** package

2) To Install Docker Engine

- Uninstall Old versions
`sudo apt-get remove docker docker-engine docker.io containerd runc`
- SetUp Repository

```
sudo apt-get update
```

```
sudo apt-get install \
```

```
ca-certificates \
```

```
curl \
```

```
gnupg \
```

```
lsb-release
```

- Add Docker's official GPG key:

```
sudo mkdir -p /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg
```

```
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
```

```
https://download.docker.com/linux/ubuntu \
```

```
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- Install Docker Engine

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

- Verify that Docker Engine is installed correctly by running the **hello-world** image.

```
sudo service docker start
```

```
sudo docker run hello-world
```

3) Open Docker Desktop and make sure its running

4) open SpringBoot Application in IntelliJ

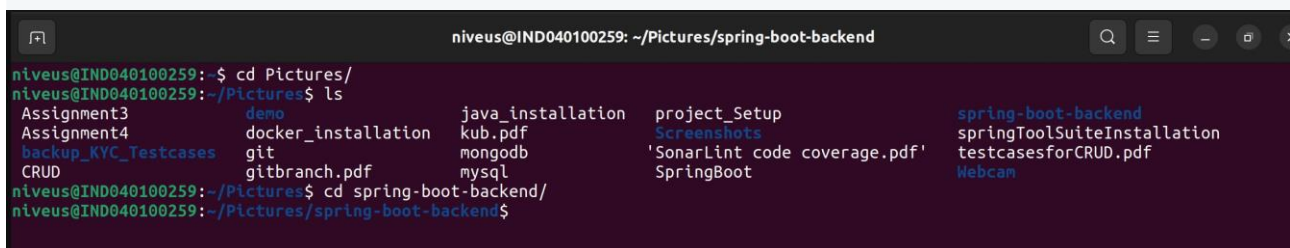
5) Maven -> Define Goal -> Maven Package

To convert project to jar file

6) Create a Dockerfile

```
FROM openjdk:17
LABEL maintainer="shivakarthika"
ADD target/spring-boot-backend-0.0.1-SNAPSHOT.jar springboot-docker-demo.jar
ENTRYPOINT ["java", "-jar", "springboot-docker-demo.jar"]
```

7) Open terminal and go to project directory



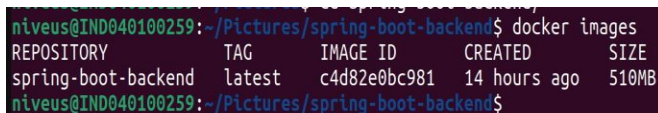
A terminal window titled 'niveus@IND040100259: ~/Pictures/spring-boot-backend'. The user runs 'cd Pictures/' and then 'ls'. The output shows a directory listing with files like Assignment3, Assignment4, backup_KYC_Testcases, CRUD, demo, docker_installation, git, gitbranch.pdf, java_installation, kub.pdf, mongodb, mysql, project_Setup, Screenshots, 'SonarLint code coverage.pdf', SpringBoot, spring-boot-backend, springToolSuiteInstallation, testcasesforCRUD.pdf, and Webcam. The user then runs 'cd spring-boot-backend/' and the prompt changes to '~/Pictures/spring-boot-backend\$'.

8) To Build Docker Image

docker build -t spring-boot-backend:latest .

9) To check if docker images are created or not

docker images



A terminal window showing the output of the 'docker images' command. The output is a table with columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. The table shows one entry: 'spring-boot-backend latest c4d82e0bc981 14 hours ago 510MB'. The prompt is '~/Pictures/spring-boot-backend\$'.

10) Run docker image in docker container

docker run -p 8080:8080 spring-boot-backend