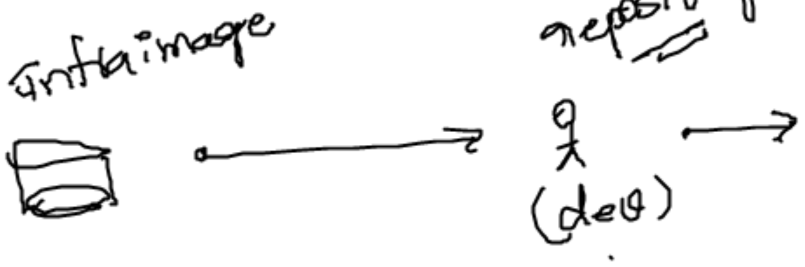
Lny

```
# git clone repository
repository # docker build -t "imagename" .
```

PaC

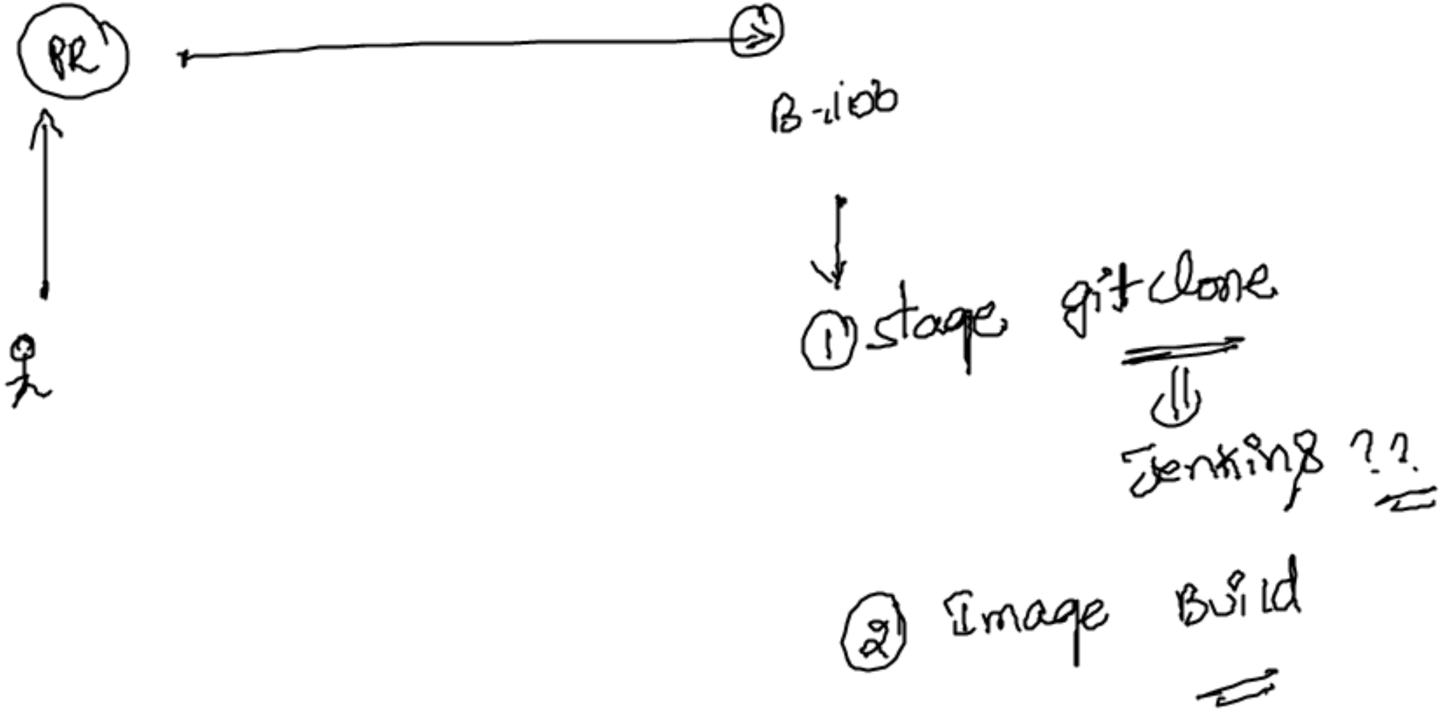
CICD on dockers



Dockerfile

```
FROM infra
WORKDIR /opt/app
EXPOSE 8080
COPY . /opt/app/
RUN mvn clean install \
   && cp ./target/*.jar
          /opt/app/app.jar

CMD [java -jar /opt/app]
```



PR

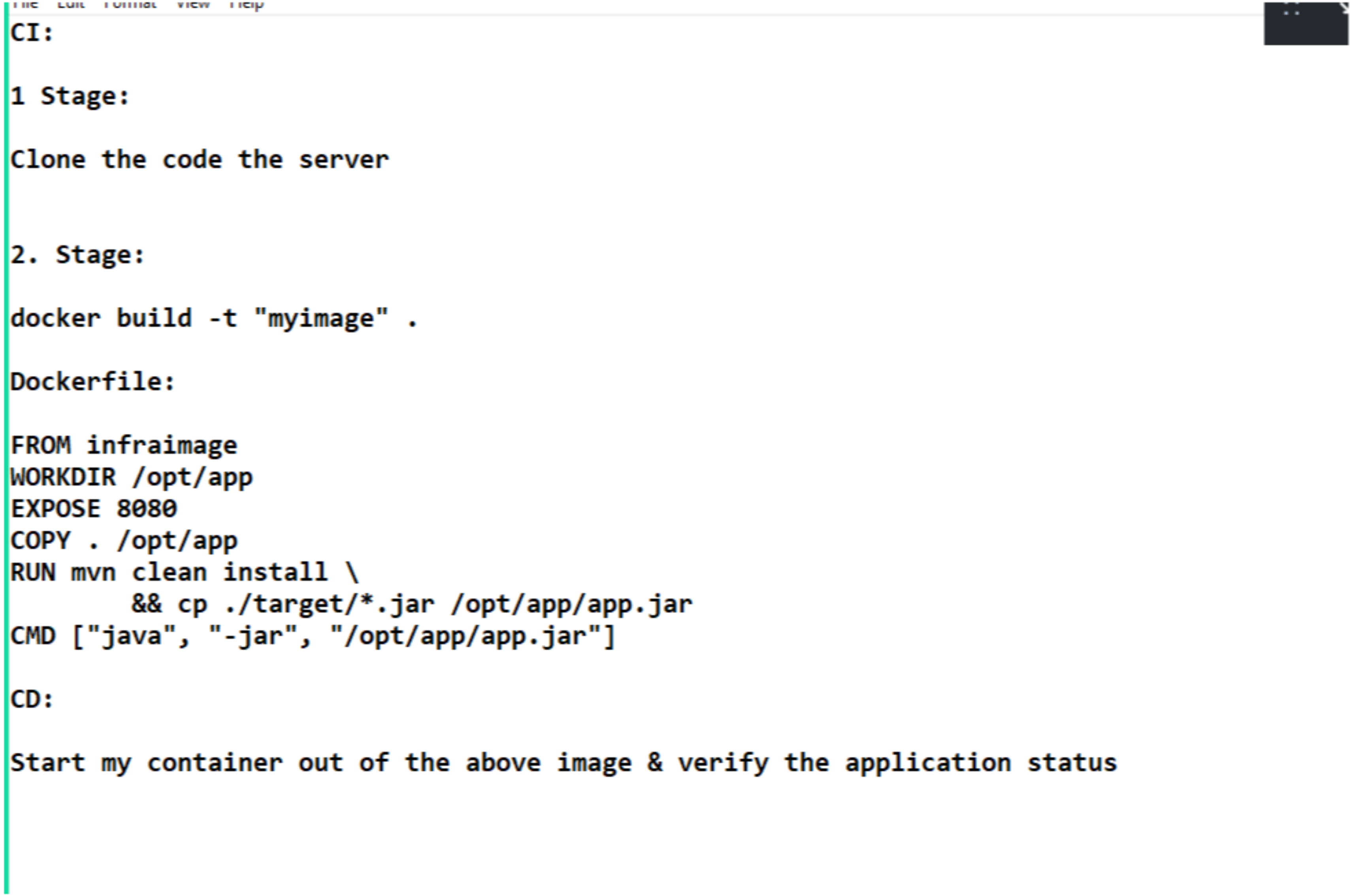
B-job

① stage git clone

⇓

Jenkins ??

② Image Build



CI:

1 Stage:

Clone the code the server

2. Stage:

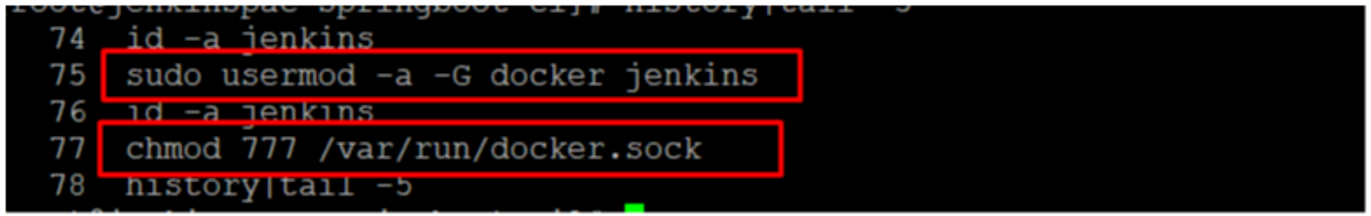
```
docker build -t "myimage" .
```

Dockerfile:

```
FROM infraimage
WORKDIR /opt/app
EXPOSE 8080
COPY . /opt/app
RUN mvn clean install \
        && cp ./target/*.jar /opt/app/app.jar
CMD ["java", "-jar", "/opt/app/app.jar"]
```

CD:

Start my container out of the above image & verify the application status



```
root@jenkinspac springboot-ci]# history|tail -5
   74  id -a jenkins
   75  sudo usermod -a -G docker jenkins
   76  id -a jenkins
   77  chmod 777 /var/run/docker.sock
   78  history|tail -5
```

nodejs

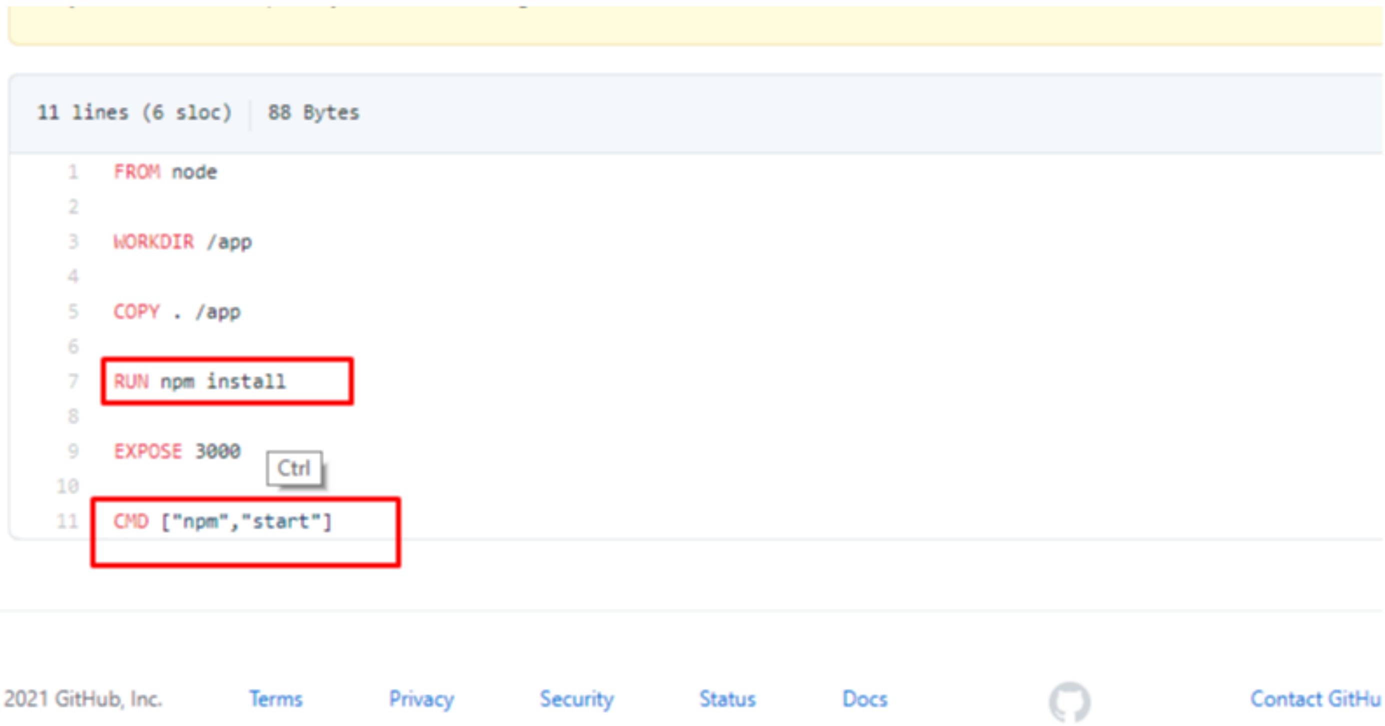
CI → ① clone

② create image

⊙ pom.xml → java

⊙ package.json → nodejs|reactjs

Nodejs Dockerfile



11 lines (6 sloc) | 88 Bytes

```
FROM node

WORKDIR /app

COPY . /app

RUN npm install

EXPOSE 3000

CMD ["npm","start"]
```