

Jenkins - Pal

Loops :- while, for

while :-

while (Cond.) {  
do something

}  
=

for:-

↓  
for ( i=1; i ≤ 10; i++) {

do something

}

for ( i in list ) {

do something

}

Java, C, C++

functions :-

Python

def myfn():

|||||

myfn()

PaC

def myfn() {  
    |||  
}

pipeline {

agent any

stages {

stage {

steps {

script {

}}}}}} myfn()

```
untitled
11 def myfn() {
12   println "Hi All welcome to Dvs Devops"
13 }
8 pipeline {
7   agent any
6   stages {
5     stage('Working with Loops') {
4       steps {
3         script {
2           myfn()
1         }
12      }
1   }
2 }
3 }
```

```
11 def myfn(int a, int b) {  
10   result = a + b  
9   println "add of ${a} and ${b} is ${result}"  
8 }  
7 pipeline {  
6   agent any  
5   stages {  
4     stage('Working with Loops') {  
3       steps {  
2         script {  
1           myfn(10,20)  
12          myfn(100,200)  
1         }  
2       }  
3     }  
4   }  
5 }
```

```
def myfn(int a=10, int b=100) {  
    result = a + b  
    println "add of ${a} and ${b} is ${result}"  
}
```

```
pipeline {  
    agent any  
    stages {  
        stage('Working with Loops') {  
            steps {  
                script {  
                    myfn()  
                    myfn(200)  
                }  
            }  
        }  
    }  
}
```

Ctrl

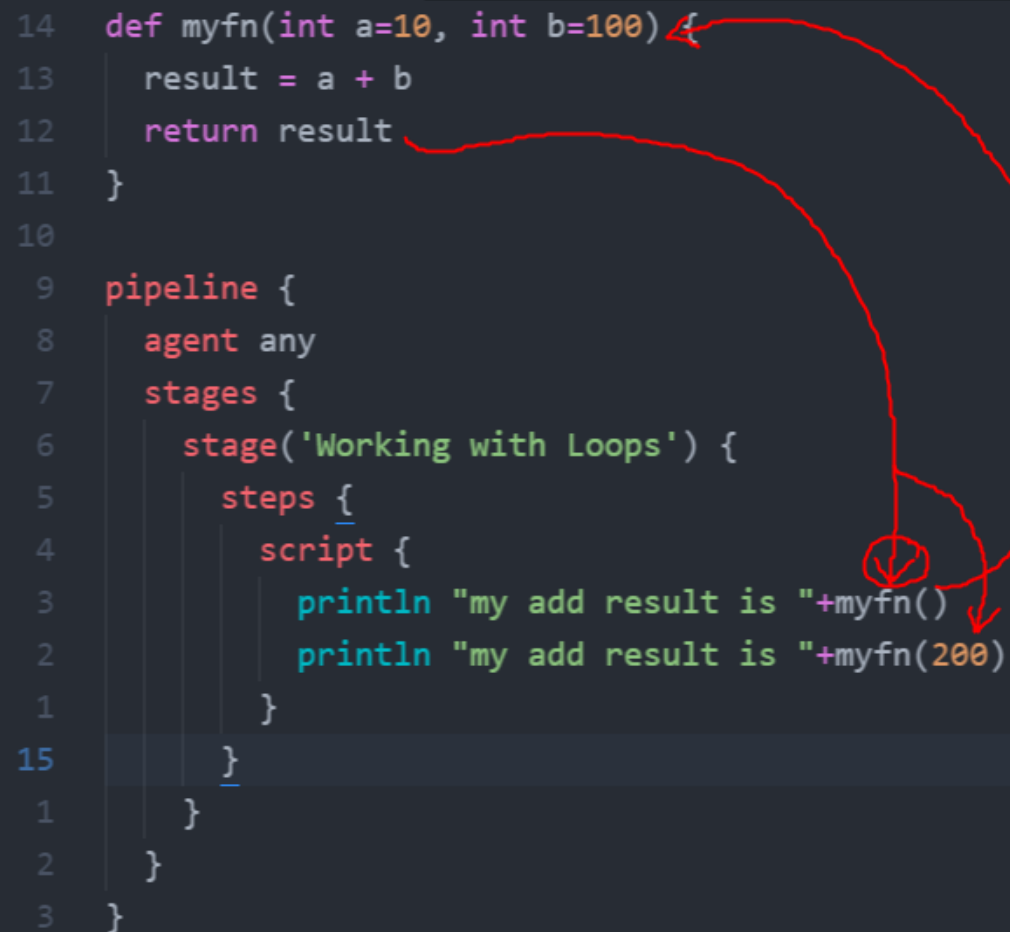
→

10

100

10 + 100 = 110 ✓  
200 + 100 = 300  
300

```
14 def myfn(int a=10, int b=100) {  
13     result = a + b  
12     return result  
11 }  
10  
9 pipeline {  
8     agent any  
7     stages {  
6         stage('Working with Loops') {  
5             steps {  
4                 script {  
3                     println "my add result is "+myfn()  
2                     println "my add result is "+myfn(200)  
1                 }  
15             }  
1         }  
2     }  
3 }
```



## File operations :-

Python:-  $obj = \text{open}(\text{"filename"}, \text{mode})$

(r, w, a)  
↑  
mode

Pac :- ~~#~~ file  $obj = \text{new file}(\text{"filepath filename"})$

readlines

$obj.readlines()$

$obj.\underline{\text{writelines}}()$



```
3 pipeline {  
2   agent any  
1   stages {  
0     stage('Working with Loops') {  
9       steps {  
8         script {  
7           File obj1 = new File("/tmp/dvsbatch1.txt")  
6           obj1.write("Hi Team welcome to Dvs")  
5           println "file content is ${obj1.text}"  
4         }  
3       }  
2     }  
1   }  
4 }  
1
```

```
untitled
11 pipeline {
10   agent any
9   stages {
8     stage('Working with Loops') {
7       steps {
6         script {
5           File obj1 = new File("/tmp/dvsbatch1.txt")
4           println "my file content is ${obj1.text}"
3           for(line in obj1.readlines()){
2             println "my line is ${line}"
1           }
12        }
1       }
2     }
3   }
4 }
5
```

File creation

file myobj = new File("myfilename")

creating

↓  
myobj.write("some content")

reading

myobj.readLines() ⇒ [line<sub>1</sub>, line<sub>2</sub>]

myobj.text ⇒ content of file

append

myobj.append("some new data")



```
pipeline {  
  agent any  
  stages {  
    stage('Working with Loops') {  
      steps {  
        script {  
          File myobj1 = new File("/tmp/batch1.txt")  
          myobj1.write("Hi I am Line1")  
          myobj1.append("Hi I am Line2")  
          println myobj1.text  
          println myobj1.readLines()  
        }  
      }  
    }  
  }  
}
```

Ctrl

↗ Springboot (Java) ↓ 200-1.  
Java, maven

# <sup>8</sup> class - class \* .jar filename \* ⊙

⇒ pom.xml → Java code

⇒ mvn clean install

80%.

⊗ Java ⊗

Java

maven (Build tool)

mvn clean install

java -jar jarfilename

100%.



nodejs / reactjs

14/23

nodejs

npm (build)

(node package manager)

npm build

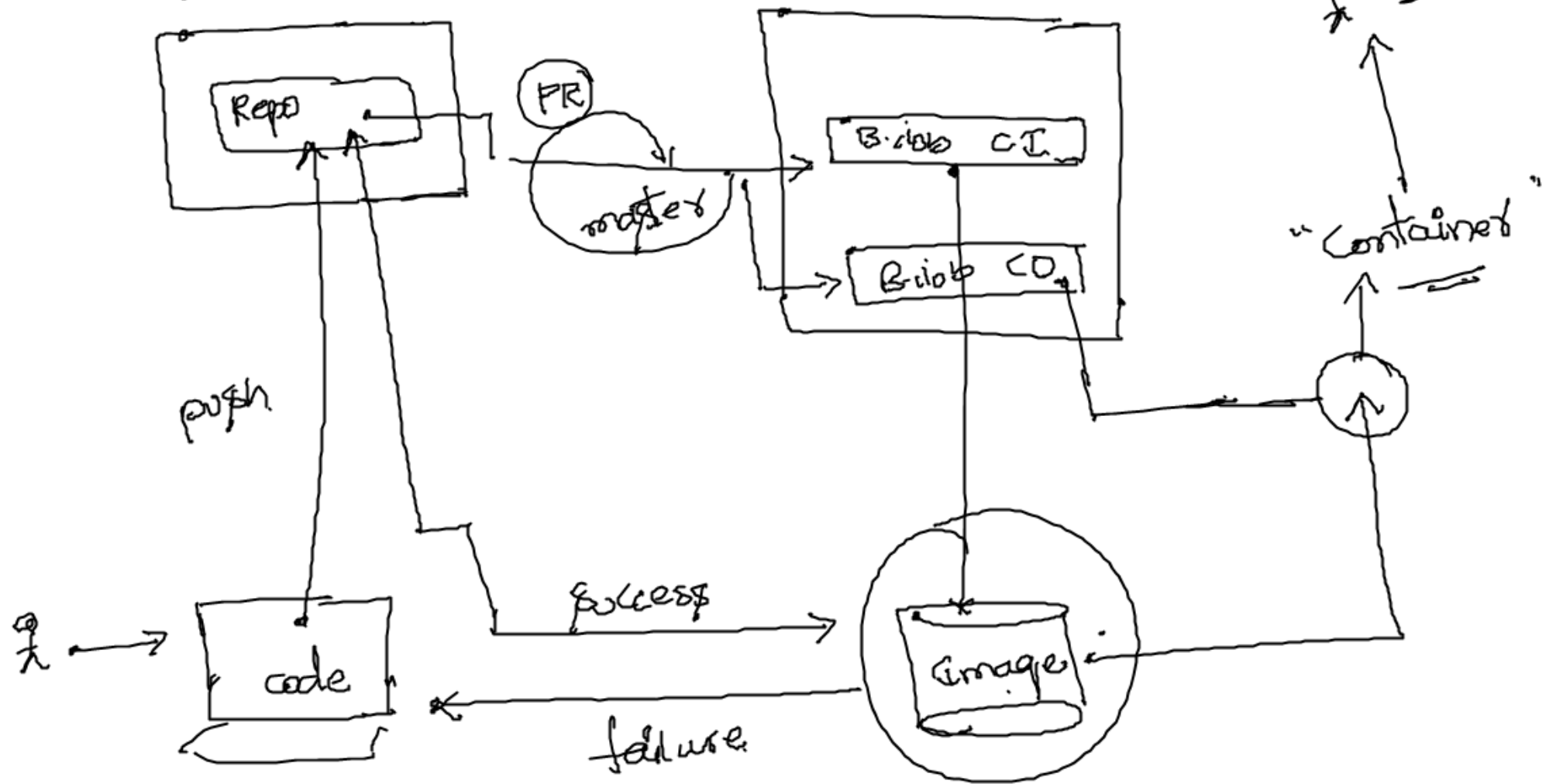
npm install

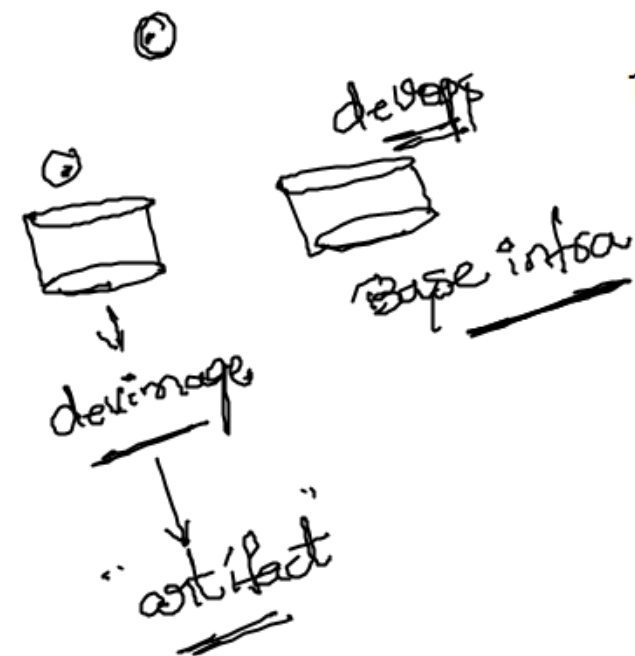
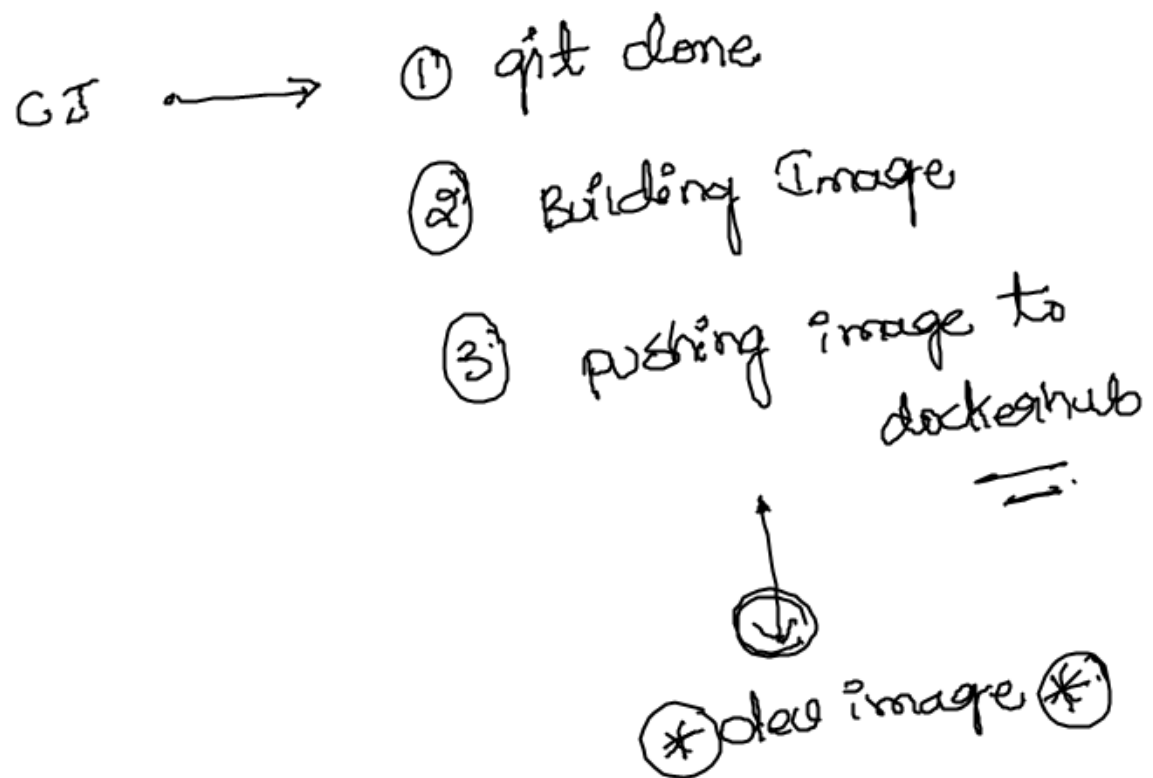
npm start



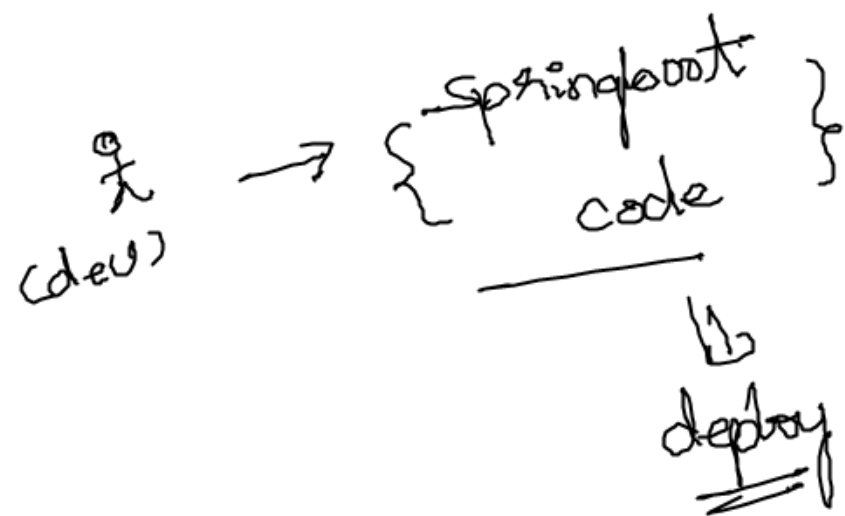
Springboot

GitHub









0. FROM intaiimage

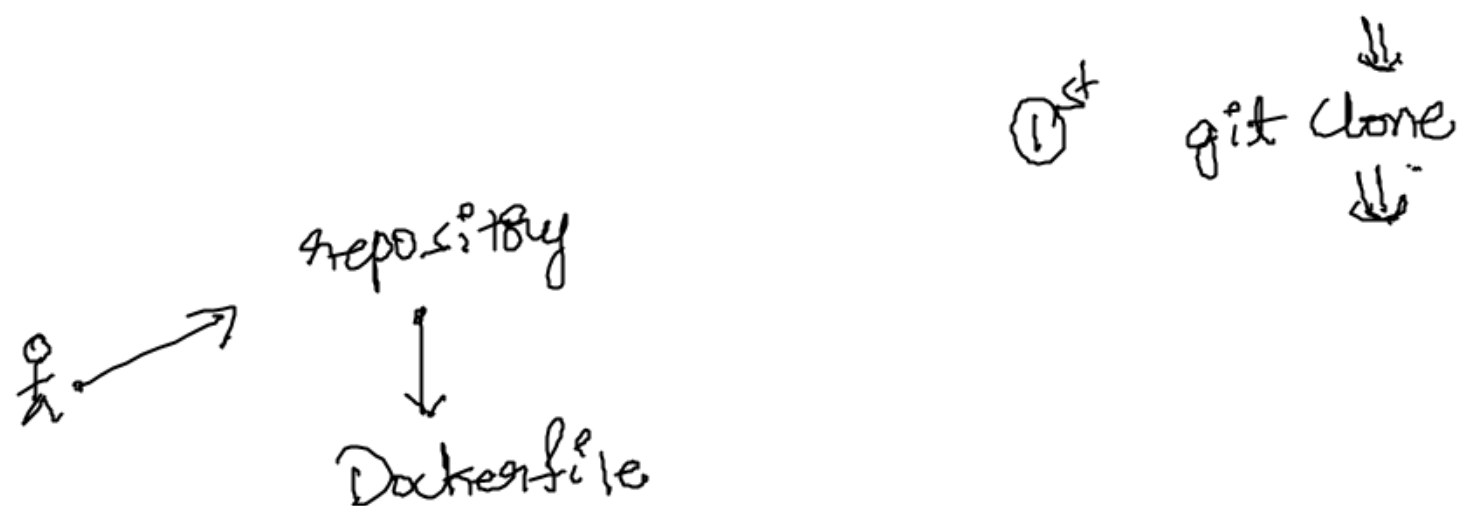
17/23

1. git clone


2. cd repo

3. mvn clean install

4. Image out of it



FROM Entrypoint

COPY  opt/mycode

WORKDIR opt/mycode

RUN rm -rf /var/lib/apt/lists/\* & install -y

CMD cp -r /target/\* . & ./app.jar

one time →

FROM CentOS: 7

RUN yum install java maven

intra image

intra image ↓

FROM maven

WORKDIR /opt/app

EXPOSE 8080

~~COPY~~

~~./opt/app~~ git clone url ⊗

RUN mvn clean package && cp ./target/\*.jar /opt/app/app.jar

ENTRYPOINT ["java", "-jar", "app.jar"]

100000

✓ 1st stage

git clone

\$workspace

⇒ cd /var/lib/docker/workspaces/B40brnme!..

appcode }  
Dockerfile

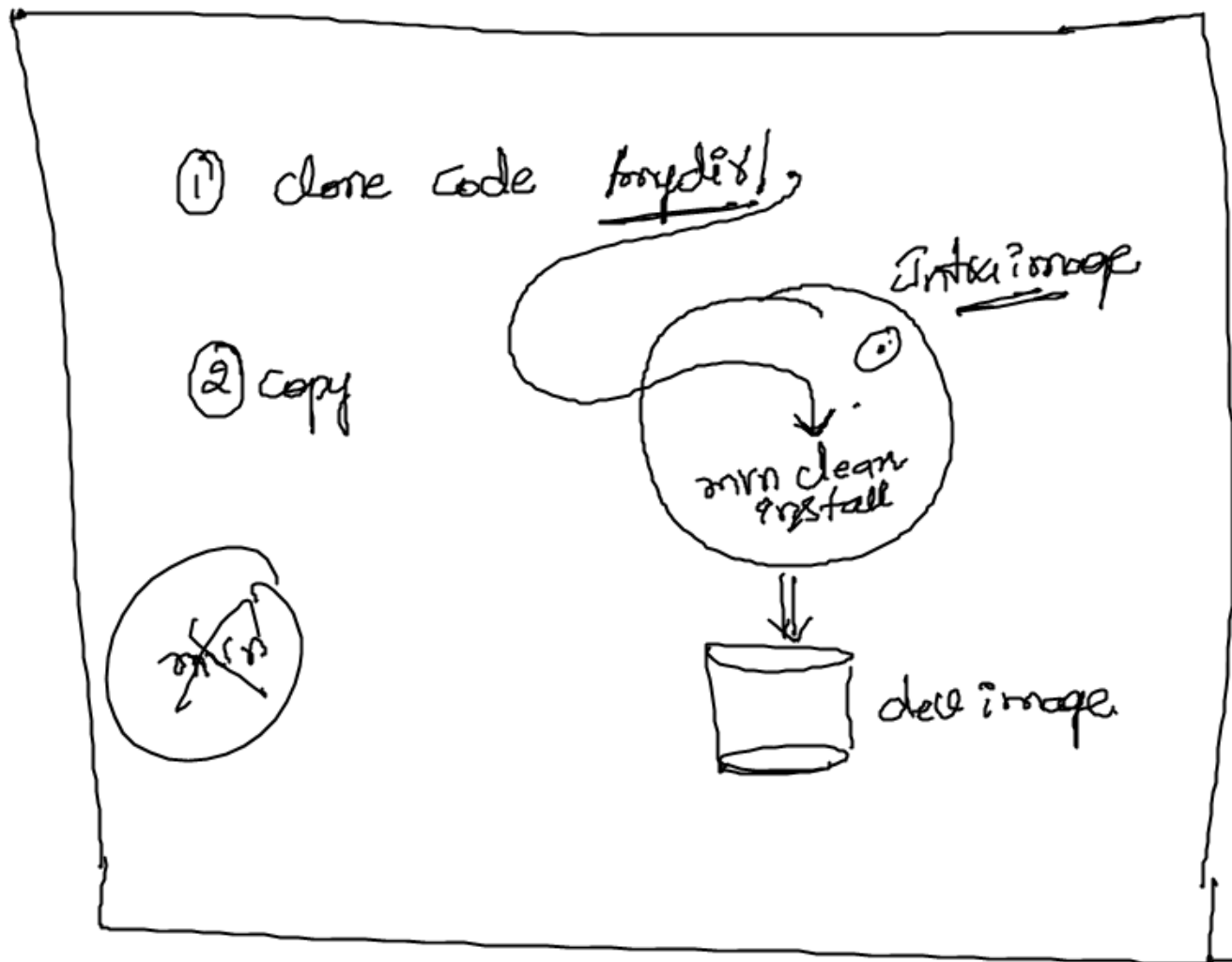
run the image

✓ 2nd stage

Build Images

Server

21/23



3 Build jobs

CI

CD

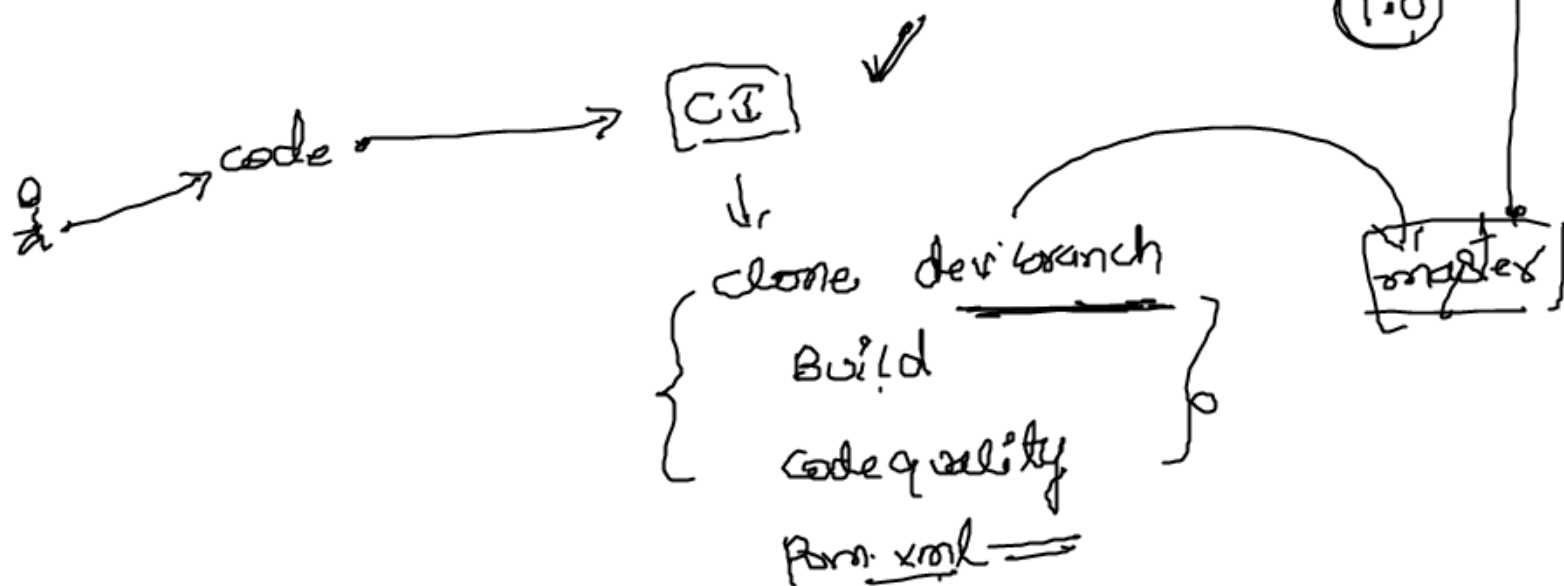
promotion

set tag : latest, 1.0

versioning{ image (81)

artifact

1.0



params

9

1.0

dev

sil

1.6

uat

1.5

pt

prod

1.4

parameter

dev  
sil

parameter

[1.0]

#

argide-playbook

-i

argidenostr web.yml

-l params.env

-e version params.version

vars

[dev]

[sil]

[prod]

23/23