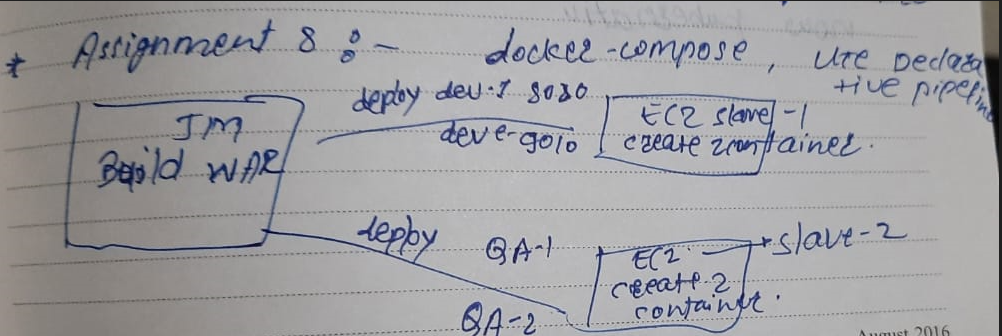
**Assignment -8**

Clone and build game of life in master, then create slave-1 - create 2 container and deploy app on 8080 & 9090 port and also create slave-1 - create 2 container and deploy app on 8080 & 9090 port.

Use docker-compose for calling multiple Dockerfiles



**Step -1 :** configure Jenkins master –

1. Yum update -y
2. Install java 8.
3. Install Apache tomcat, start tomcat – deploy Jenkins in tomcat webapps.
4. Install maven. Set environment variable.

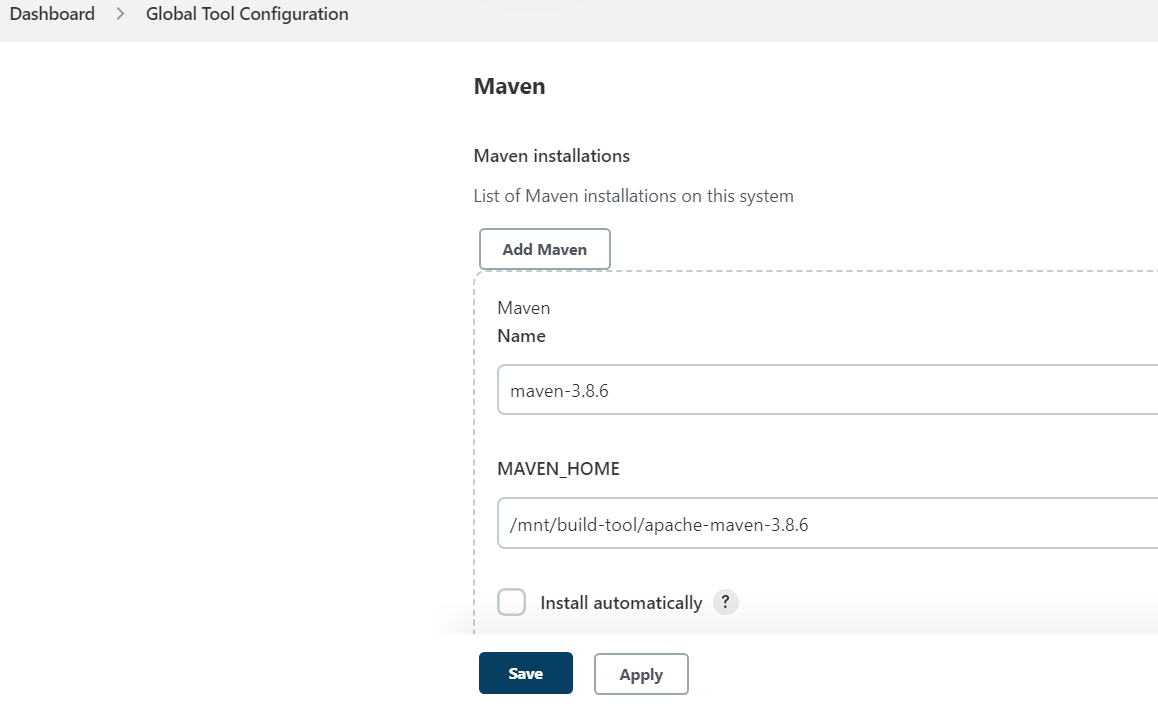
configure Jenkins slave-1 and slave-2 ec2 :-

1. Yum update -y
2. Install java 8.
3. Yum install docker -y
4. Systemctl start docker
5. Install docker compose :- use below commands

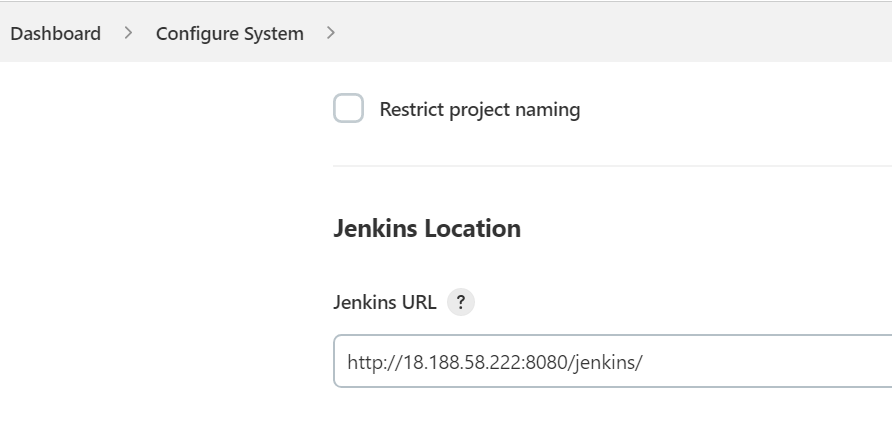
* sudo curl -L https://github.com/docker/compose/releases/download/1.21.0/docker-compose-`uname -s`-`uname -m` | sudo tee /usr/local/bin/docker-compose > /dev/null
* sudo chmod +x /usr/local/bin/docker-compose
* ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
* docker-compose --version

**Step-2 :** Jenkins console :

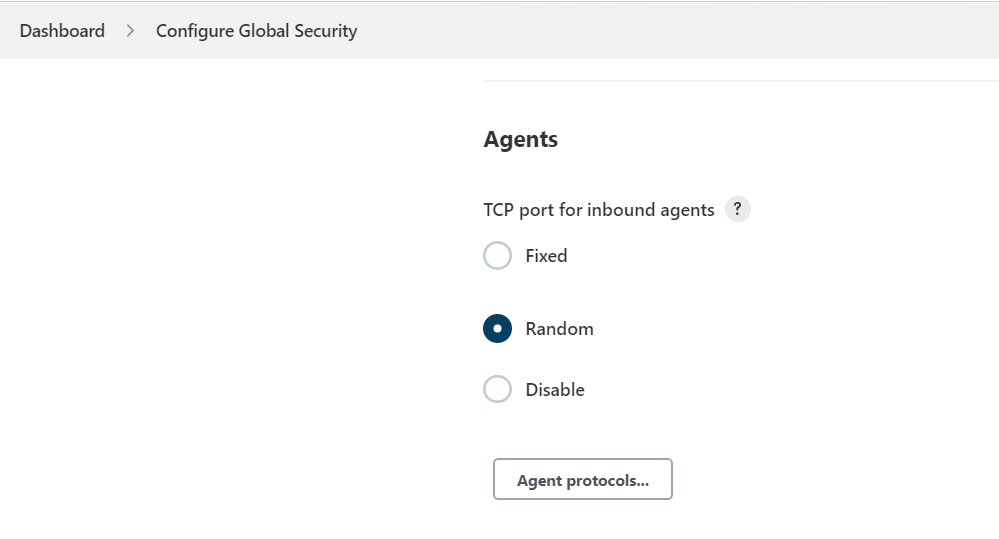
1. Setup maven tool in global tool configuration.



1. Set url of Jenkins:

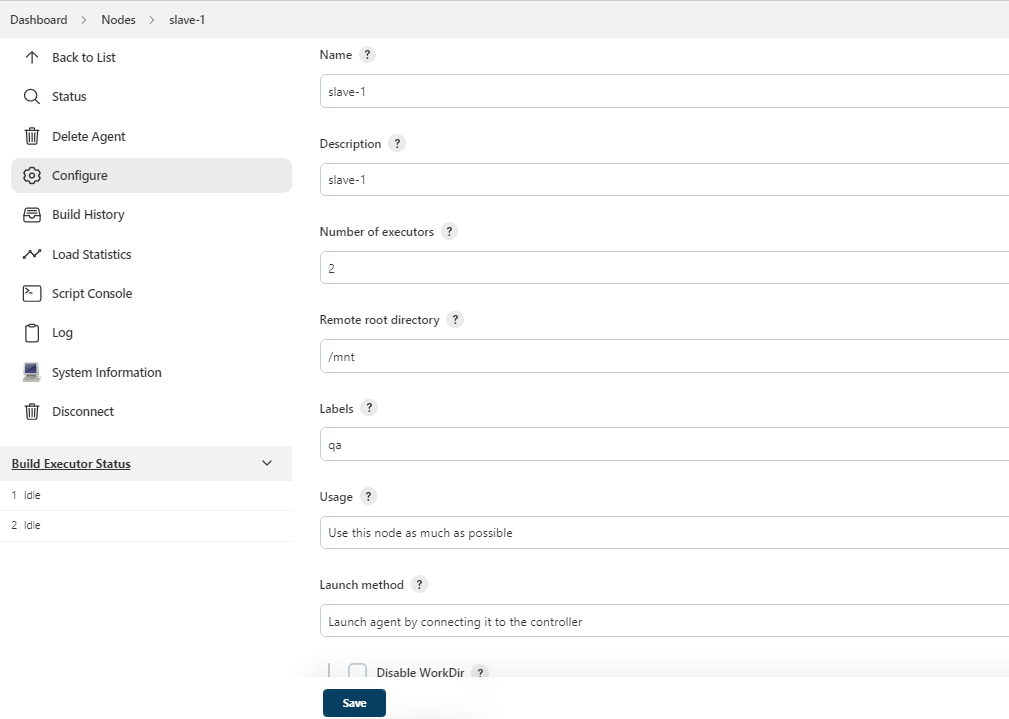


1. Select random port :

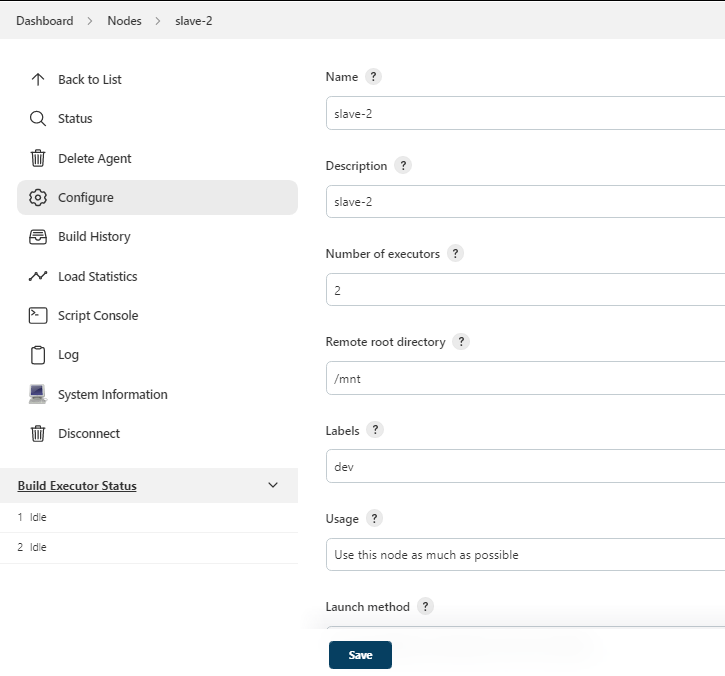


**Step-3 :** create master slave configuration using JNLP

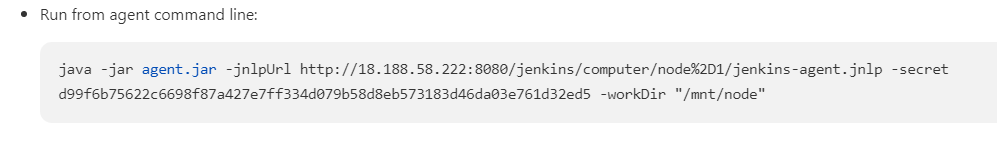
1. create slave-1 in Jenkins console. (by JNPL)



1. create slave-2 in Jenkins console. (by JNPL)



1. Connect these slaves to master by running command in slave ec2 .



Copy agent.jar and wget in slave ec-2

Run above command : nohup “ command ” & ------- for continuous connection

Check it by tail -100f nohup.out. or in custom dir files created -remoting.jar, nohup.out etc

Step-4 : Create docker file and docker-compose in slave-1 and slave-2 :

In slave-1 ec2 (same for slave-2 ec2) create one directory compose in that create 2 directories file-1 ,file2 and docker-compose.yaml

in file1 vi index, vi Dockerfile

in file2 vi index, vi Dockerfile.

Vi Dockerfile

FROM tomcat:9.0

MAINTAINER abhay

CMD ["catalina.sh", "run"]

COPY gameoflife.war /usr/local/tomcat/webapps

EXPOSE 8080

Vi compose-docker.yaml

version: '3'

services:

one:

build:

context: ./file-1

image: server-1

ports:

- "8080:8080"

two:

build:

context: ./file-2

image: server-2

ports:

- "9090:8080"

**Step-5:** write pipeline for job

pipeline {

agent {

node {

label "master"

customWorkspace "/mnt/project"

}

}

stages {

stage ('clone repo') {

steps {

sh "rm -rf game-of-life\*"

sh "git clone https://github.com/Abhay-Dubal/game-of-life.git"

}

}

stage ('create build') {

steps {

dir ('/mnt/project/game-of-life') {

sh "mvn clean install"

}

}

}

stage ('copy on slave') {

steps {

sh "chmod -R 777 /mnt/project/game-of-life/gameoflife-web/target/gameoflife.war"

sh "scp -i linserver.pem /mnt/project/game-of-life/gameoflife-web/target/gameoflife.war ec2-user@172.31.47.125:/mnt"

sh "scp -i linserver.pem /mnt/project/game-of-life/gameoflife-web/target/gameoflife.war ec2-user@172.31.39.87:/mnt"

}

}

stage ('copy app slave-1'){

agent {

node {

label "qa"

customWorkspace "/mnt"

}

}

steps {

sh "sudo cp gameoflife.war /mnt/compose/file-1"

sh "sudo cp gameoflife.war /mnt/compose/file-2"

}

}

stage ('deploy app using docker compose slave-1'){

agent {

node {

label "qa"

customWorkspace "/mnt/compose"

}

}

steps {

sh "sudo docker-compose up -d"

}

}

stage ('copy app slave-2'){

agent {

node {

label "dev"

customWorkspace "/mnt"

}

}

steps {

sh "sudo cp gameoflife.war /mnt/compose/file-1"

sh "sudo cp gameoflife.war /mnt/compose/file-2"

}

}

stage ('deploy app using docker compose slave-2'){

agent {

node {

label "dev"

customWorkspace "/mnt/compose"

}

}

steps {

sh "sudo docker-compose up -d"

}

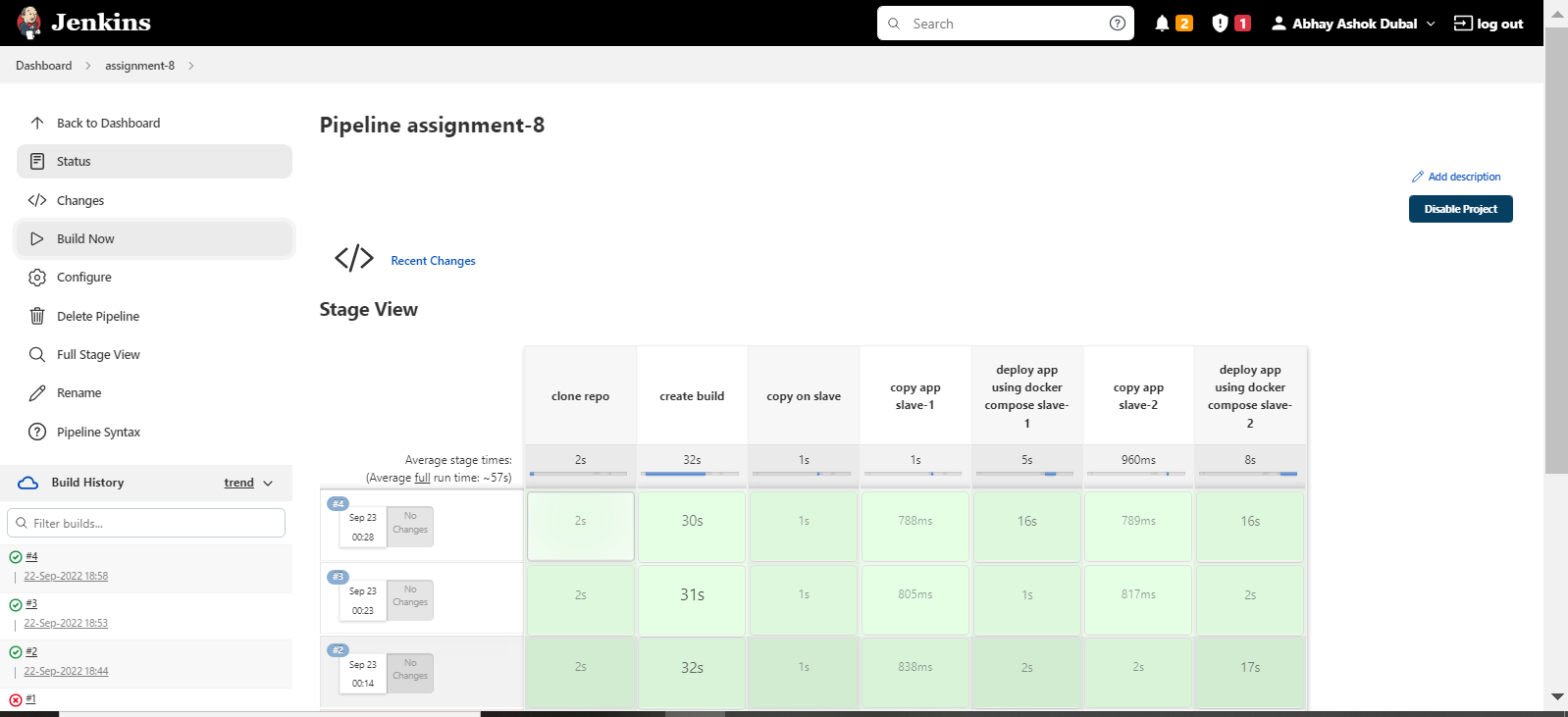
}

}

}

**Step-6 :** befor trigger build if in pipeline you use scp command then first check by manually transfer file through scp (run command at master ec2).

**Step-7 :** trigger build

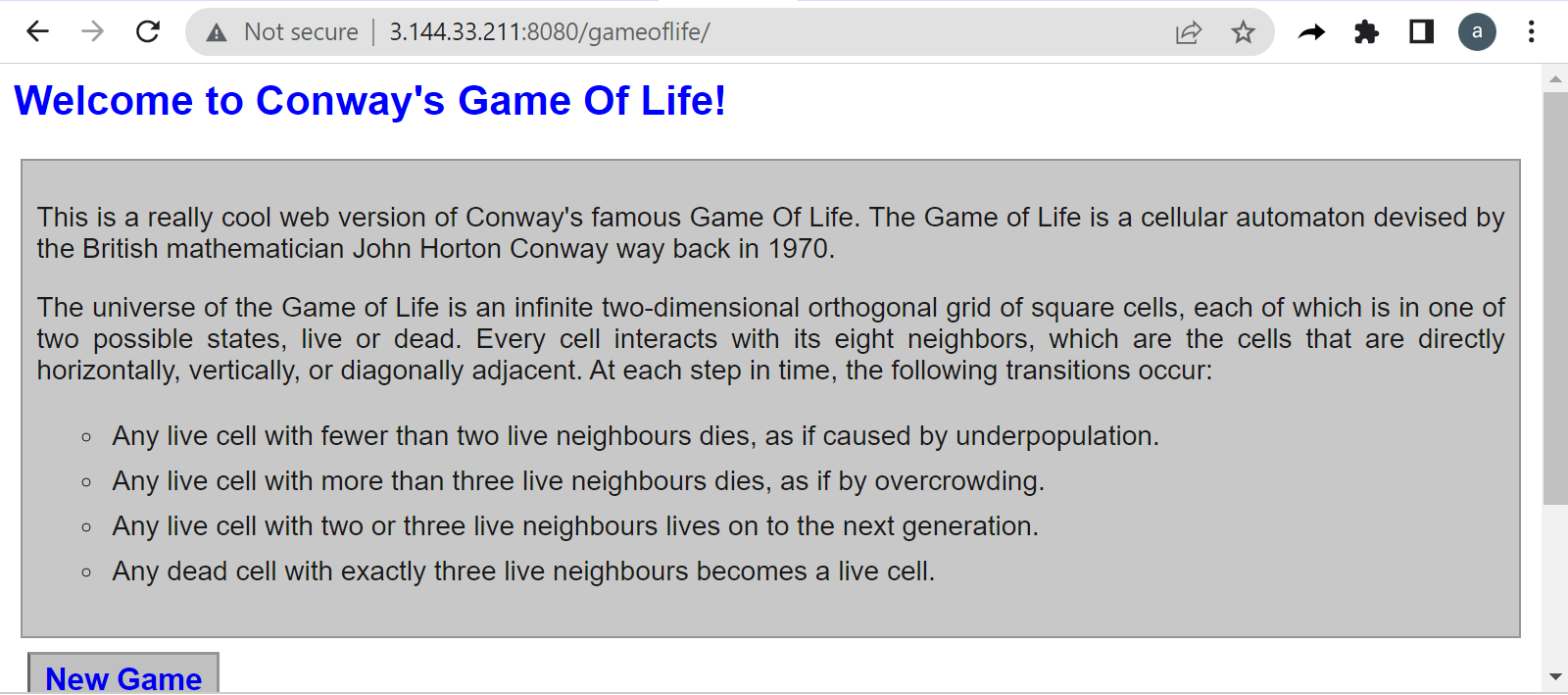


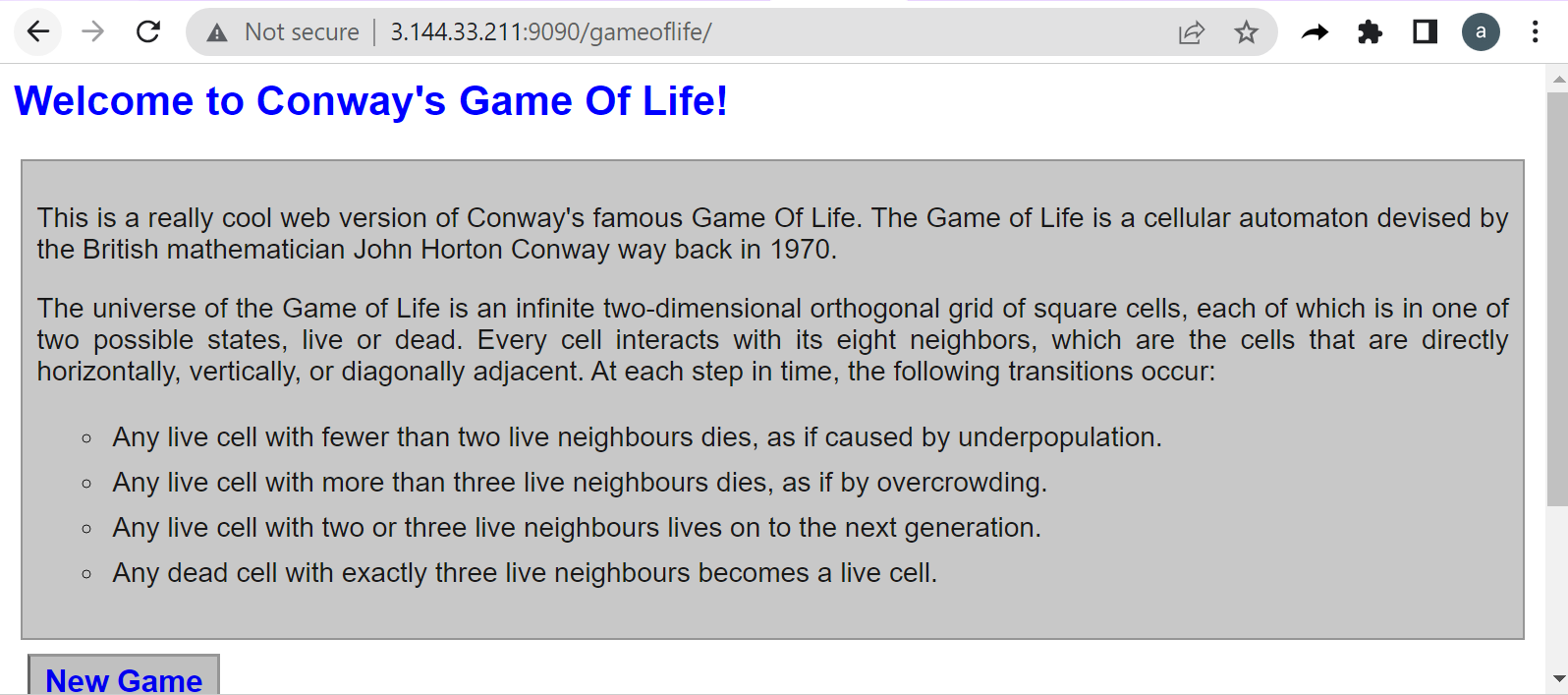
**Step-8 :** check output –

enter slave-1 public ip in browser :port that u bind in container and war file name.

Ex. http://3.144.33.211:8080/gameoflife/

**Slave -2 output :**





**Slave -1 output :**

