

## ASSIGNMENT

**Problem Statement 1:** Deploy **index.html** file on HTTPD server of Jenkins-slave machine

**Problem Statement 2:** Deploy **game-of-life.war** on Tomcat Apache server of Jenkins-slave machine

### Solution:

1. Launched an EC2 instance and configured the Jenkins-slave.
2. Downloaded the **agent.jar** file inside the Jenkins-slave Workspace as per below image.



```
[root@ip-172-31-1-36 jenkins-slave]# ls -ltr
total 1492
-rw-r--r-- 1 root root 1522481 Aug 10 2022 agent.jar
drwxr-xr-x 4 root root      34 Apr 28 04:36 remoting
-rw----- 1 root root    1970 Apr 28 04:37 nohup.out
```

3. Established the Jenkins Master and Slave connections by executing the following command on Linux CLI of Jenkins-slave server as per below image.

```
^C[root@ip-172-31-1-36 jenkins-slave]# nohup java -jar agent.jar -jnlpUrl http://3.109.182.179:8080/jenkins/computer/slave%2D2/jenkins-agent.jnlp -secret 7a098fef04e4c44d947b667086c4ca572e0af8241439c267a9823465821826ca -workDir "/mnt/jenkins-slave" &
[1] 7656
[root@ip-172-31-1-36 jenkins-slave]# nohup: ignoring input and appending output to 'nohup.out'
```

4. Repeated the above steps and configured second Jenkins-slave. (Please refer below image)

Q Search










 4 2

Chitrang Sawant

log out

Manage nodes and clouds

Refresh status

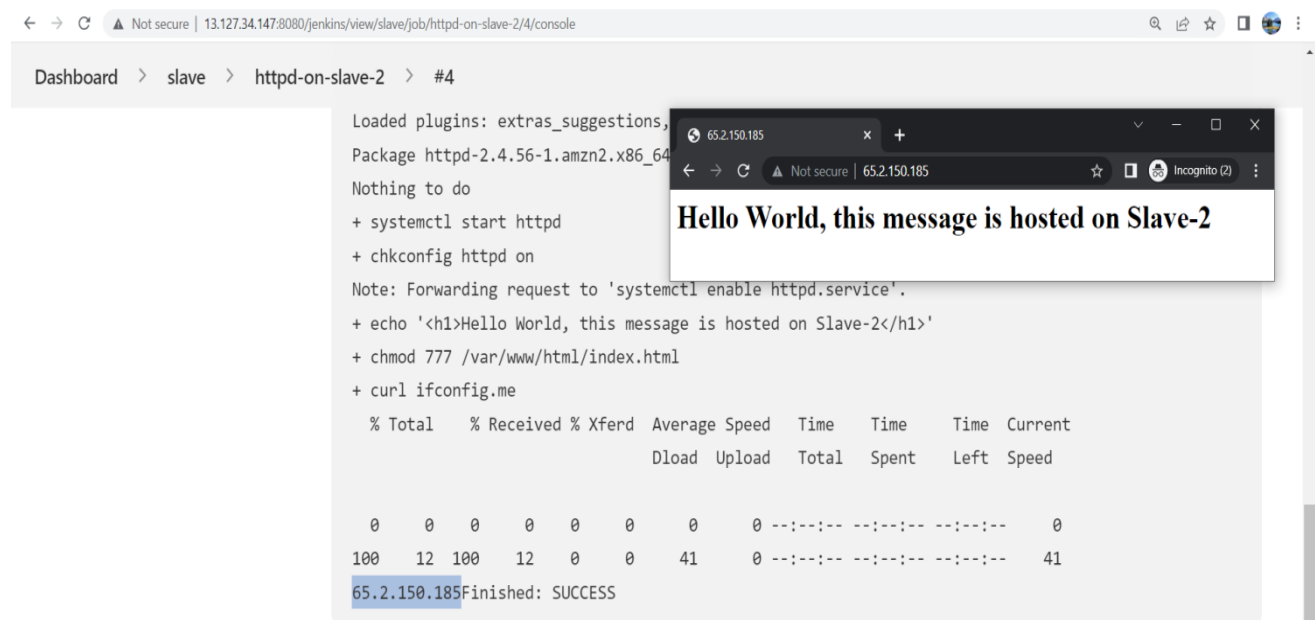
S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.07 GB	 0 B	5.07 GB	0ms 
	slave-1	Linux (amd64)	In sync	6.14 GB	 0 B	6.14 GB	60ms 
	slave-2	Linux (amd64)	In sync	6.14 GB	 0 B	6.14 GB	117ms 
Data obtained		9.8 sec	9.8 sec	9.7 sec	9.7 sec	9.7 sec	9.7 sec

5. Installed java on both the slave machines using the command

**yum install java-1.8.0-openjdk-devel.x86\_64 -y**

6. Installed Apache Httpd server on slave-2 directly during performing BUILD of the Project using the **Build Actions** in Execute Shell. The following commands were used to deploy the file on the HTTPD server. Please also refer the screenshot image of the successful build and deployment.

```
yum install httpd -y
systemctl start httpd
chkconfig httpd on
echo "<h1>Hello world, this message is hosted on Slave-2</h1>" > /var/www/html/index.html
chmod 777 /var/www/html/index.html
curl ifconfig.me
```



The two different IP addresses indicate that the index.html file is running successfully on HTTPD server (slave-2) with IP address of **65.2.150.185** default Port 80, whereas, the Jenkins-master IP address is **13.127.34.147** running on Port 8080

7. For the 2<sup>nd</sup> problem statement Git, and Tomcat 9 Apache server were separately installed on Slave-1 machine. The following commands were run to install the Git, and download and start the Tomcat Server.

```
yum install git -y
wget https://d1cdn.apache.org/tomcat/tomcat-9/v9.0.74/bin/apache-tomcat-9.0.74.zip
unzip apache-tomcat-9.0.74.zip
rm -rf apache-tomcat-9.0.74.zip
cd apache-tomcat-9.0.74
cd bin
./startup.sh
```

8. After starting the Tomcat server on slave-1, the project was configured with github to clone the game-of-life repository into the workspace of slave-1. BUILD of the Project game-of-life was done with the help of **Maven** which was already configured inside the project by using the 'Invoking the top level targets in Maven'. The following commands were used:

```
clean install
skipTests=true
```

9. Using the **Build Actions** in Execute Shell, the following commands were used to deploy the **gameoflife.war** file on the Tomcat Apache server. Please refer the screenshot image of the successful build and deployment.

```
chmod -R 777 /mnt/game-of-life
cp -r /mnt/game-of-life/gameoflife-web/target/gameoflife.war
/mnt/servers/apache-tomcat-9.0.74/webapps
```

The screenshot shows a Jenkins console output on the left and a web browser on the right. The console output displays the successful build of the game-of-life project, including the deployment of the gameoflife.war file to the Tomcat server. The web browser shows the 'Welcome to Conway's Game Of Life!' page, which includes a description of the game and a 'New Game' button.

Dashboard > game-of-life-slave > #5

```
[INFO] gameoflife-build .....
[INFO] gameoflife-core .....
[INFO] gameoflife-web .....
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 21.548 s
[INFO] Finished at: 2023-04-28T09:51:10Z
[INFO] -----
[game-of-life] $ /bin/sh -xe /tmp/jenkins-XXXXXX
+ chmod -R 777 /mnt/game-of-life
+ cp -r /mnt/game-of-life/gameoflife-web/target/gameoflife.war /mnt/servers/apache-tomcat-9.0.74/webapps
+ curl ifconfig.me
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total   Spent    Left   Speed
  0     0    0     0    0     0      0  0 --:--:-- --:--:-- --:--:--    0
  0     0    0     0    0     0      0  0 --:--:-- --:--:-- --:--:--    0
100  13 100   13    0     0   44    0 --:--:-- --:--:-- --:--:--   44
13.127.23.224Finished: SUCCESS
```

Welcome to Conway's Game Of Life!

This is a really cool web version of Conway's famous Game Of Life. The Game of Life is a cellular automaton devised by the British mathematician John Horton Conway way back in 1970.

The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, live or dead. Every cell interacts with its eight neighbors, which are the cells that are directly horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

- Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.
- Any live cell with more than three live neighbours dies, as if by overcrowding.
- Any live cell with two or three live neighbours lives on to the next generation.
- Any dead cell with exactly three live neighbours becomes a live cell.

New Game

Game Of Life version 1.0: SNAPSHOT (build job game-of-life: #13)

The two different IP addresses indicate that the gameoflife.war file is deployed successfully on Tomcat Apache server (slave-1) with IP address of **13.123.23.224** on Port 8080/gameoflife/, whereas, the Jenkins-master IP address is **13.127.34.147** running on default Port 8080/Jenkins/