RELEASE part of Integrated Learning Project (ILP):

Prerequisite:

1. You should have completed the Build part of ILP and ready with the war file
2. You should have completed the hands-on activity provided for Day16 to Day20.

What to do?

1. Pull the required environment image (Ex: tomcat) to create tomcat container and copy the application (Ex: Application.war) into container (Copy to Webapps folder using docker cp command).
2. Use Volume sharing concept to share the application (Ex: Application.war) into container through an overlaid volume.
3. Write Dockerfile to implement the required environment (Ex: JDK and Tomcat) and deploy the application inside the image.
4. Create a Docker Service that would put your application into production on Docker Swarm
5. Use Jenkins to deploy your image into container
6. Use Jenkins to build docker image, create container to deploy application.

What is Required?

1. Steps 1 and 2 are part of Handson activity and are used to test whether deployment can be done.
2. Steps 3 and 4 are essential and are part of deliverables to be submitted
3. Steps 5 and above are optional and are not part of deliverable and are used as stepping stone for Capstone Project.

What to submit as Deliverable?

1. Dockerfile listing

Capture the following commands in your script file, and submit along with screenshot. Screenshot may be submitted as png file or may be copied to word document before submitting in GITLAB.

Example:

```
osgdev@TG-DevOps-OS004:~/dockerlab/java$ cat Dockerfile
FROM java:8
WORKDIR /appjava
COPY . /appjava
RUN javac HelloWorld.java
RUN jar cfm HelloWorld.jar manifest.txt HelloWorld.class
CMD java -jar HelloWorld.jar

osgdev@TG-DevOps-OS004:~/dockerlab/java$ docker build -t hellojava6 .
Sending build context to Docker daemon   5.12kB
Step 1/6 : FROM java:8
 ---> d23bdf5b1b1b
Step 2/6 : WORKDIR /appjava
 ---> Using cache
 ---> 8e8008c6d6ac
```

Sensitivity: Internal & Restricted

```
Step 3/6 : COPY . /appjava
 ---> ab3a849e71d7
Step 4/6 : RUN javac HelloWorld.java
 ---> Running in 15e0ea8d81a1
Removing intermediate container 15e0ea8d81a1
 ---> 7b0fac9e364b
Step 5/6 : RUN jar cfm HelloWorld.jar manifest.txt HelloWorld.class
 ---> Running in 8ddac6f4cb34
Removing intermediate container 8ddac6f4cb34
 ---> 4aaf269031af
Step 6/6 : CMD java -jar HelloWorld.jar
 ---> Running in b771eacc021e
Removing intermediate container b771eacc021e
 ---> 0d9a1c04714b
Successfully built 0d9a1c04714b
Successfully tagged hellojava6:latest

osgdev@TG-DevOps-OS004:~/dockerlab/python$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hellojava6          latest              02b691805eb2        10 seconds ago      150MB

osgdev@TG-DevOps-OS004:~/dockerlab/python$ docker service create <service
name> <replicas> <Imagename>
```

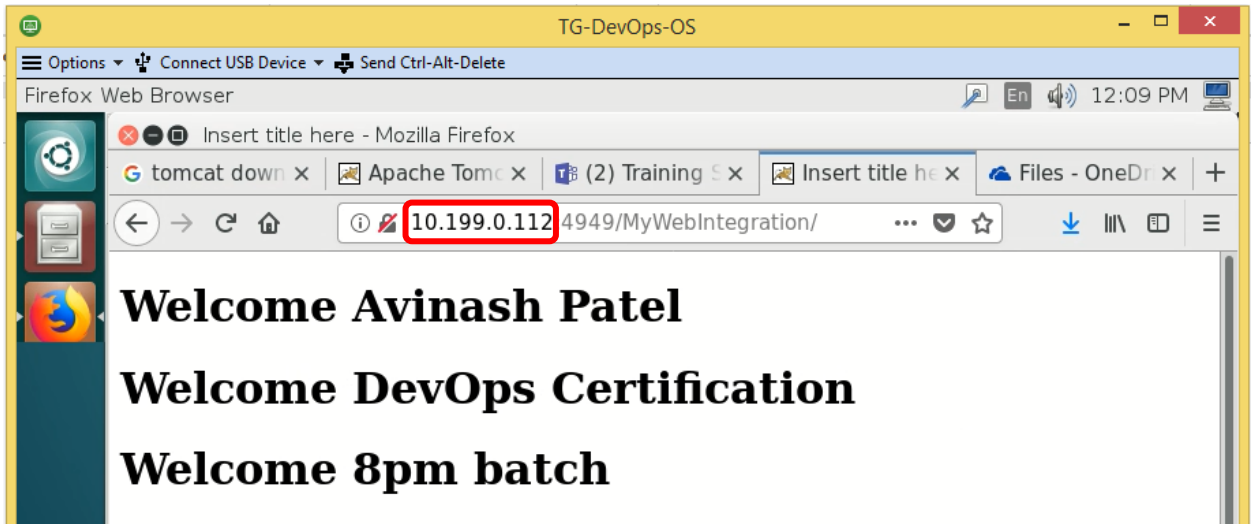
2. You must submit the screen shot capturing the output of your project:

Make sure the IP address of your machine shown below matches with the IP address appearing in the screenshot (Do not use localhost:4949)

Example: Capture the IP address of your machine:

```
osgdev@TG-DevOps-OS004:~$ ifconfig
ens160    Link encap:Ethernet  HWaddr 00:50:56:a0:70:f2
          inet addr:10.199.0.112  Bcast:10.199.15.255  Mask:255.255.240.0
          inet6 addr: fe80::250:56ff:fea0:70f2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16552146 errors:0 dropped:0 overruns:0 frame:0
          TX packets:231571 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1896872943 (1.8 GB)  TX bytes:77480740 (77.4 MB)
```

Sensitivity: Internal & Restricted



Sensitivity: Internal & Restricted