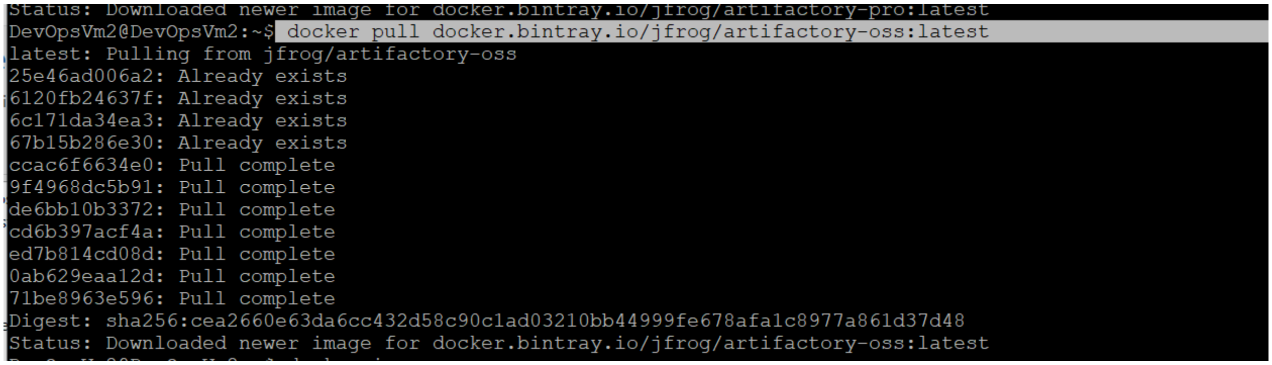
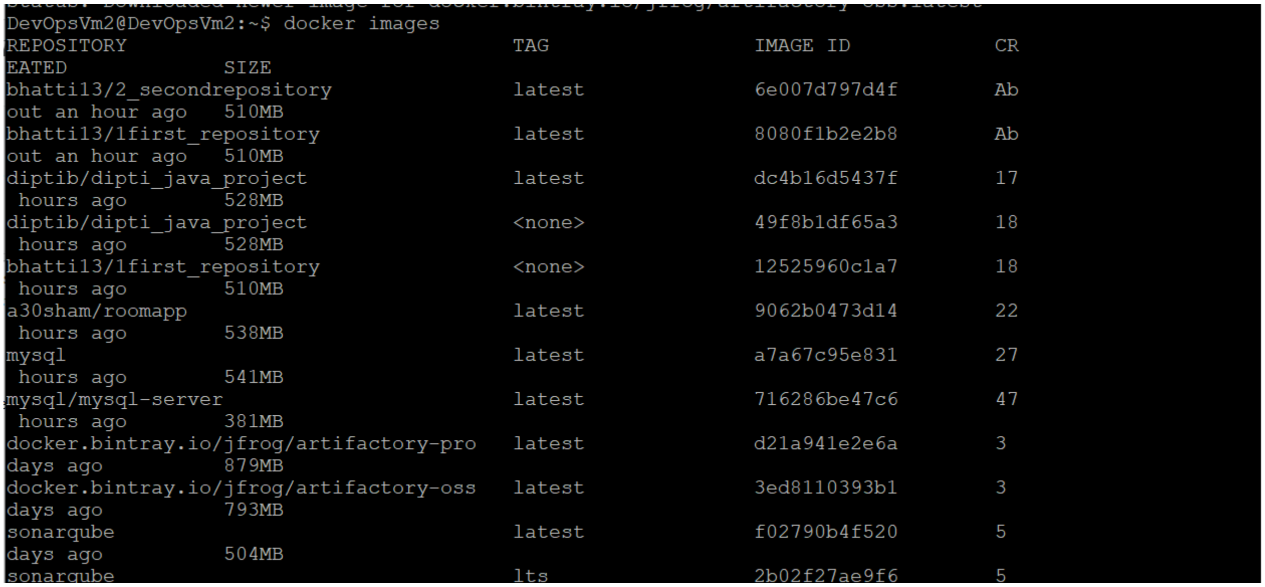
Installing Jfrog using Docker and creating the package of Application and store in Artifactory

1. To install jfrog use the command below :
   docker pull docker.bintray.io/jfrog/artifactory-oss:latest



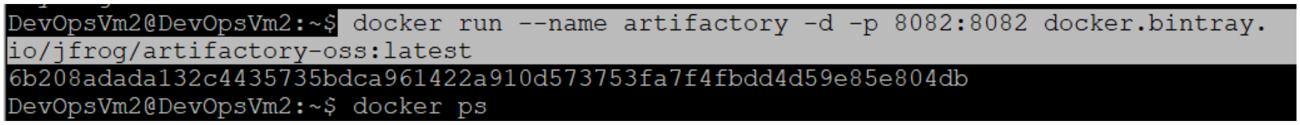
```
Status: Downloaded newer image for docker.bintray.io/jfrog/artifactory-pro:latest
DevOpsVm2@DevOpsVm2:~$ docker pull docker.bintray.io/jfrog/artifactory-oss:latest
latest: Pulling from jfrog/artifactory-oss
25e46ad006a2: Already exists
6120fb24637f: Already exists
6c171da34ea3: Already exists
67b15b286e30: Already exists
ccac6f6634e0: Pull complete
9f4968dc5b91: Pull complete
de6bb10b3372: Pull complete
cd6b397acf4a: Pull complete
ed7b814cd08d: Pull complete
0ab629eaa12d: Pull complete
71be8963e596: Pull complete
Digest: sha256:cea2660e63da6cc432d58c90c1ad03210bb44999fe678afa1c8977a861d37d48
Status: Downloaded newer image for docker.bintray.io/jfrog/artifactory-oss:latest
```

2. To check all images in the vm use the below command :
   docker images



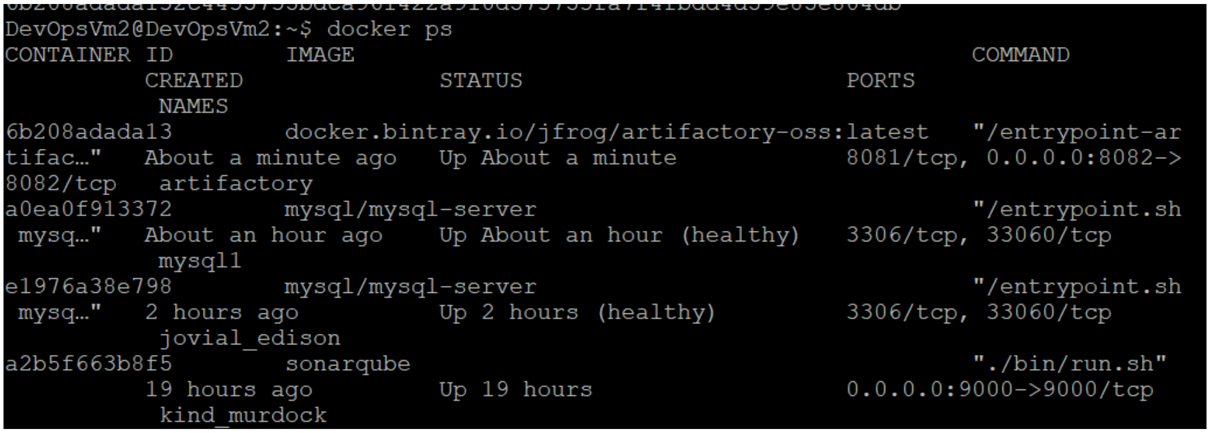
```
DevOpsVm2@DevOpsVm2:~$ docker images
REPOSITORY                               TAG                 IMAGE ID            CR
EATED                SIZE
bhatti13/2_secondrepository              latest              6e007d797d4f        Ab
out an hour ago      510MB
bhatti13/1first_repository               latest              8080f1b2e2b8        Ab
out an hour ago      510MB
diptib/dipti_java_project                latest              dc4b16d5437f        17
 hours ago           528MB
diptib/dipti_java_project                <none>              49f8b1df65a3        18
 hours ago           528MB
bhatti13/1first_repository               <none>              12525960c1a7        18
 hours ago           510MB
a30sham/roomapp                          latest              9062b0473d14        22
 hours ago           538MB
mysql                                    latest              a7a67c95e831        27
 hours ago           541MB
mysql/mysql-server                       latest              716286be47c6        47
 hours ago           381MB
docker.bintray.io/jfrog/artifactory-pro  latest              d21a941e2e6a        3
days ago             879MB
docker.bintray.io/jfrog/artifactory-oss  latest              3ed8110393b1        3
days ago             793MB
sonarqube                                latest              f02790b4f520        5
days ago             504MB
sonarqube                                lts                 2b02f27ae9f6        5
```

3. To run the jfrog on port 8082 we will be using command as given below :
   docker run --name artifactory -d -p 8082:8082 docker.bintray. io/jfrog/artifactory-oss:latest



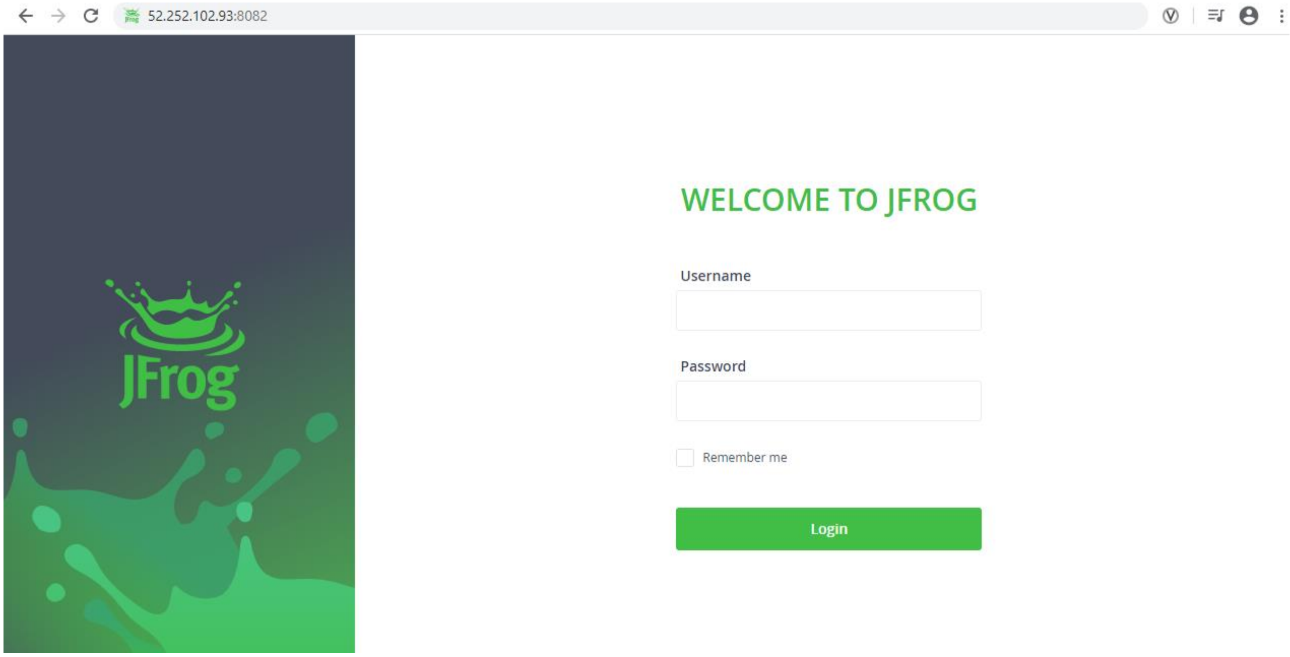
```
DevOpsVm2@DevOpsVm2:~$ docker run --name artifactory -d -p 8082:8082 docker.bintray.
io/jfrog/artifactory-oss:latest
6b208adada132c4435735bdca961422a910d573753fa7f4fbdd4d59e85e804db
DevOpsVm2@DevOpsVm2:~$ docker ps
```

4. To display all the containers in the docker we will be using the following command :
   docker ps



```
DevOpsVm2@DevOpsVm2:~$ docker ps
CONTAINER ID        IMAGE                                              COMMAND
          CREATED             STATUS                           PORTS
           NAMES
6b208adada13        docker.bintray.io/jfrog/artifactory-oss:latest   "/entrypoint-ar
tifac…"   About a minute ago   Up About a minute               8081/tcp, 0.0.0.0:8082->
8082/tcp   artifactory
a0ea0f913372        mysql/mysql-server                                "/entrypoint.sh
 mysq…"   About an hour ago    Up About an hour (healthy)      3306/tcp, 33060/tcp
           mysql1
e1976a38e798        mysql/mysql-server                                "/entrypoint.sh
 mysq…"   2 hours ago          Up 2 hours (healthy)            3306/tcp, 33060/tcp
           jovial_edison
a2b5f663b8f5        sonarqube                                         "./bin/run.sh"
          19 hours ago         Up 19 hours                     0.0.0.0:9000->9000/tcp
           kind_murdock
```

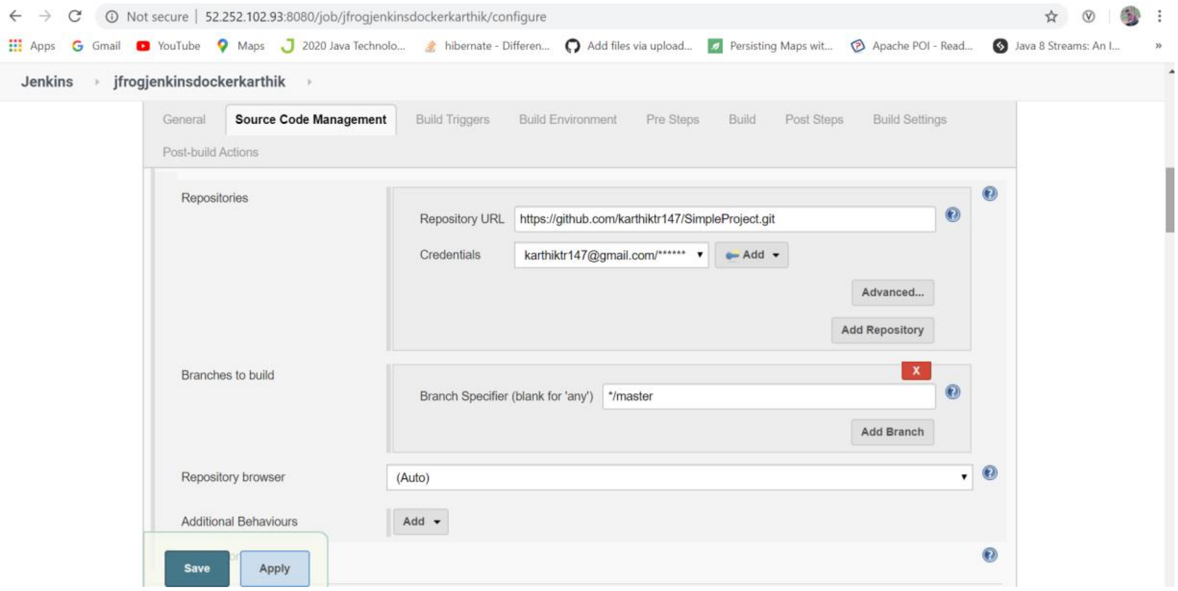
5. After Running jfrog image on docker use the url http://server:8082/artifactory and register to jfrog



Creating the Package of application and storing it in the Artifactory :

Step 1: Create a new

project in Jenkins, clone your project of your git by mentioning the repository url of your project in source code management and give the credentials as shown in the image below :



Repositories

Repository URL: https://github.com/karthiktr147/SimpleProject.git

Credentials: karthiktr147@gmail.com/******

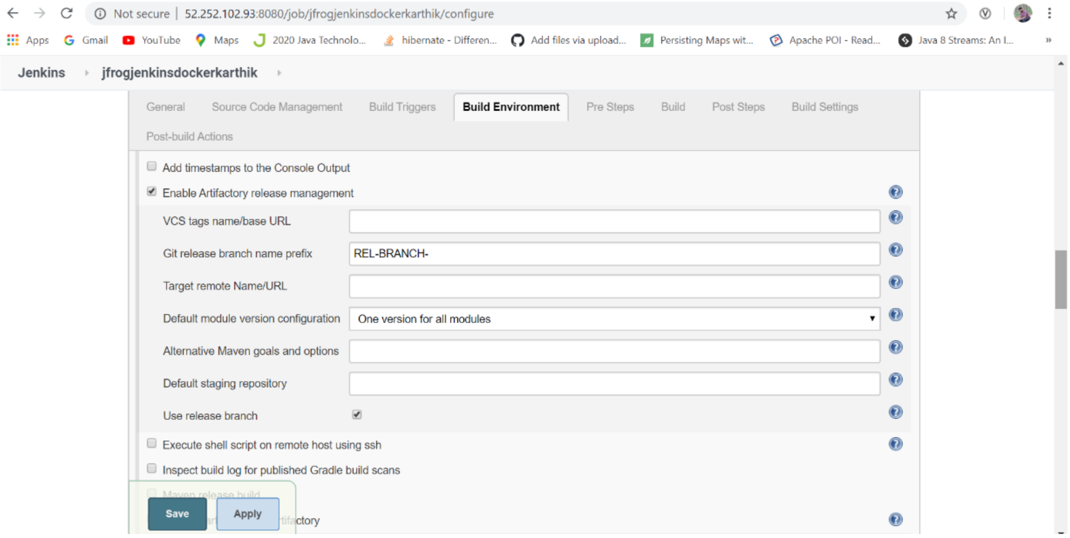
Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Additional Behaviours

Step 2: In the Build Environment tick Enable Artifactory release management as shown in the image below.



☐ Add timestamps to the Console Output

☑ Enable Artifactory release management

VCS tags name/base URL:

Git release branch name prefix: REL-BRANCH-

Target remote Name/URL:

Default module version configuration: One version for all modules

Alternative Maven goals and options:

Default staging repository:

Use release branch ☑

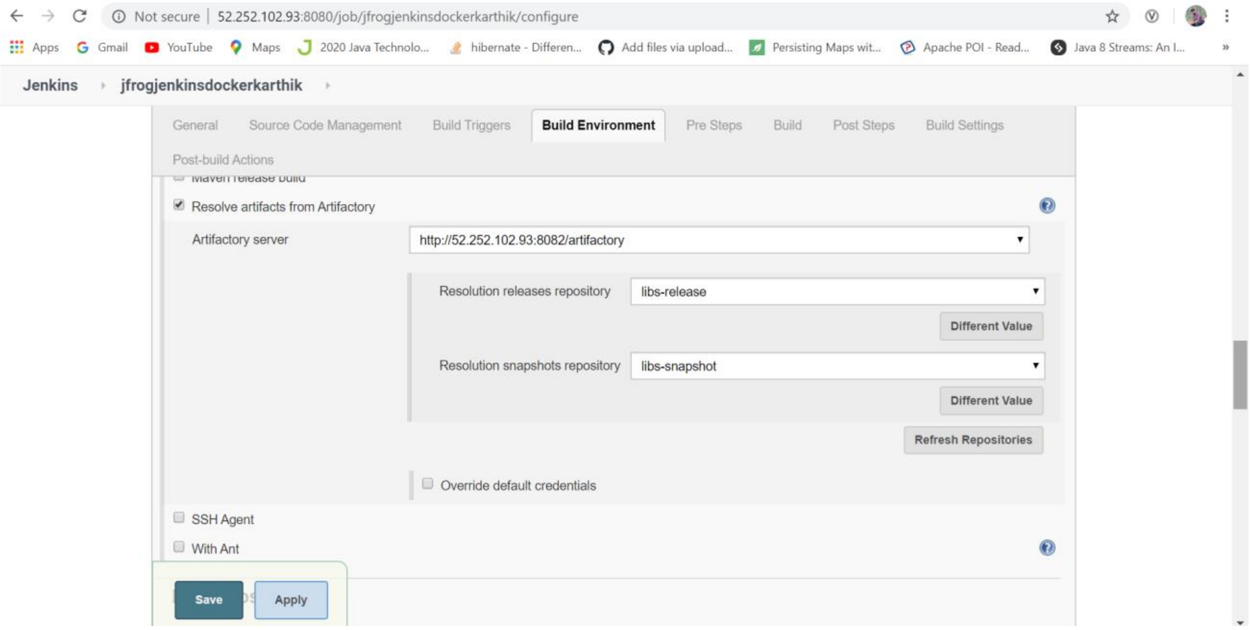
☐ Execute shell script on remote host using ssh

☐ Inspect build log for published Gradle build scans

☐ Maven release build

tifactory

Step 3: In the Build environment tick Resolve artifacts from Artifactory and click on refresh Repositories and from the dropdown in Resolution snapshots repository select libs-snapshot.



Not secure | 52.252.102.93:8080/job/jfrogjenkinsdockerkarthik/configure

Jenkins › jfrogjenkinsdockerkarthik ›

☑ Resolve artifacts from Artifactory

Artifactory server: http://52.252.102.93:8082/artifactory

Resolution releases repository: libs-release

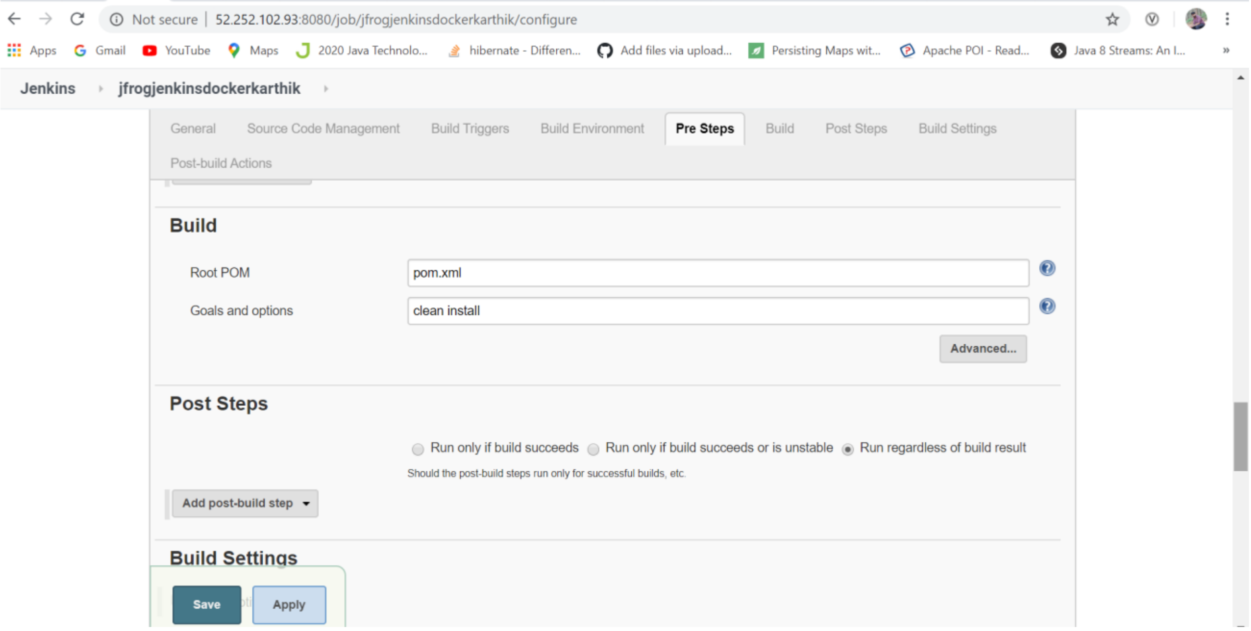
Resolution snapshots repository: libs-snapshot

□ Override default credentials

□ SSH Agent

□ With Ant

Step 4 : In Build mention I have mentioned clean install.



Not secure | 52.252.102.93:8080/job/jfrogjenkinsdockerkarthik/configure

Jenkins › jfrogjenkinsdockerkarthik ›

Build

Root POM: pom.xml

Goals and options: clean install

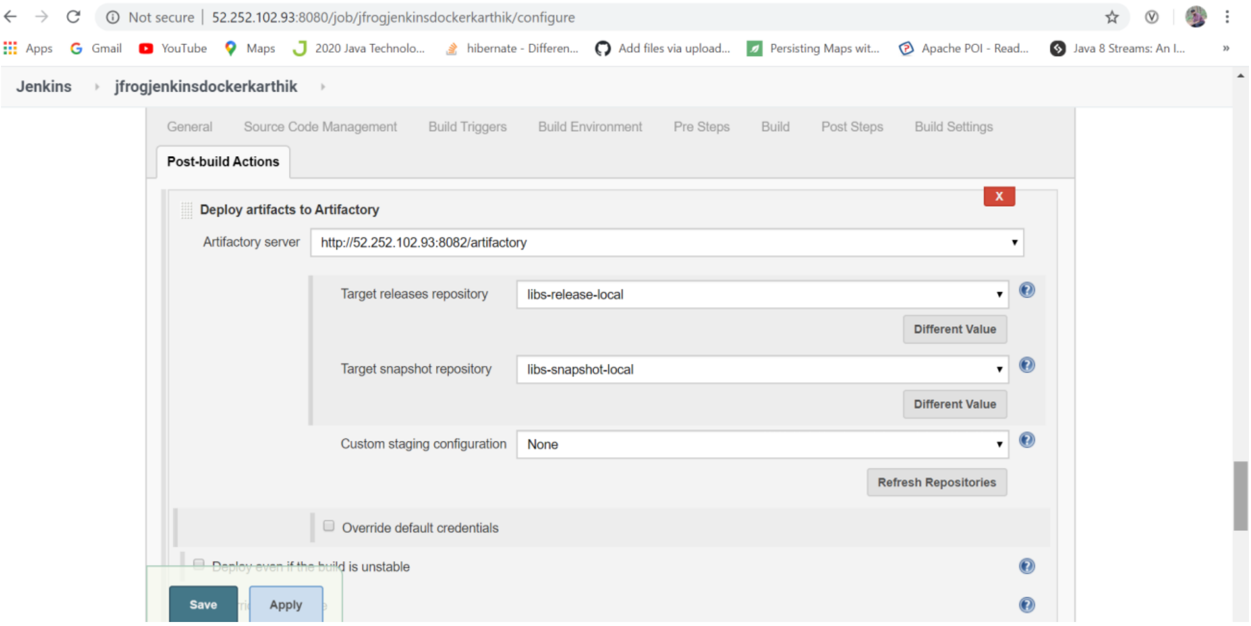
Post Steps

○ Run only if build succeeds ○ Run only if build succeeds or is unstable ● Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Build Settings

Step 5: In post build actions tick select Deploy artifacts to Artifactory and click on refresh Repositories and from the dropdown in Target snapshots repository select libs-snapshot.



Deploy artifacts to Artifactory

Artifactory server: http://52.252.102.93:8082/artifactory

Target releases repository: libs-release-local

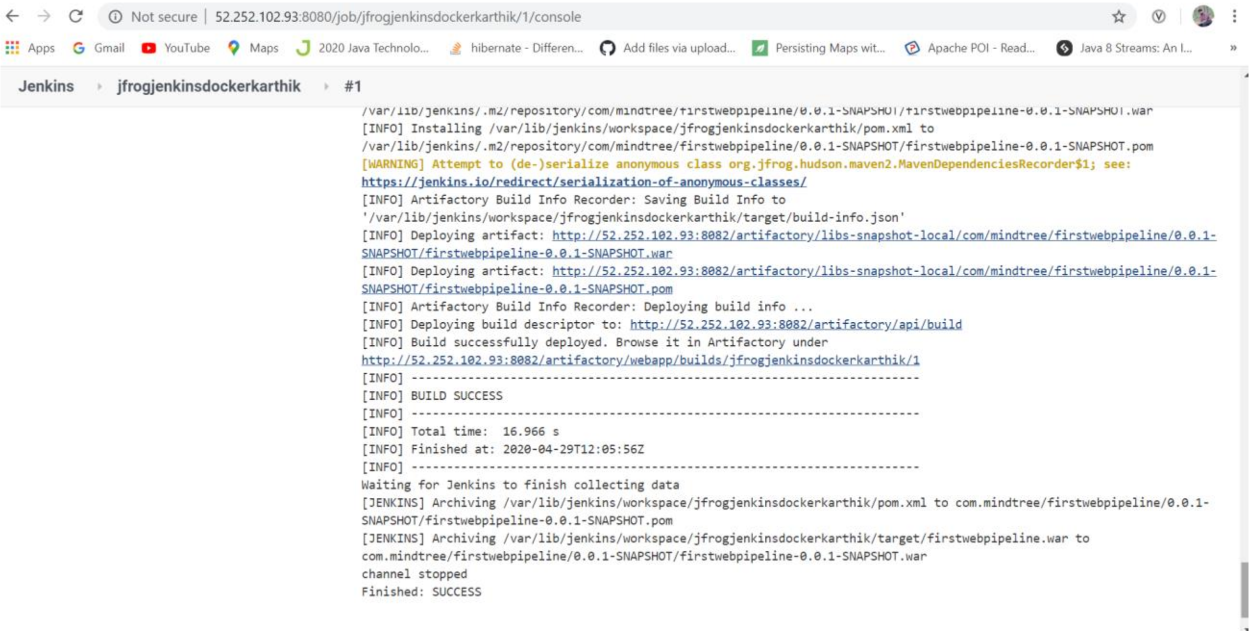
Target snapshot repository: libs-snapshot-local

Custom staging configuration: None

□ Override default credentials

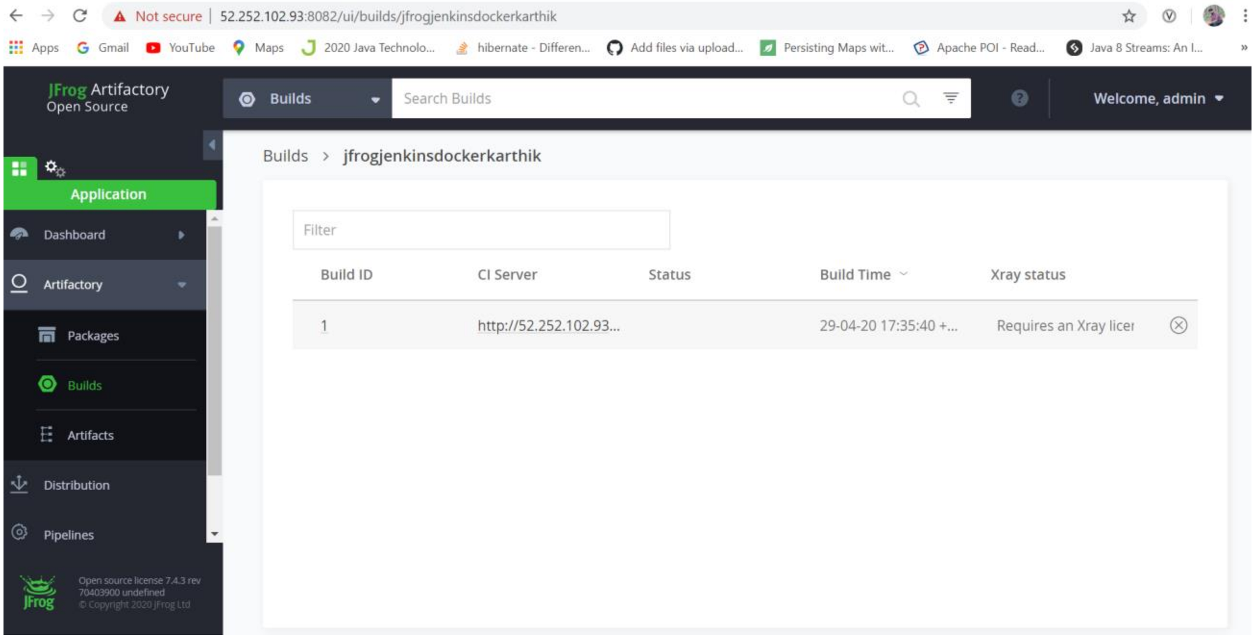
□ Deploy even if the build is unstable

Step 6: After Following above click on save and apply and click on build now after that below is the console output of the build results.



```
/var/lib/jenkins/.m2/repository/com/mindtree/firstwebpipeline/0.0.1-SNAPSHOT/firstwebpipeline-0.0.1-SNAPSHOT.war
[INFO] Installing /var/lib/jenkins/workspace/jfrogjenkinsdockerkarthik/pom.xml to
/var/lib/jenkins/.m2/repository/com/mindtree/firstwebpipeline/0.0.1-SNAPSHOT/firstwebpipeline-0.0.1-SNAPSHOT.pom
[WARNING] Attempt to (de-)serialize anonymous class org.jfrog.hudson.maven2.MavenDependenciesRecorder$1; see:
https://jenkins.io/redirect/serialization-of-anonymous-classes/
[INFO] Artifactory Build Info Recorder: Saving Build Info to
'/var/lib/jenkins/workspace/jfrogjenkinsdockerkarthik/target/build-info.json'
[INFO] Deploying artifact: http://52.252.102.93:8082/artifactory/libs-snapshot-local/com/mindtree/firstwebpipeline/0.0.1-
SNAPSHOT/firstwebpipeline-0.0.1-SNAPSHOT.war
[INFO] Deploying artifact: http://52.252.102.93:8082/artifactory/libs-snapshot-local/com/mindtree/firstwebpipeline/0.0.1-
SNAPSHOT/firstwebpipeline-0.0.1-SNAPSHOT.pom
[INFO] Artifactory Build Info Recorder: Deploying build info ...
[INFO] Deploying build descriptor to: http://52.252.102.93:8082/artifactory/api/build
[INFO] Build successfully deployed. Browse it in Artifactory under
http://52.252.102.93:8082/artifactory/webapp/builds/jfrogjenkinsdockerkarthik/1
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  16.966 s
[INFO] Finished at: 2020-04-29T12:05:56Z
[INFO] ------------------------------------------------------------------------
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/jfrogjenkinsdockerkarthik/pom.xml to com.mindtree/firstwebpipeline/0.0.1-
SNAPSHOT/firstwebpipeline-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/jfrogjenkinsdockerkarthik/target/firstwebpipeline.war to
com.mindtree/firstwebpipeline/0.0.1-SNAPSHOT/firstwebpipeline-0.0.1-SNAPSHOT.war
channel stopped
Finished: SUCCESS
```

Below is the image of the artifacts added to jfrog artifactory .



Builds > jfrogjenkinsdockerkarthik

Build ID	CI Server	Status	Build Time	Xray status
1	http://52.252.102.93...		29-04-20 17:35:40 +...	Requires an Xray licer