**Prerequisites**

1. Create a group
2. Create a service account user
3. Add the service account user to the below groups
   1. To the newly created group
   2. OSE3: Openshift Enterprise 3 User/Developer Access Group
   3. OpenShiftContainerPlatform\_nonprod\_users
   4. OpenShiftContainerPlatform\_nonprod\_users\_DMZ
   5. dtr\_users
   6. github\_users
4. Setup a docker org. Open a help desk request to create a new docker organization. Provide them with the group name that you created and the name of the org. After that is done create a repo in the org.
5. Request OSE to be provisioned. Login with your user account after they are provisioned and give admin access to the service account user.
6. Setup Jenkins Job. There are lot of configurations you have to do here. If you are doing it for the first time then its best to get someone’s help. (note to self:- Document this). Make a note of the name of the job.

**Configurations that you need to add your git project.**

1. Add the following files to your project. All Files are below.
   1. Docker File
   2. Jenkins folder with Jenkins-build.props
   3. .gitignore file
2. Login to your github and go to your project
   1. Go to Settings
   2. Click on “Hooks & Services”
   3. For the Payload URL use this
      1. [https://jenkins.optum.com/rxclaimmodernization/job/**pattern\_deploy**/buildWithParameters?token=**my-token**&RepoName=**SpringBootExample**](https://jenkins.optum.com/rxclaimmodernization/job/pattern_deploy/buildWithParameters?token=my-token&RepoName=SpringBootExample)
      2. The **token** is the token you used while creating the **job** and repo is the name of your github project. Replace **pattern\_deploy** with the name of the Jenkins job
   4. For content type pick application/json
   5. Save the hook.

From now whenever you make changes to your git repo the Jenkins job would automatically kick off and deploy to OSE.

Docker File

RUN\_AS\_USER = the service user account you created in the prerequisites.

APP = The name of the app in the pom.xml.

VERSION = The version you specified in the pom.xml

FROM docker.optum.com/optum\_et/alpine:3.3

MAINTAINER RxClaim Architecture <rxclaim\_mod\_d01@ds.uhc.com>

ENV JAVA\_HOME /usr/lib/jvm/java-1.8-openjdk

ENV PATH $PATH:$JAVA\_HOME/bin

ENV JAVA\_ALPINE\_VERSION 8.92.14-r0

ENV AMQP\_HOST 172.30.122.196

#ENV AMQP\_HOST 10.1.62.9

ENV RUN\_AS\_USER **rxclaim\_mod\_d01**

ENV APP **modernizer**

ENV VERSION **0.0.1-SNAPSHOT**

RUN apk add --no-cache openjdk8="$JAVA\_ALPINE\_VERSION"

RUN mkdir /app

ADD target/$APP-$VERSION.jar /app/app.jar

RUN sh -c 'touch /app/app.jar'

RUN addgroup ${RUN\_AS\_USER} && \

adduser -D -G ${RUN\_AS\_USER} ${RUN\_AS\_USER} && \

chown -R ${RUN\_AS\_USER}:${RUN\_AS\_USER} /app

USER **rxclaim\_mod\_d01**

WORKDIR /app

EXPOSE 8080

CMD ["/usr/lib/jvm/java-1.8-openjdk/bin/java", "-Xms64M", "-Xmx1G", "-jar", "app.jar"]

jenkins-build.props

The entries highlighted in yellow need to be changed to match your project.

DOCK\_REPO is the path to your repo in Docker. The path is {ORG NAME}/{REPO NAME}

CLOUD\_APP is the name of the application that you want inside your project. You don’t have to create this. OSE will automatically do this for you.

CLOUD\_PROJECT is the name of the project that you used while provisioning your OSE instance.

#Jenkins Dev build environment properties

DOCK\_REPO=**rxclaim\_modernization/springbootexample**

DOCK\_LABEL=latest

# Oso Dev

CLOUD\_APP=**example**

CLOUD\_PROJECT=**rxclaim-modernization-services**

OSE\_SERVER=https://ocp-ctc-core-nonprod.optum.com

PLATFORM=Openshift

PROFILE=oso-dev,docker

# Mesos Dev

MESOS\_PROJECT=**rxclaim-modernization-services**

MESOS\_SERVER=http://dbslp1728/marathon/v2/apps

#PLATFORM=Mesos

#PROFILE=mesos-dev,docker

.gitignore

Make sure that your .gitignore has the following excludes. You can append to whatever your current .gitignore file has

-----------------------------------------.gitignore----------------------------------------------------------------

# Optum Standard Excludes

#Java Excludes

\*.class

\*.log

#Mobile tools for java(J2ME)

.mtj.tmp/

#packaging files

\*.jar

\*.war

\*.ear

#virtual machine crach logs

hs\_err\_pid\*

#Maven Excludes

target/

pom.xml.tag

pom.xml.releaseBackup

pom.xml.versionsBackup

pom.xml.next

release.properites

dependency-reduced-pom.xml

buildNumber.properties

.mvn/timing.properties

#Eclipse Excludes

.classpath

.factorypath

.tern-project

.gradle/

.project

.settings/

#Gradle Excludes

.gradle

build/

#Ignore Gradle GUI Config

gradle-app.settings

#Avoid Ignoring Gradle Wraper jar file

!gradle-wrapper.jar

# All bak files

\*.bak

#OS Generated Files

.DS\_Store

.DS\_Store?

.\_\*

Spotlight-V100

.Trashes

ehthumbs.db

Thumbs.db

#Packages

\*.7z

\*.dmz

\*.gz

\*.iso

\*.jar

\*.rar

\*.tar

\*.zip

#compiled sources

\*.com

\*.dll

\*.exe

\*.o

\*.so

#pmd files

.pmd

.pmdruleset.xml

----------------------------------------END .gitignore----------------------------------------------------------