9/4/2018

Peter Tam

Oracle

Reference Menu

Internal

Contents

[Install Docker on Windows 2](#_Toc523830199)

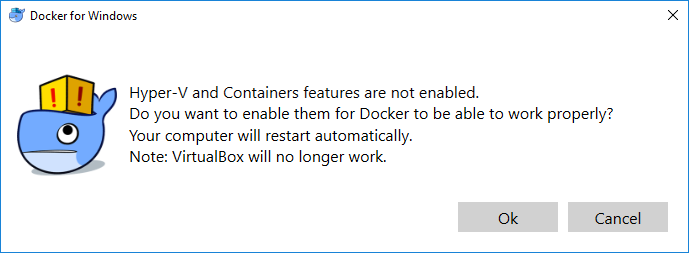
[Run Spring Boot App with Standalone WebLogic Server in STS IDE 10](#_Toc523830200)

[Reference of committing source to GitHub (New Repository) 26](#_Toc523830201)

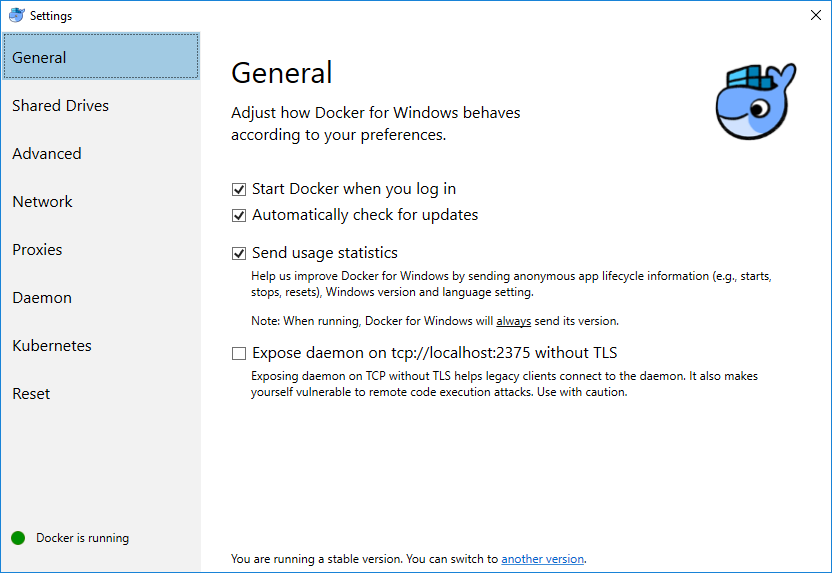
[Setup CAS + Cluster Environment 30](#_Toc523830202)

# Install Docker on Windows

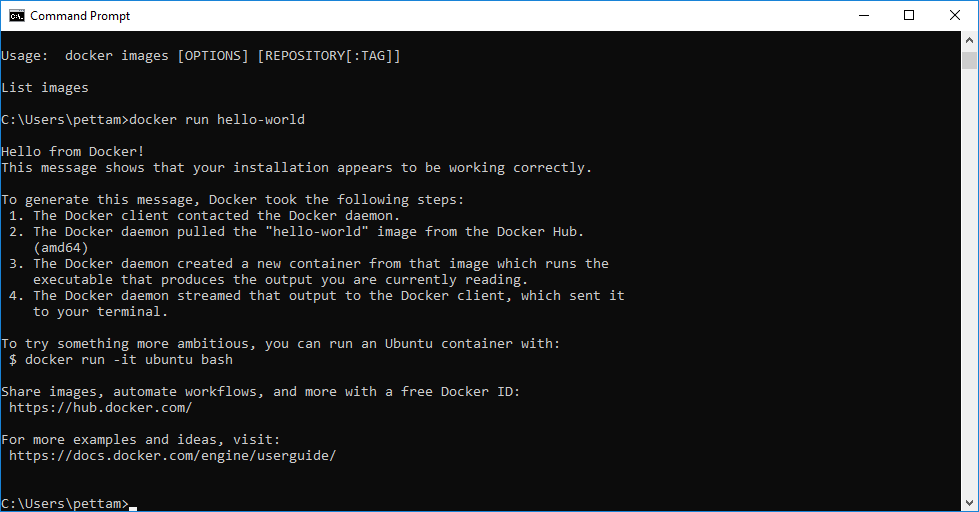
1. Download and install the Docker (Stable) version for Windows from <https://store.docker.com/editions/community/docker-ce-desktop-windows>
2. Enable Hyper-V and Containers Features. Don’t turn it on and use docker on windows if you have emulator (e.g. Android) running in Windows because this emulator must use HAX which is mutual exclusive with Hyper-V.



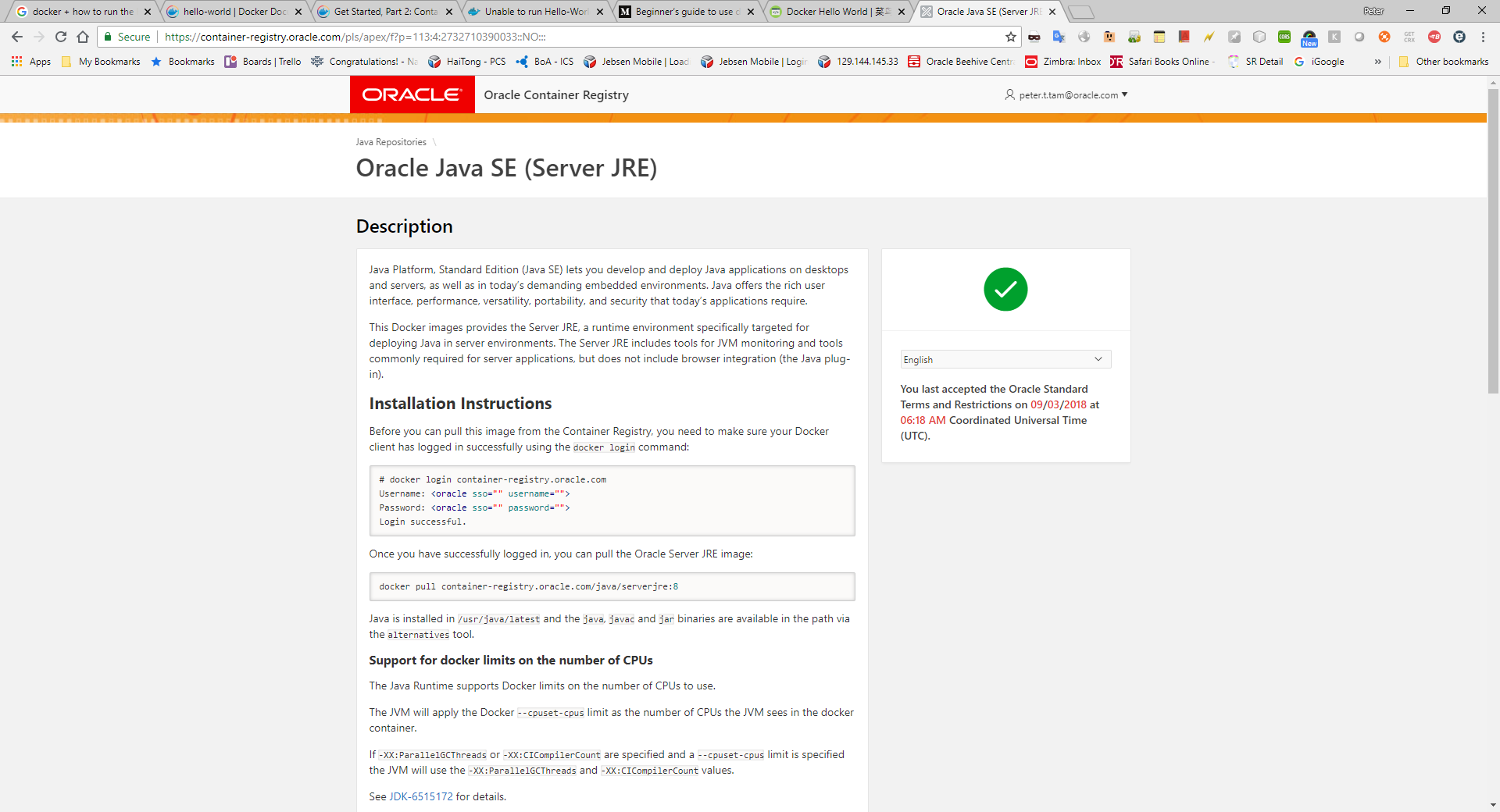
1. Docker is up and running



1. Verify the Docker by hello-world

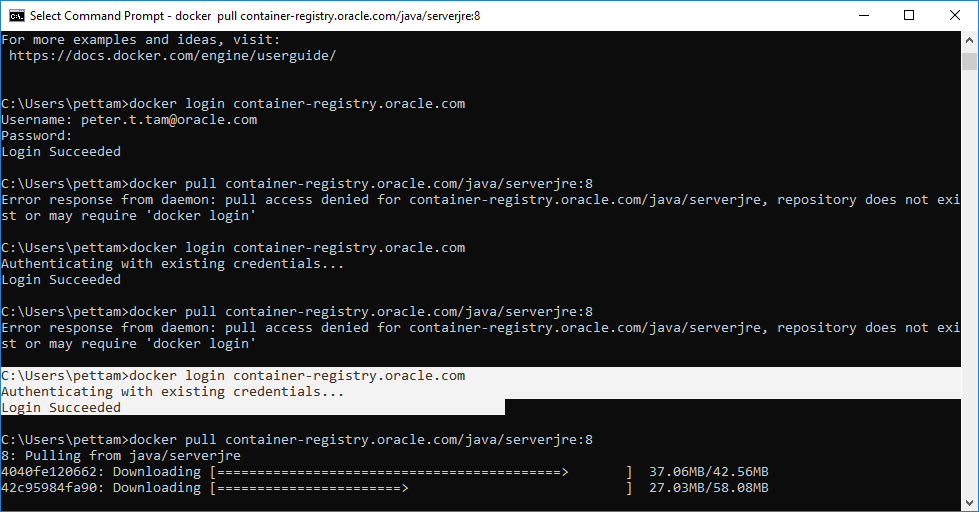


1. Access to and Login to the container-registry.orcle.com



1. Logon Oracle Container registry

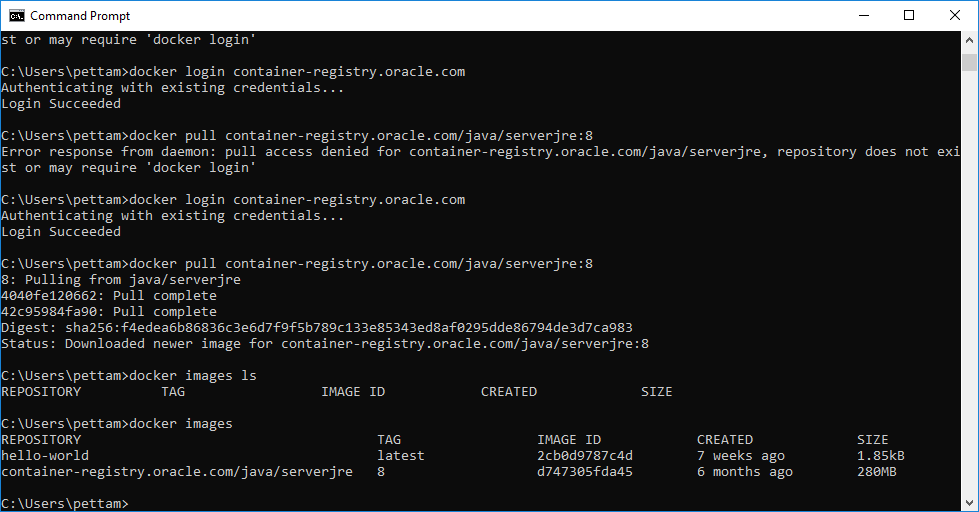
docker login container-registry.oracle.com



1. Download the Oracle JRE (server) if you need the JRE for Docker

docker pull container-registry.oracle.com/java/serverjre:8

1. Verify the Oracle Server JRE



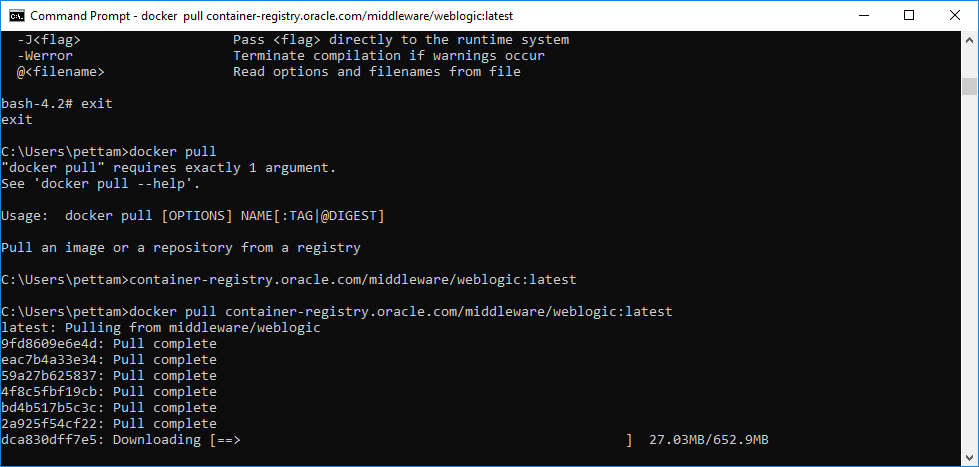
1. Run the Java in the docker

docker run -it -–name java8Test container-registry.oracle.com/java/serverjre:8 bash

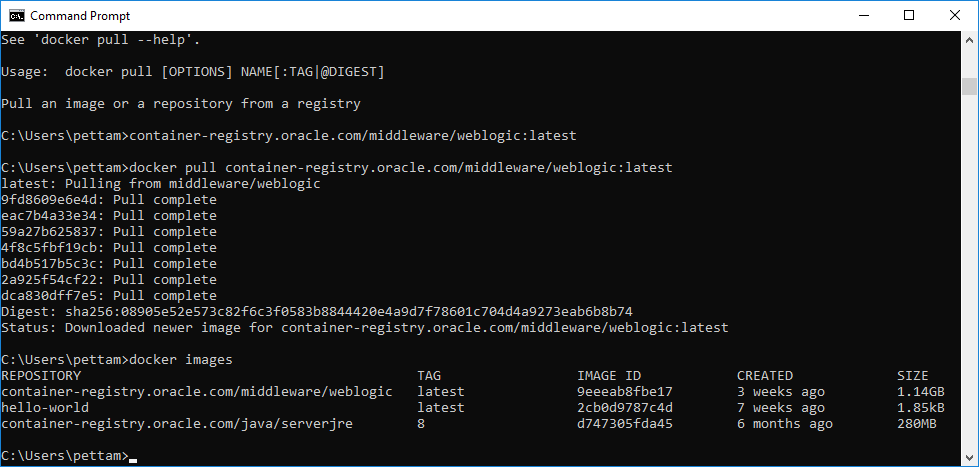
java -version

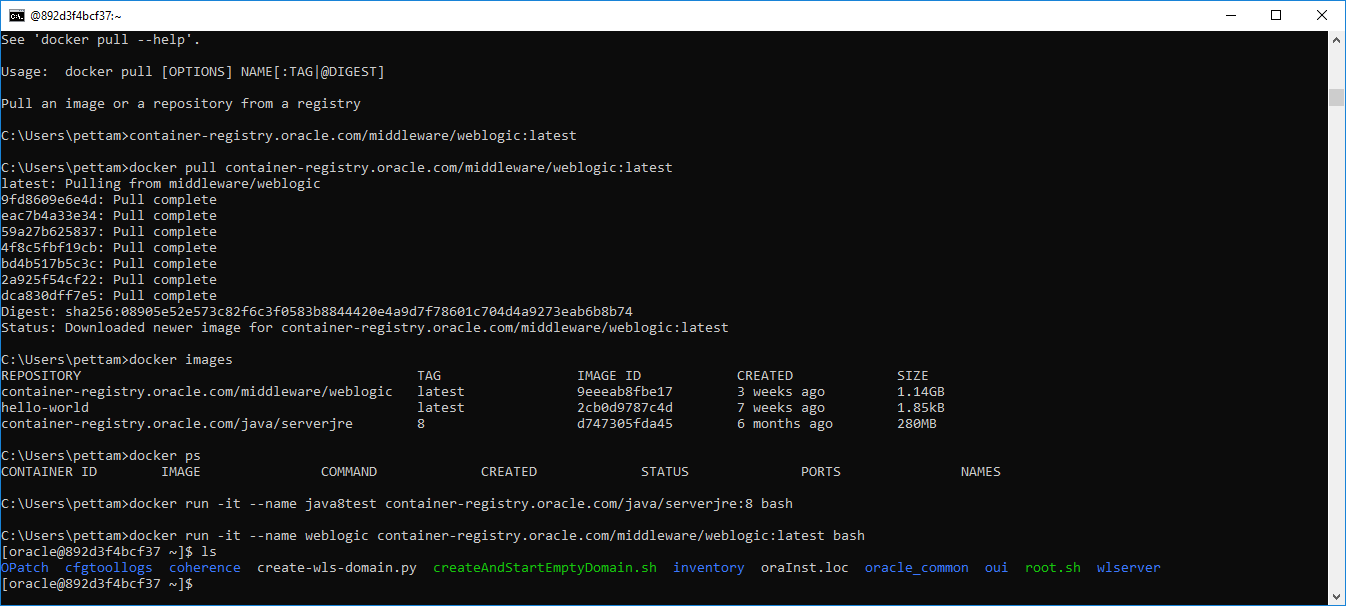
1. Pull the Weblogic Server Images if you need the Weblogic run on Docker

docker pull container-registry.oracle.com/middleware/weblogic:latest



1. Verify by docker images





1. Create default domain.properties file in your host directory (e.g. c:/users/pettam)

domain.properties

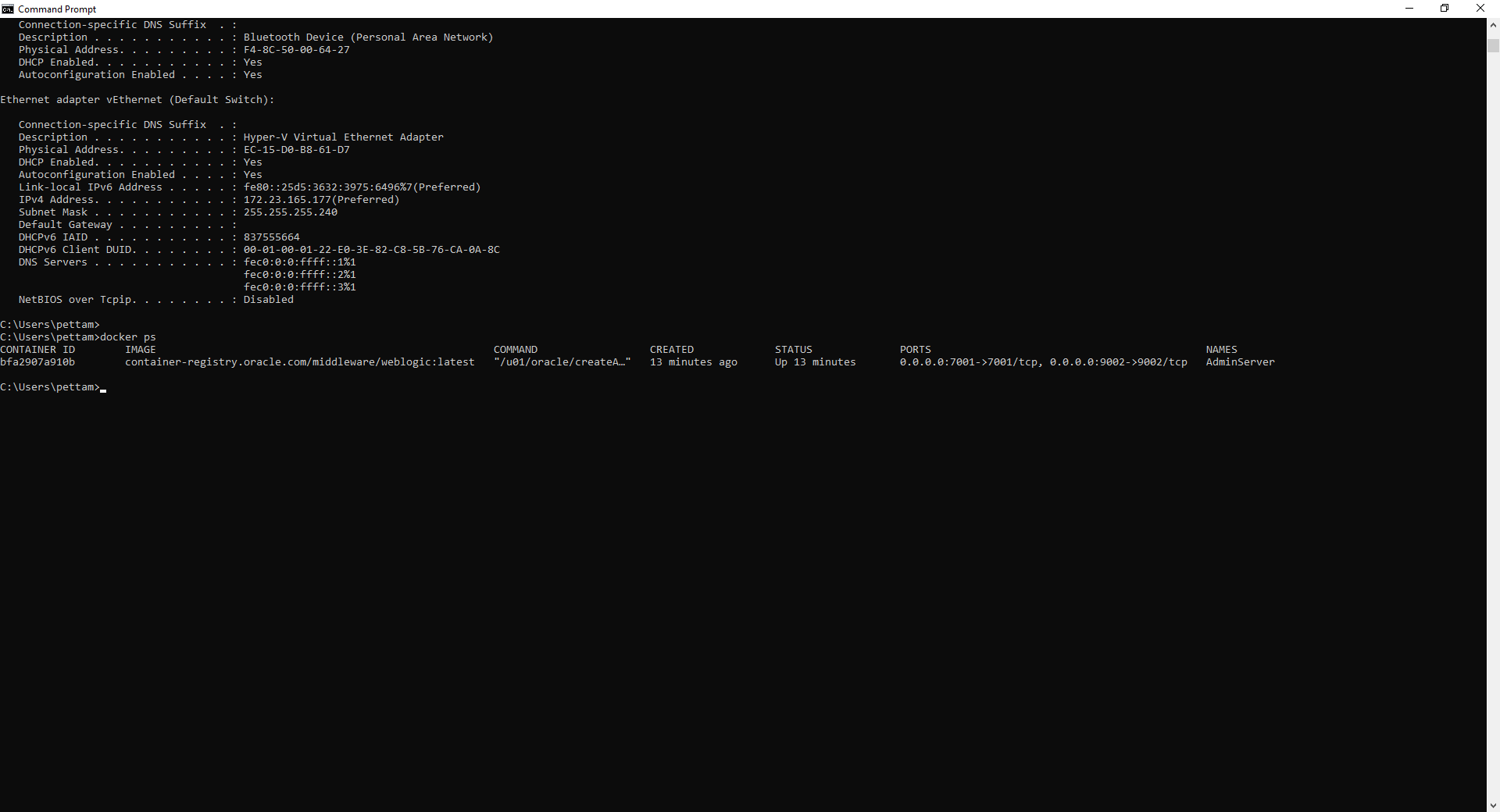
username=weblogic

password=welecome1

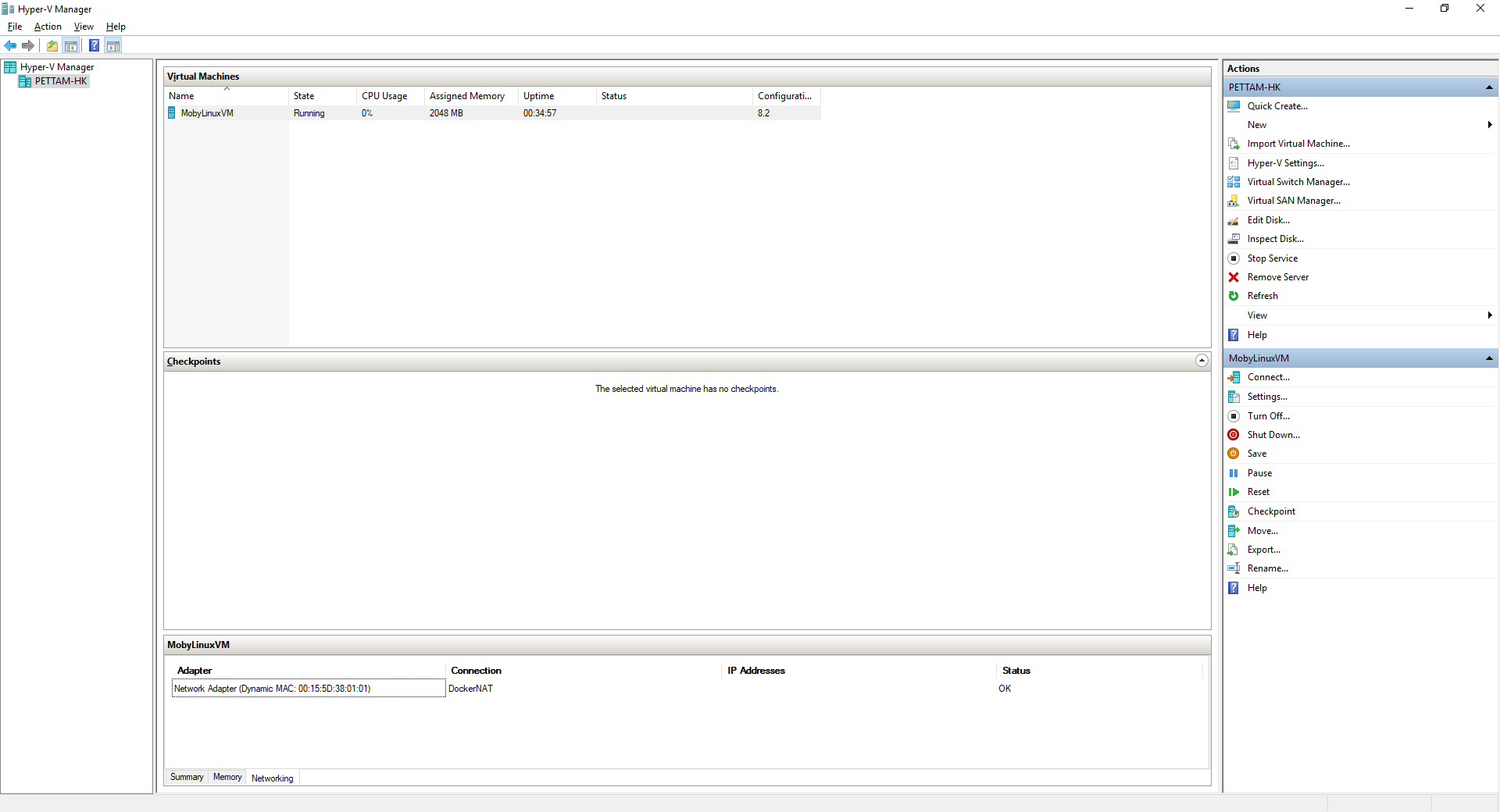
1. Based on the oracle container registry version. We can create & startup the Admin Server as below command. (Please note each docker image in the docker store has different startup)

docker run -itd --name AdminServer -p 7001:7001 -p 9002:9002 -v **c:/users/pettam/**domain.properties:**/u01/oracle/properties/domain.properties** container-registry.oracle.com/middleware/weblogic**:latest**

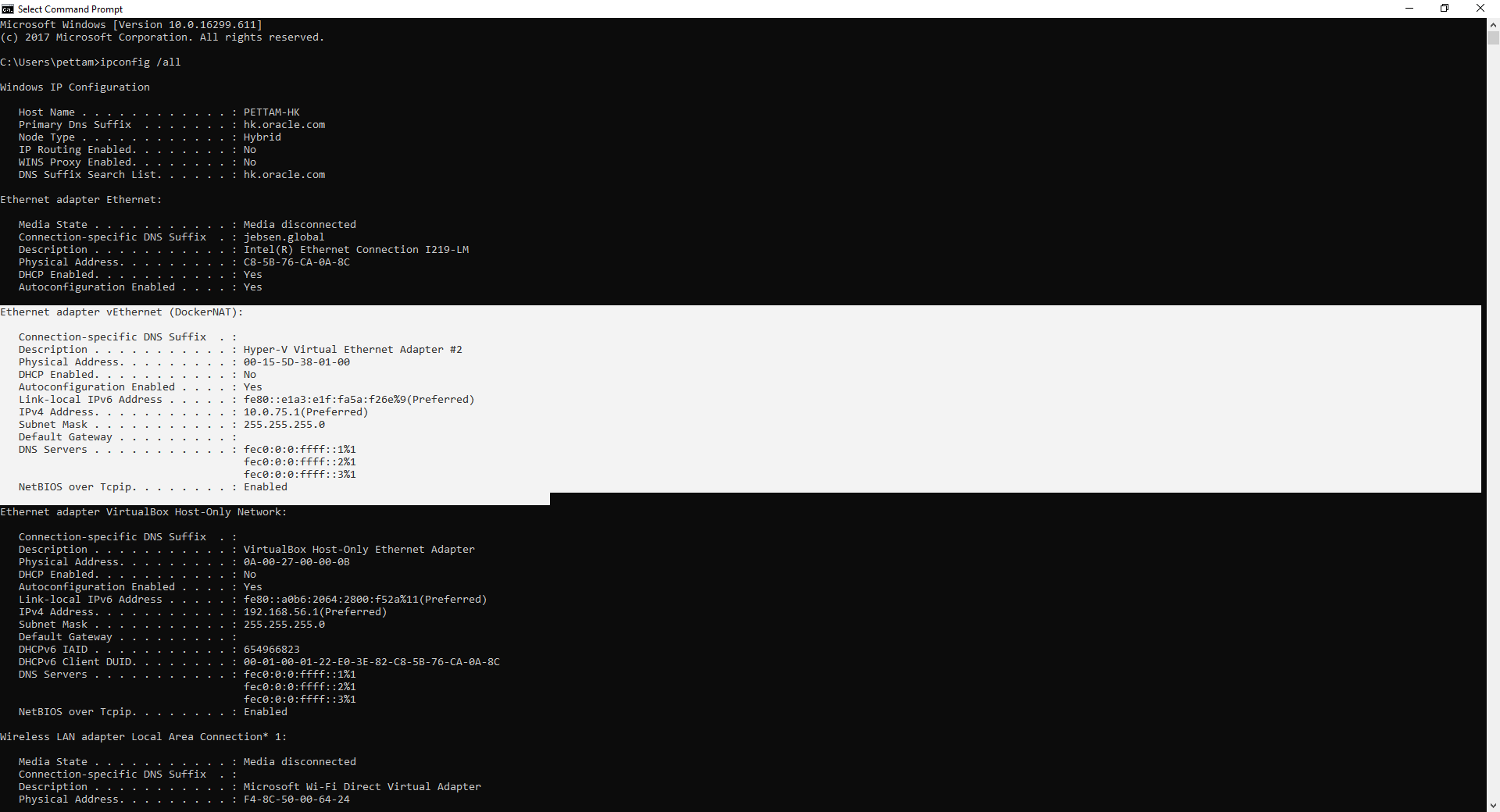
1. Verify it by docker ps



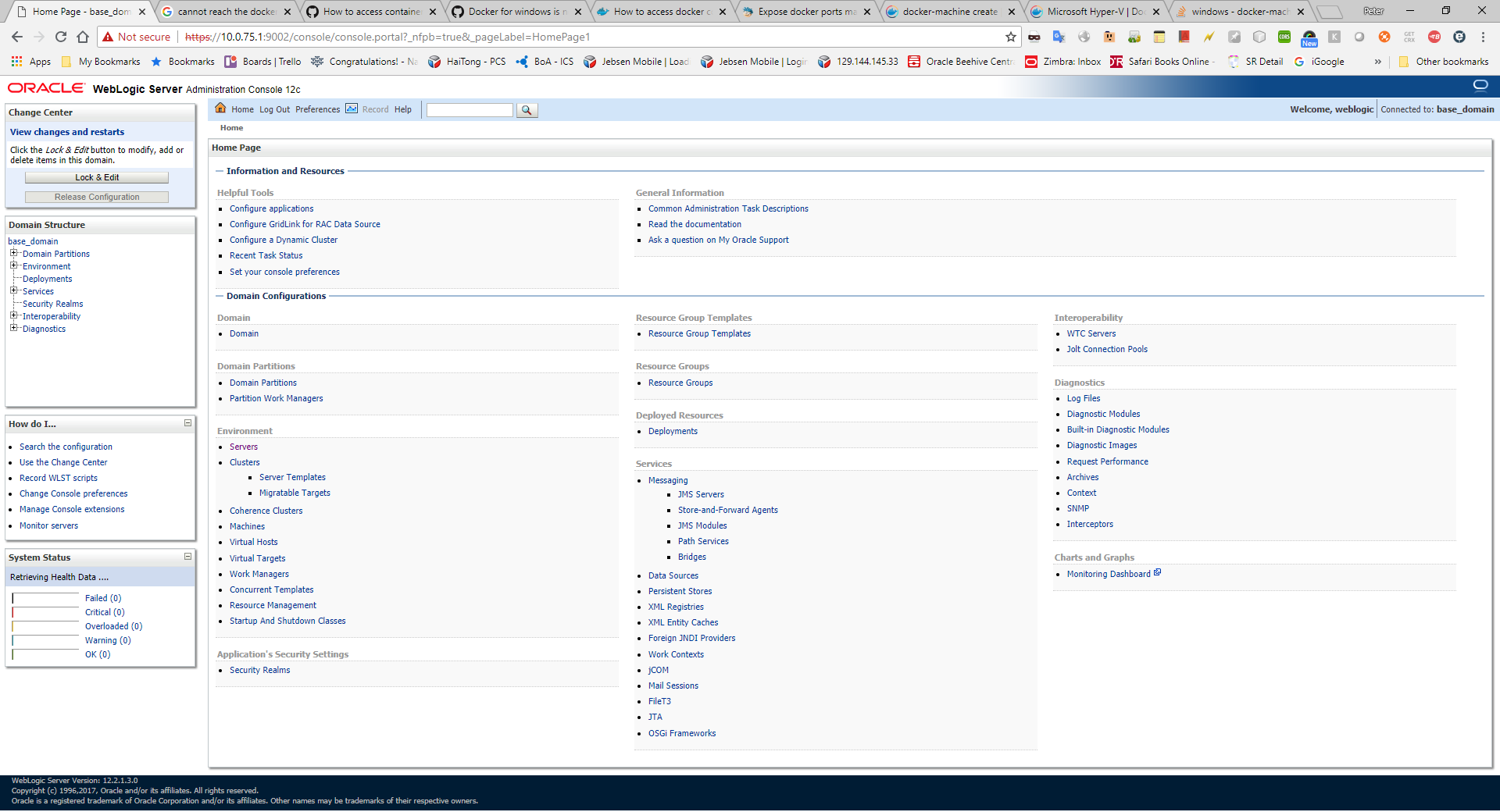
1. Find your docker-machine ip address by Hyper-V Manager
2. Lookup the mac address by Hyper-V Manager



1. Map the mac address from “ipconfig /all” (e.g. docker-machine ip is 10.0.75.1)



1. Access to the WebLogic Console with your browser (<https://docker-machine-ip-address:9002/console>) if you are running the Docker on windows



1. Reference Docker commands
   1. docker exec -t {Docker ID/Name} {command}

e.g. docker exec -t weblogic\_bash cat domain.properties

* 1. docker start/stop {Docker ID/Name}  
     e.g. docker start AdminServer (to start up the docker image and then you can see it by docker ps)

e.g. docker stop AdminServer (to shutdown the docker image)

* 1. docker attach {Docker ID/Name}

e.g. docker attach AdminServer (to attach to the docker image)

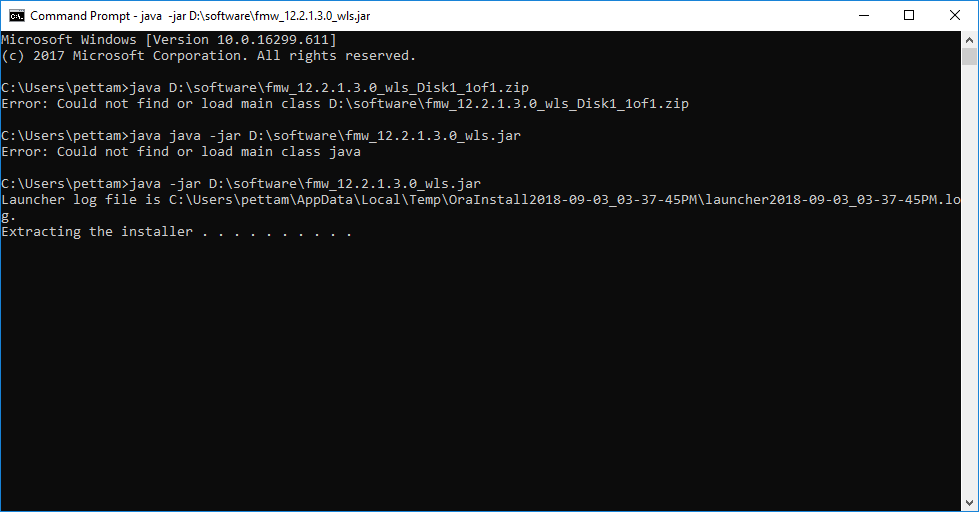
* 1. docker commit {Docker ID/Name} {optional another docker name}

e.g. docker commit AdminServer AdminServer2 (if you changed something and want to apply to another image)

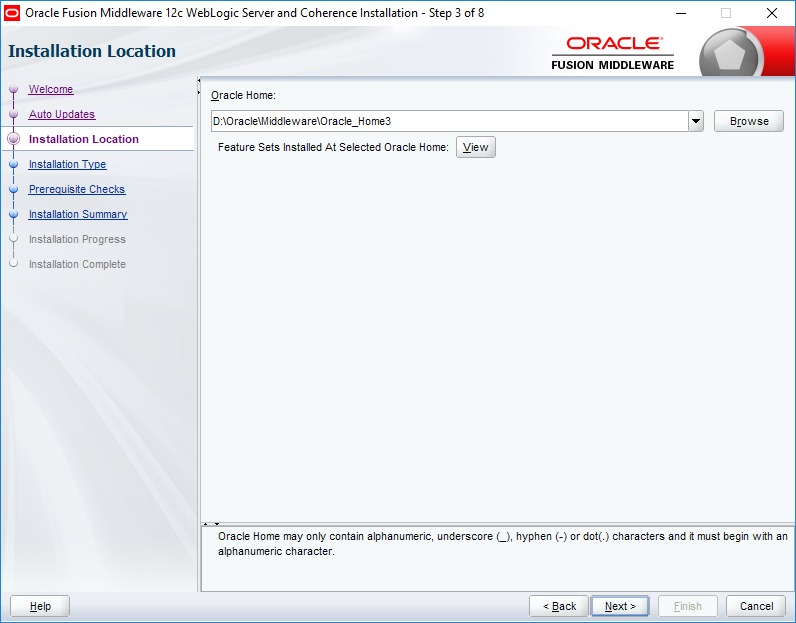
* 1. docker stats {Docker ID/Name}
  2. docker inspect --format='{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' {Docker ID/Name}

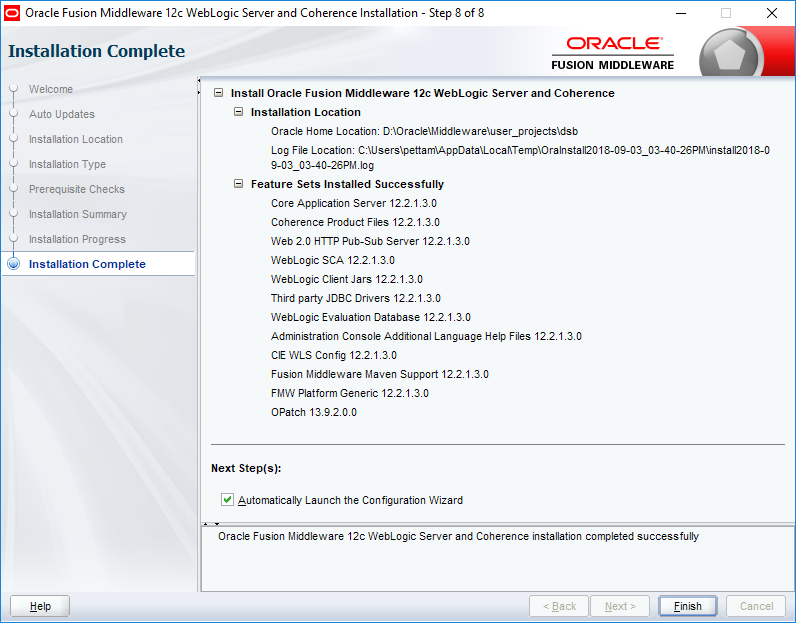
# Run Spring Boot App with Standalone WebLogic Server in STS IDE

1. Download the Oracle Weblogic
2. Install the Java JDK (ignore it if your local machine has JDK installed)
3. Open a cmd prompt with Administrator mode and install the Oracle Weblogic locally

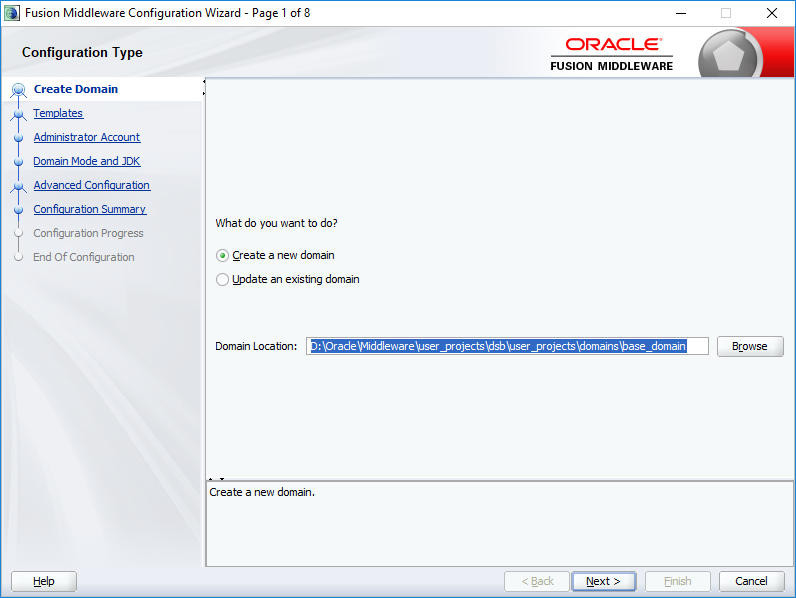


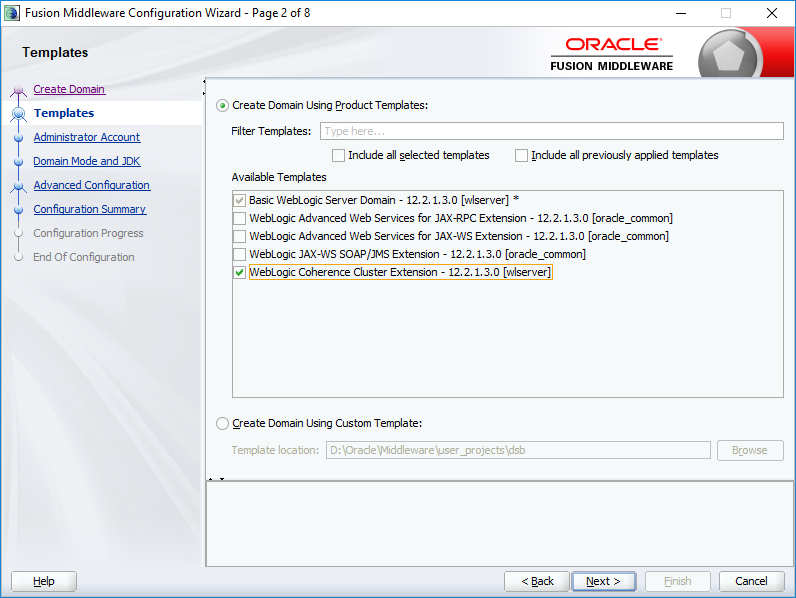
1. Follow the instruction to complete the installation of Oracle Weblogic

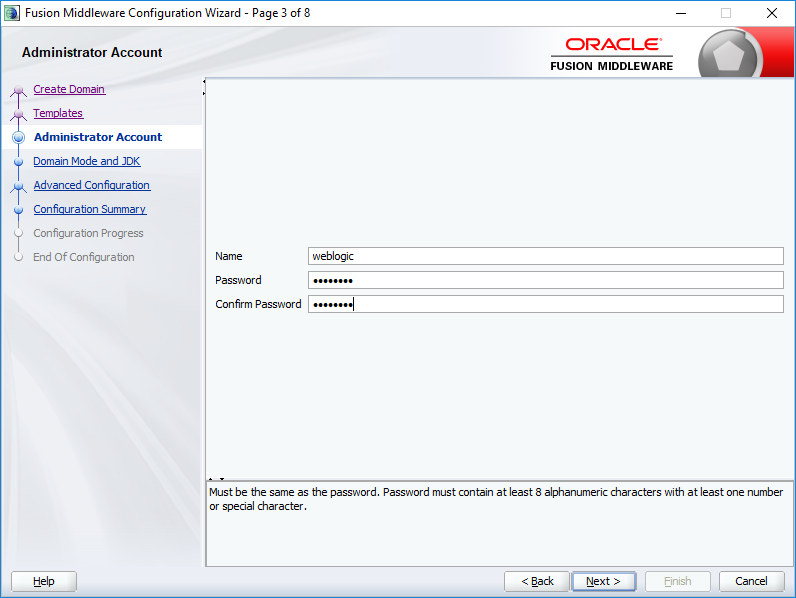


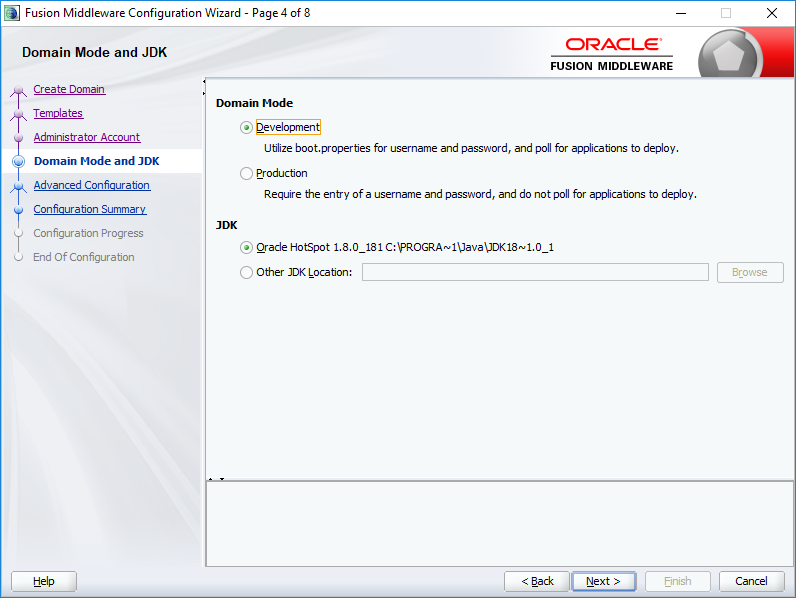


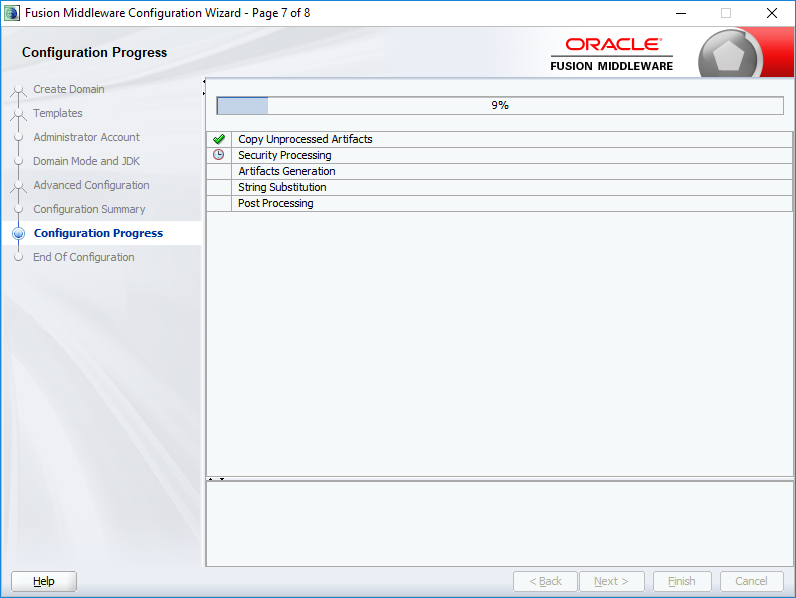
1. Configuration: Create a new domain



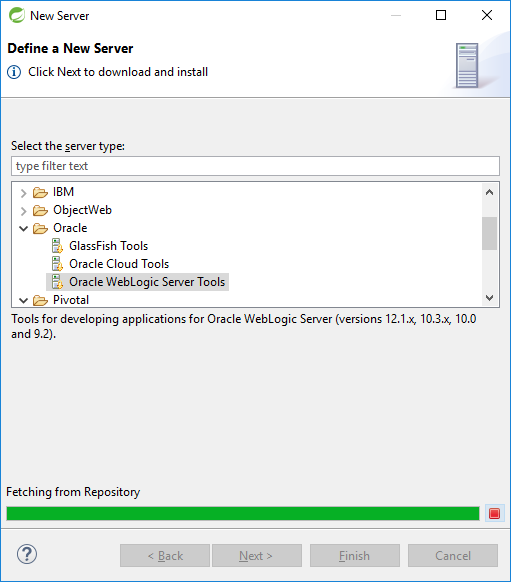


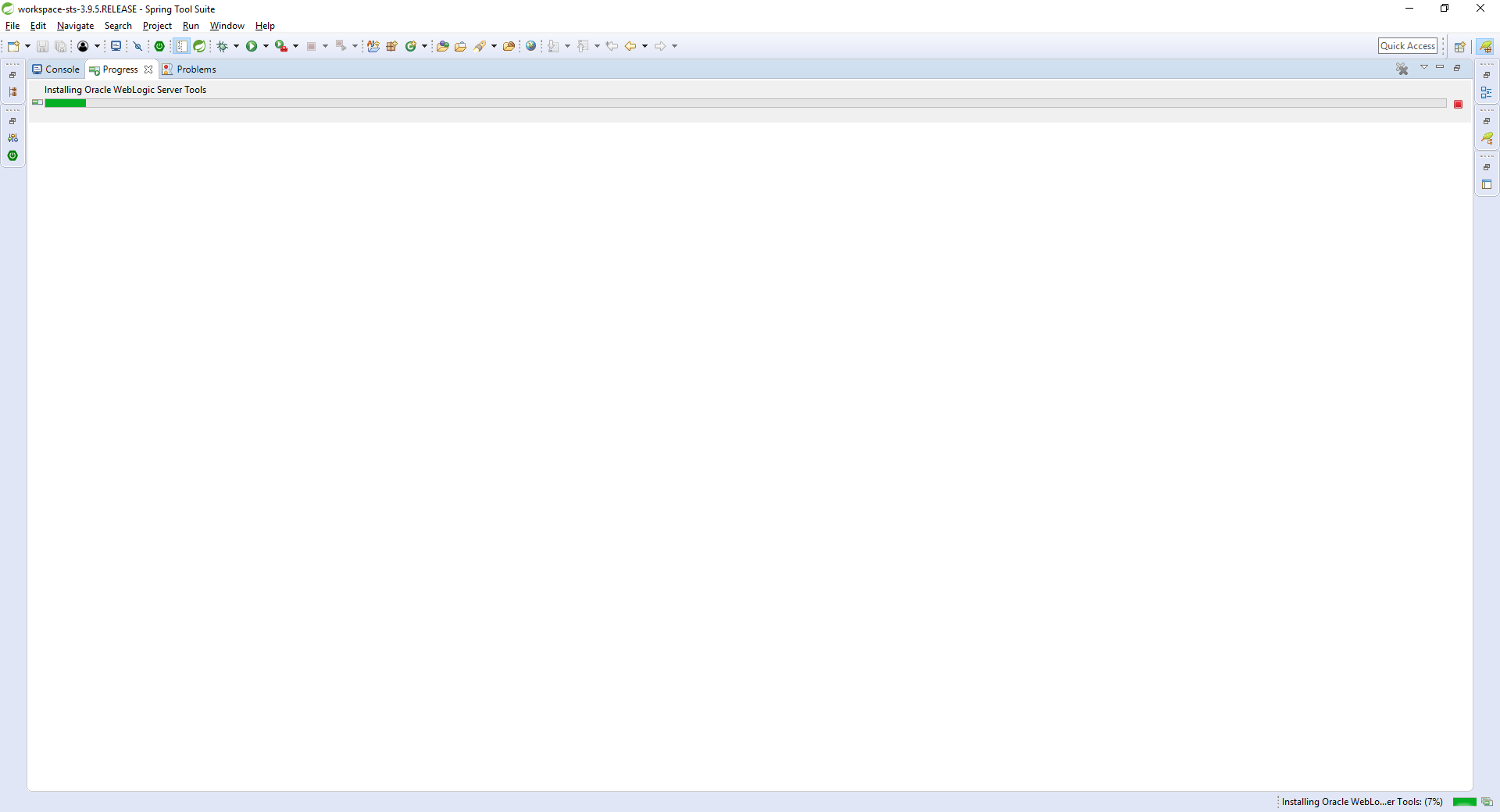




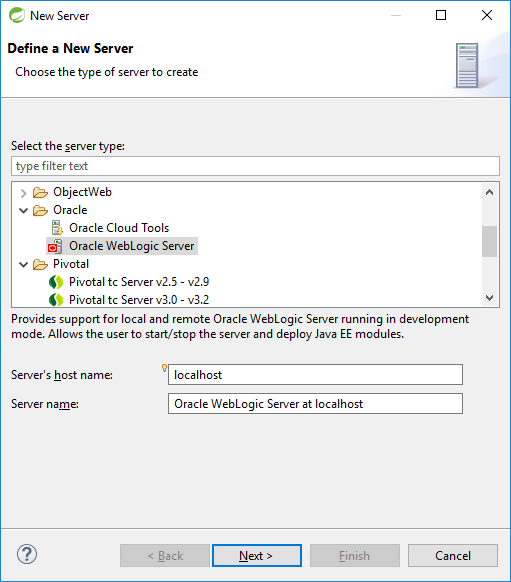


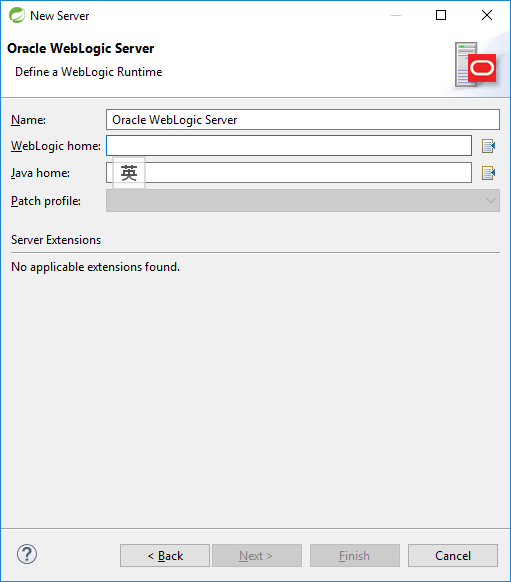
1. Open the “Servers” property and click the link to create a new server or right click on the property to create a new server
2. Download the Tools / Extension for Oracle WebLogic Server



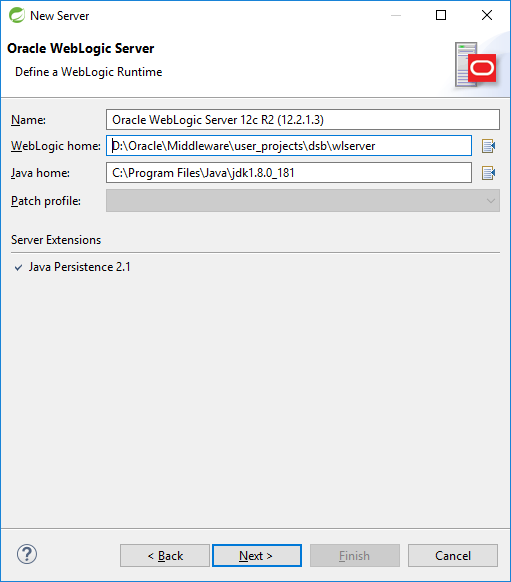


1. Define the New Server of Oracle WebLogic Server

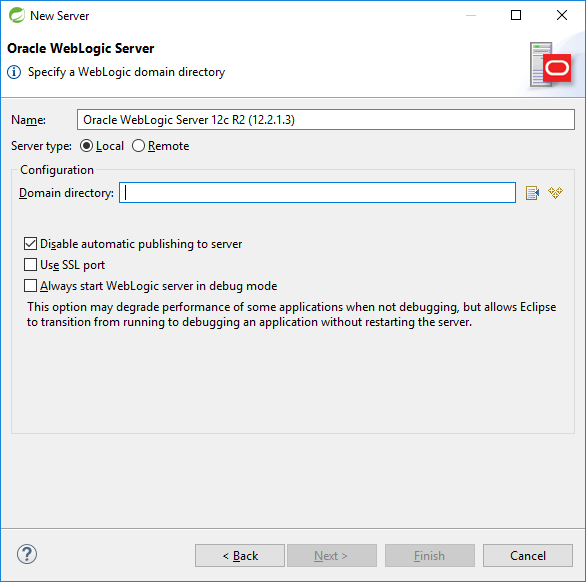


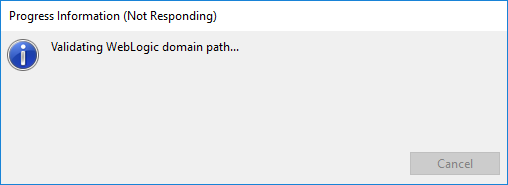


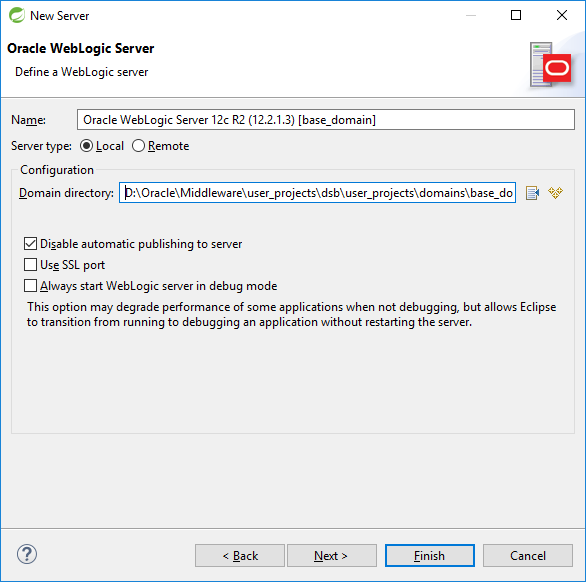
1. Provide the local WebLogic Home and Java Home and complete the setup

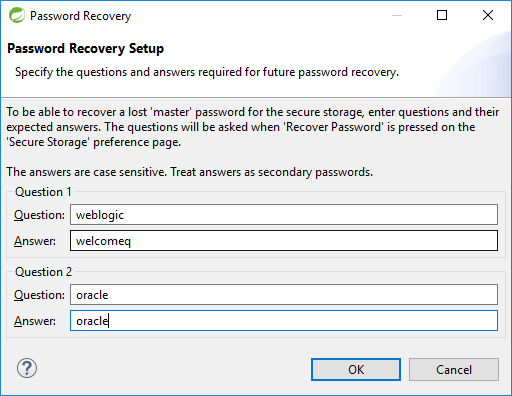


1. Specify the domain





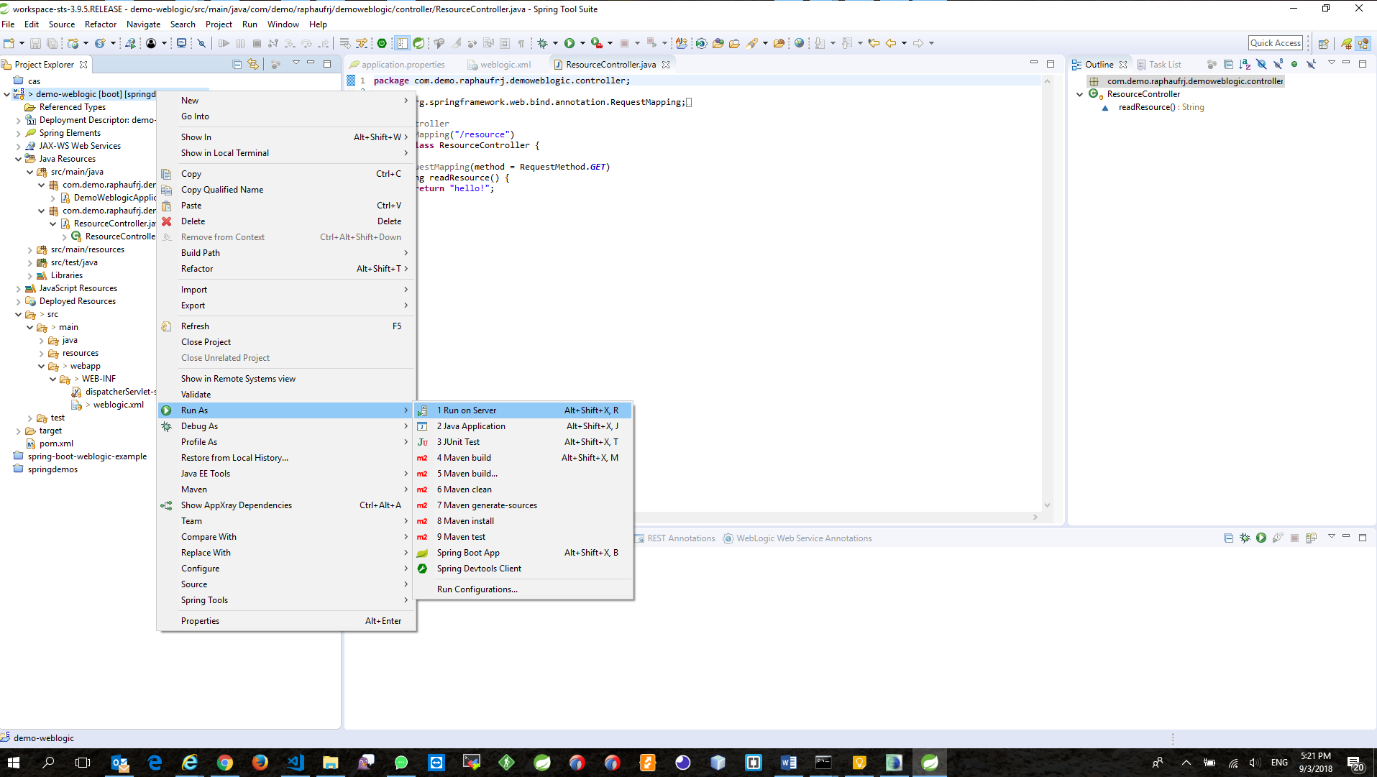




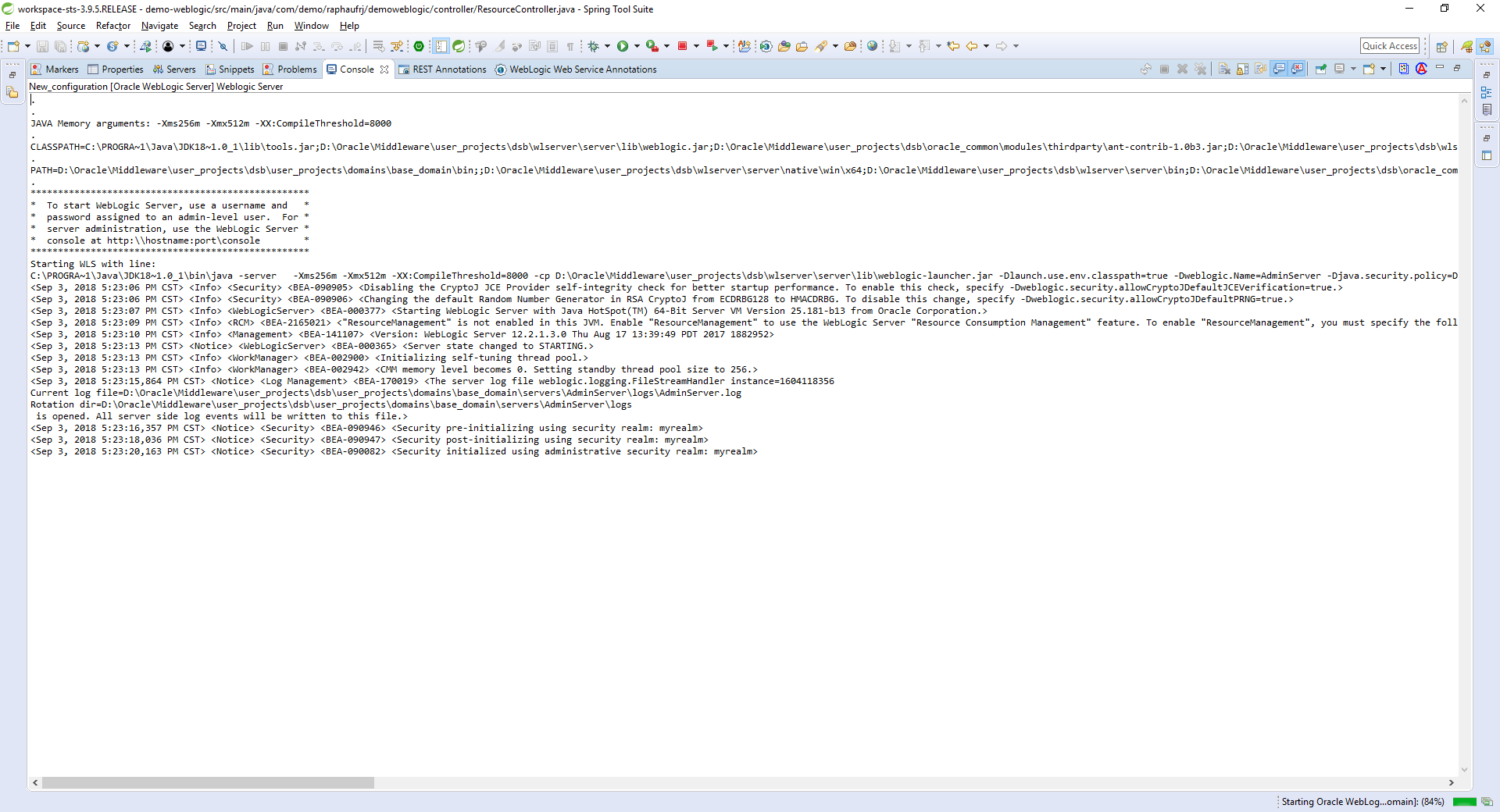
1. Sample Spring Boot application running on weblogic server

git clone https://github.com/saiips/springboot-weblogic.git

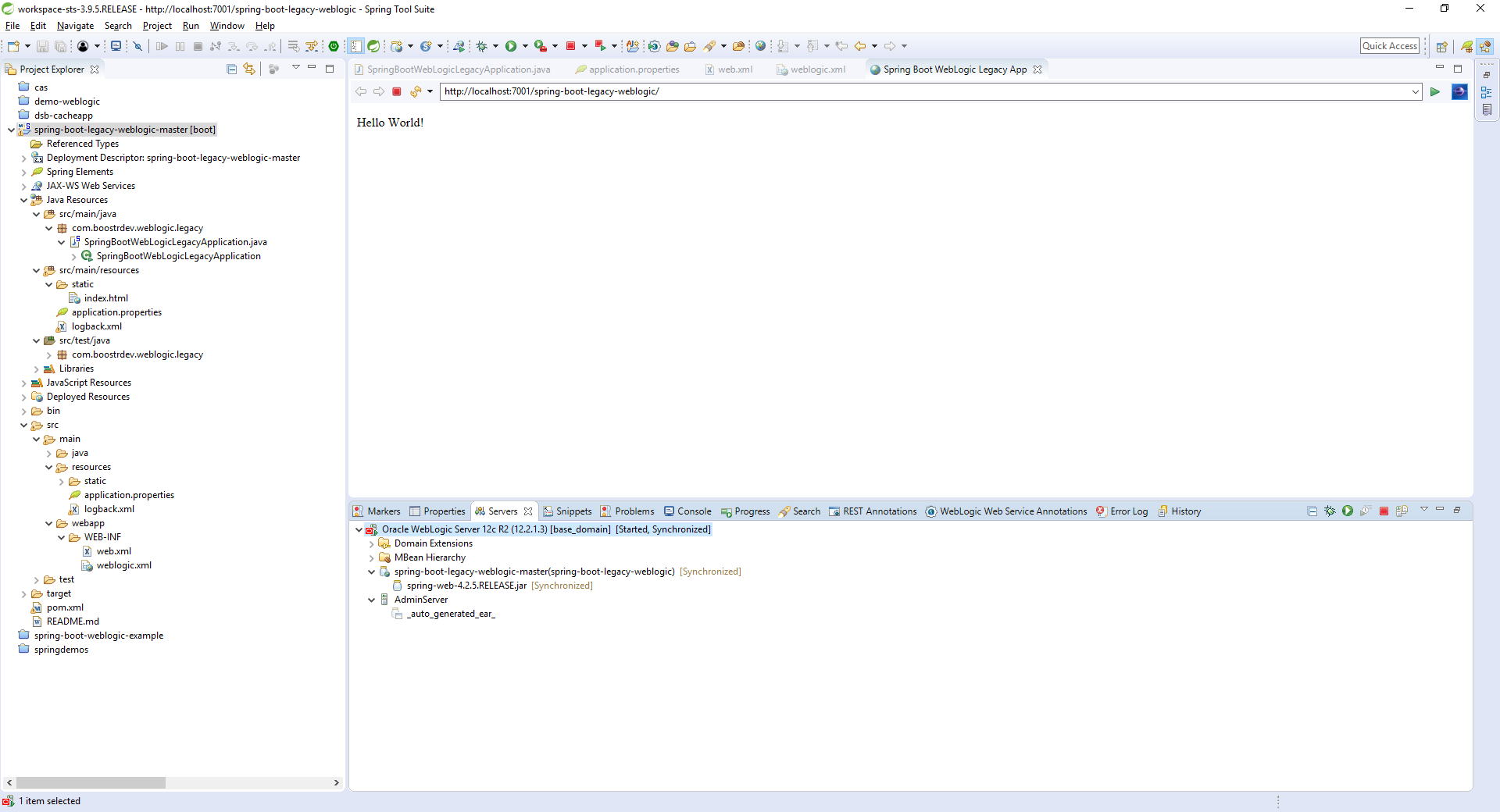
1. Press Alt+F5 to update the maven project or Run Maven Install
2. Right click and run as “Run on Server” (locally)



1. Follow the Instruction to deploy the apps to the standalone weblogic server



1. The app runs automatically if the deployment is successfully



# Reference of committing source to GitHub (New Repository)

pettam@PETTAM-HK MINGW64 /d/consulting/projects/DSB e-Banking Project/demo reference/spring-boot-legacy-weblogic-master

**$ git init**

Initialized empty Git repository in D:/consulting/projects/DSB e-Banking Project/demo reference/spring-boot-legacy-weblogic-master/.git/

pettam@PETTAM-HK MINGW64 /d/consulting/projects/DSB e-Banking Project/demo reference/spring-boot-legacy-weblogic-master (master)

**$ git add .**

warning: LF will be replaced by CRLF in .gitignore.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in README.md.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/.gitignore.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/README.md.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/pom.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/src/main/resources/application.properties.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/src/main/resources/logback.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/src/main/resources/static/index.html.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/src/main/webapp/WEB-INF/web.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in bin/src/main/webapp/WEB-INF/weblogic.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in pom.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in src/main/java/com/boostrdev/weblogic/legacy/SpringBootWebLogicLegacyApplication.java.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in src/main/resources/application.properties.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in src/main/resources/logback.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in src/main/resources/static/index.html.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in src/main/webapp/WEB-INF/web.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in src/main/webapp/WEB-INF/weblogic.xml.

The file will have its original line endings in your working directory.

warning: LF will be replaced by CRLF in src/test/java/com/boostrdev/weblogic/legacy/SpringBootWeblogicLegacyApplicationTests.java.

The file will have its original line endings in your working directory.

pettam@PETTAM-HK MINGW64 /d/consulting/projects/DSB e-Banking Project/demo reference/spring-boot-legacy-weblogic-master (master)

**$ git commit -m "First commit"**

[master (root-commit) ac8eb18] First commit

Committer: pettam <pettam@PETTAM-HK.hk.oracle.com>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly:

git config --global user.name "Your Name"

git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

20 files changed, 539 insertions(+)

create mode 100644 .gitignore

create mode 100644 README.md

create mode 100644 bin/.gitignore

create mode 100644 bin/README.md

create mode 100644 bin/pom.xml

create mode 100644 bin/src/main/java/com/boostrdev/weblogic/legacy/SpringBootWebLogicLegacyApplication.class

create mode 100644 bin/src/main/resources/application.properties

create mode 100644 bin/src/main/resources/logback.xml

create mode 100644 bin/src/main/resources/static/index.html

create mode 100644 bin/src/main/webapp/WEB-INF/web.xml

create mode 100644 bin/src/main/webapp/WEB-INF/weblogic.xml

create mode 100644 bin/src/test/java/com/boostrdev/weblogic/legacy/SpringBootWeblogicLegacyApplicationTests.class

create mode 100644 pom.xml

create mode 100644 src/main/java/com/boostrdev/weblogic/legacy/SpringBootWebLogicLegacyApplication.java

create mode 100644 src/main/resources/application.properties

create mode 100644 src/main/resources/logback.xml

create mode 100644 src/main/resources/static/index.html

create mode 100644 src/main/webapp/WEB-INF/web.xml

create mode 100644 src/main/webapp/WEB-INF/weblogic.xml

create mode 100644 src/test/java/com/boostrdev/weblogic/legacy/SpringBootWeblogicLegacyApplicationTests.java

pettam@PETTAM-HK MINGW64 /d/consulting/projects/DSB e-Banking Project/demo reference/spring-boot-legacy-weblogic-master (master)

**$ git remote add origin https://github.com/saiips/springboot-weblogic.git**

pettam@PETTAM-HK MINGW64 /d/consulting/projects/DSB e-Banking Project/demo reference/spring-boot-legacy-weblogic-master (master)

**$ git push origin master**

Enumerating objects: 48, done.

Counting objects: 100% (48/48), done.

Delta compression using up to 4 threads.

Compressing objects: 100% (27/27), done.

Writing objects: 100% (48/48), 7.18 KiB | 163.00 KiB/s, done.

Total 48 (delta 3), reused 0 (delta 0)

remote: Resolving deltas: 100% (3/3), done.

To https://github.com/saiips/springboot-weblogic.git

\* [new branch] master -> master

pettam@PETTAM-HK MINGW64 /d/consulting/projects/DSB e-Banking Project/demo reference/spring-boot-legacy-weblogic-master (master)

Add coherence.jar to maven repository

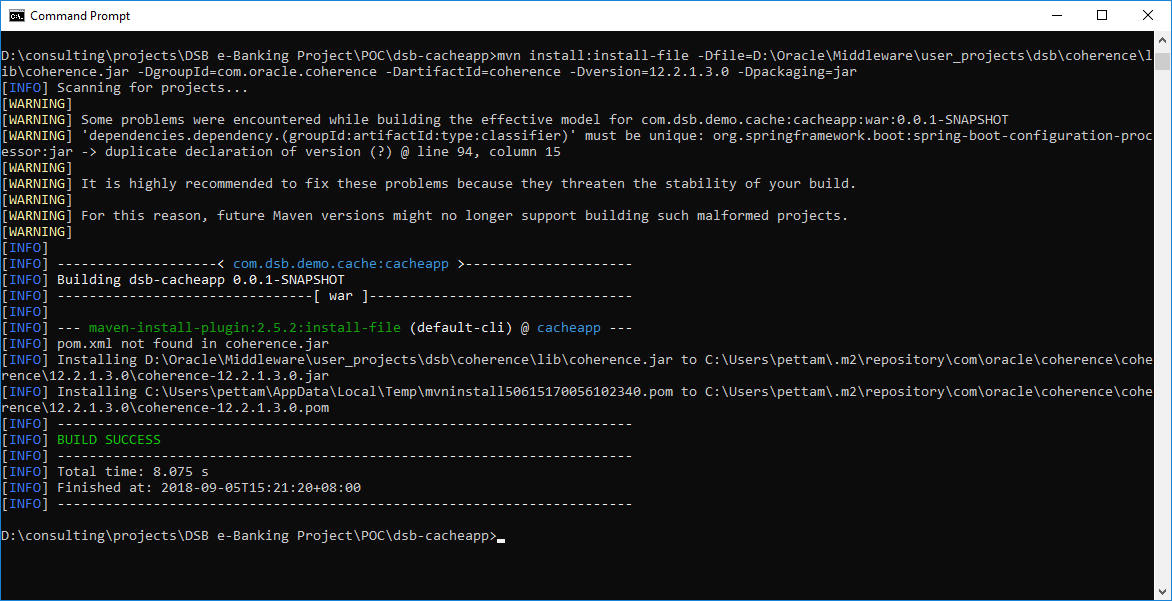
1. Install the Weblogic with Coherence version. You can find the coherence.jar at

\\ **<Domain>**\coherence\lib\coherence.jar

1. Run the command to add it to the maven repository

hvn install:install-file -Dfile=**<Domain Directory>**\coherence\lib\coherence.jar -DgroupId=com.oracle.coherence -DartifactId=coherence -Dversion=12.2.1.3.0 -Dpackaging=jar  
  
e.g.

mvn install:install-file -Dfile=**D:\Oracle\Middleware\user\_projects\dsb**\coherence\lib\coherence.jar -DgroupId=com.oracle.coherence -DartifactId=coherence -Dversion=12.2.1.3.0 -Dpackaging=jar



# Setup CAS + Cluster Environment

1. Alternatives: Download the CAS-SERVER Overlay for DSB directly

git clone <https://github.com/saiips/cas-server.git> cas-server

1. Please modify the configuration accordingly

1) Update the keystore file at your local drive

//cas-server/src/main/resources/application.properties

Server.ssl.key-store=file:c:/etc/cas/thekeystore

***\* And copy the keystore file to your specified path from***

***//cas-server/src/main/resources/etc/cas/thekeystore***

2) Export the thekeystore.crt from the keystore file thekeystore  
keytool -export -alias thekeystore -file thekeystore.crt -keystore thekeystore

***\* Alternatives, you can use the thekeystore.crt directly from //cas-server/src/main/resources/etc/cas/thekeystore.crt***

3) import to the JDK

Keytool -import -alias thekeystore -storepass changeit -file thekeystore.crt -keystore \\YOU JDK location path\cacerts

***\* For Example, keytool -import -alias thekeystore -storepass changeit -file thekeystore.crt -keystore “C:\Program Files\Java\jdk1.8XXXXX\jre\lib\security\cacerts”***

1. Reference:
   1. Download CAS Overlay from <https://github.com/apereo/cas/releases>

git clone https://github.com/apereo/cas-overlay-template.git cas-server

* 1. Alternatives: Generate a CAS 5.1.0 Overlay

https://casinitializr.herokuapp.com/