

**BUILD AND DEPLOY ORACLE
SERVICE BUS PROJECTS FROM
ORACLE DEVELOPER CLOUD
SERVICE TO ORACLE SOA
CLOUD SERVICE USING THE
ORACLE SERVICE BUS PLUG-IN**

Kishor Kumar

Contents

1. Introduction	2
2. Prerequisites	2
3. Patching the Oracle Service Bus Server Installation	2
4. Create a New Project	2
5. Clone the Project.....	5
6. Populate Local Maven Repository	7
7. Populate Oracle Developer Cloud Service Maven Repository.....	10
8. Build Your Project on Oracle Developer Cloud Service and Deploy it to Oracle SOA Cloud Service ..	12
9. Issues With Certificates While Executing a Build Job	21

1. Introduction

This document describes one of the ways to build your Oracle Service Bus projects from Oracle Developer Cloud Service and deploy from Oracle Developer Cloud Service to Oracle SOA Cloud Service.

2. Prerequisites

This document requires the following software be installed on the local machine:

1. Download Git from:
<https://git-scm.com/>
2. Download JDK 1.8 or later from:
<http://download.oracle.com/otn-pub/java/jdk/8u151tub12/e758a0de34e24606bca991d704f6dcbf/jdk-8u151-linux-x64.tar.gz>
3. Download Maven version 3 or later from:
<http://www-us.apache.org/dist/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz>
4. Install the Oracle Service Bus ship-home on the local machine. The next section describes how to patch the Oracle Middleware Home directory on the Oracle Service Bus server.

3. Patching the Oracle Service Bus Server Installation

The patch must be applied based on the Oracle Service Bus version that is installed. The following patch is provided for Oracle Service Bus release 12.2.1.2.0:

12.2.1.2.0 Patch: 22392646 (BLR Request #25104574)

Patch Name: `p22392646_122120_Generic.zip`

ARU Link to: http://aru.us.oracle.com:8080/ARU/ViewPatchRequest/process_form?aru=20785020

To install the patch, follow the instructions in [Patching Your Environment Using](#).

4. Create a New Project

Use Oracle Developer Cloud Service to create a new project.

1. Log in to Oracle Developer Cloud Service console and click **New Project**.

Project Calculator Demo x

Secure | https://developer.us2.oraclecloud.com/developer34301-orchwdsdtz14soa/#projects/calculator-demo

Apps Bookmarks Imported Personal ORACLE OSB Work IN YouTube Google Maps Prime Video WP! Wow! Sign in

ORACLE Developer Cloud Service wd-ops-soacs_vw@oracle.com

Calculator Demo Search Activities

developer34301

+ New Project

Member Favorites Owner All

Filter Projects

Calculator Demo
No description

DeveloperCloudServiceExamples
No description

devsoacs
devCS-SOACS.inten.repo.by.Vamsee

Reset Filters

RECENT

REPOSITORIES + New Repository

Filter Git Repositories

All Favorites Favorites First

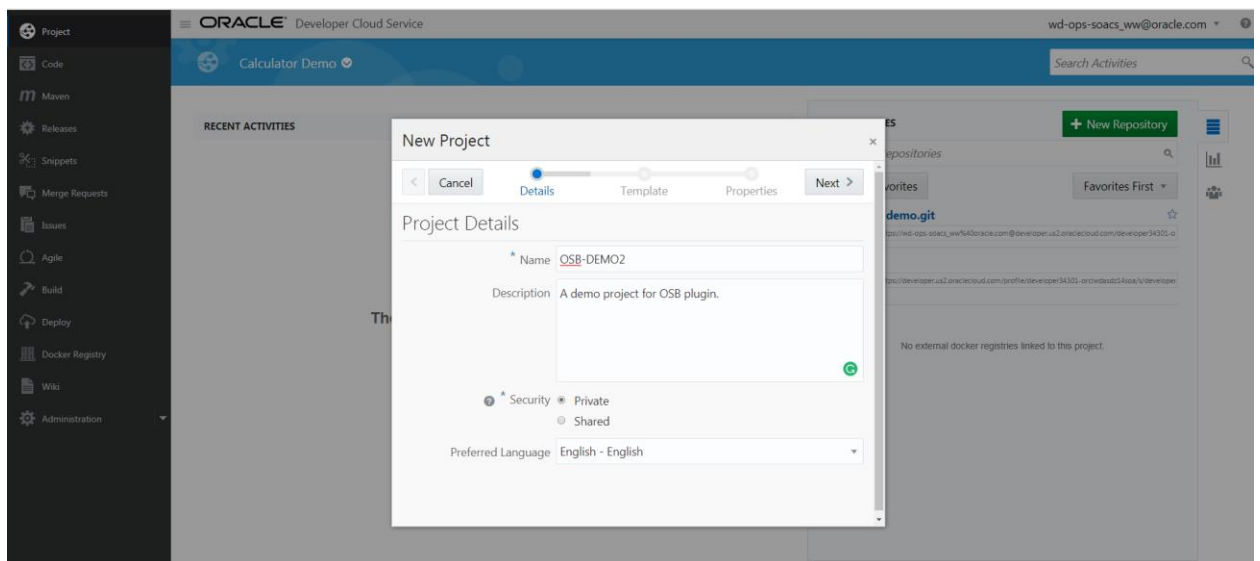
calculator-demo.git
HTTP SSH https://wd-ops-soacs_vw%40oracle.com@developer.us2.oraclecloud.com/developer34301-...

Maven
HTTP DAV https://developer.us2.oraclecloud.com/profile/developer34301-orchwdsdtz14soa/developer...

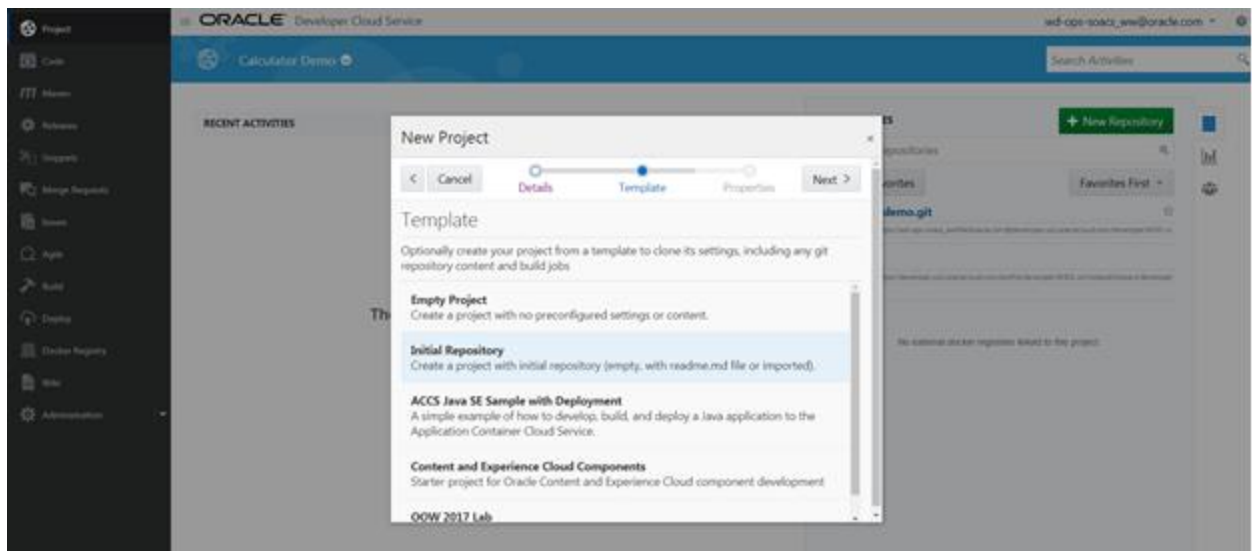
Docker
No external docker registries linked to this project.

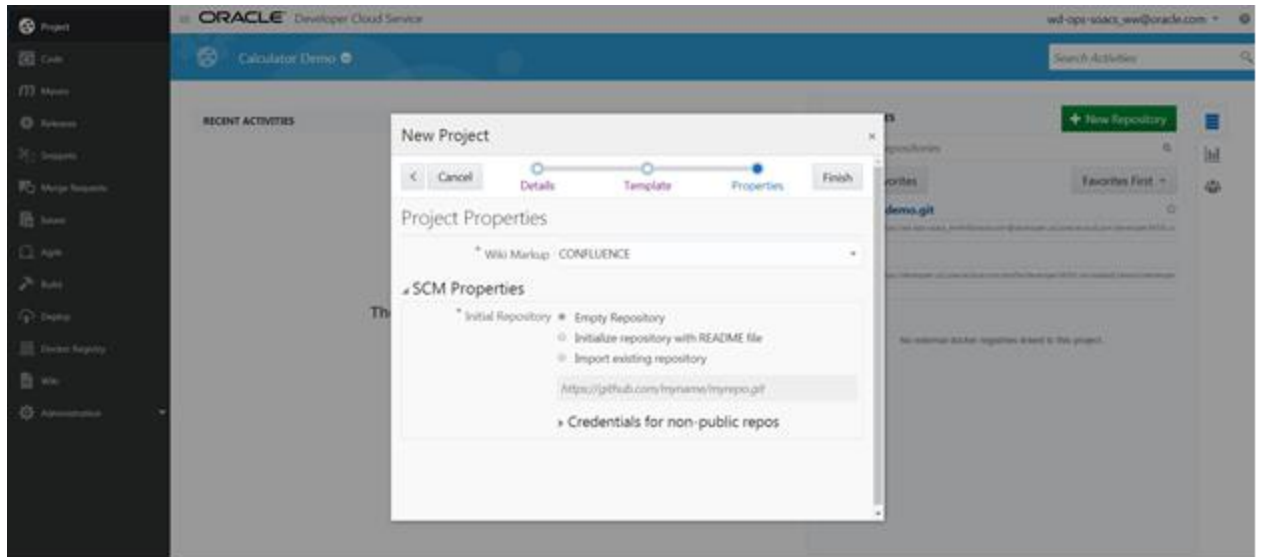
display.

2. Name the project "OSB-DEMO2". Click **Next**.



3. Choose the **Initial Repository** template. Click **Next**.





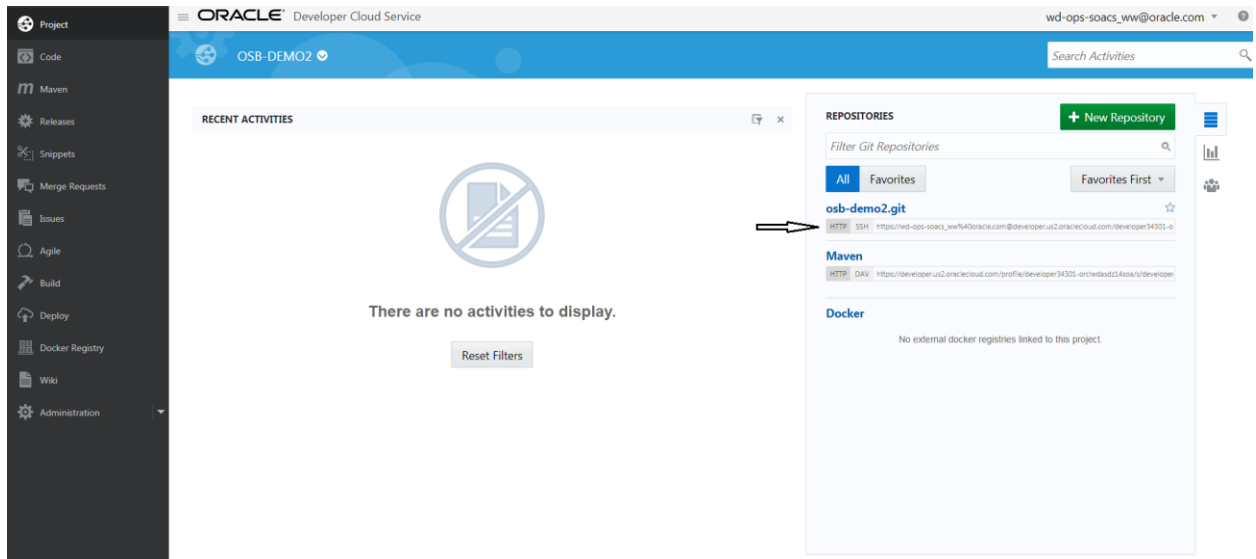
4. Click **Finish** and wait for Oracle Developer Cloud Service to set up the repository. When that is complete you will be at the projects landing page.

5. Clone the Project

After creating your project, the source code resides within Oracle Developer Cloud Service Git source repository. In order to work on it locally, you must download the contents of the remote Git repository into your local Git repository.

You can use SSH or HTTPS to clone the project you created in the previous step. The HTTPS method is shown here.

1. From the project's landing page in Oracle Developer Cloud Service console, find the Git URL.



- Clone the above mentioned repository into your local machine. Run the following command:

```
% git clone https://wd-ops-soacs\_ww%40oracle.com@developer.us2.oraclecloud.com/developer34301-orclwdasd14soa/s/developer34301-orclwdasd14soa\_osb-demo2\_21358/scm/osb-demo2.git
```

```
bash-4.1$ pwd
/scratch/kishork/DEVCS_to_SOACS
bash-4.1$ ls -ltr
total 4
drwxr-xr-x 4 kishork dba 4096 Oct 27 02:23 installation
bash-4.1$ git clone https://wd-ops-soacs_ww%40oracle.com@developer.us2.oraclecloud.com/developer34301-orclwdasd14soa/s/developer34301-orclwdasd14soa_osb-demo2_21358/scm/osb-demo2.git
```

Change the above URL to your Oracle Developer Cloud Service Project's repository URL.

You should see the following output after you run the clone command:

```
Initialized empty Git repository in
/scratch/kishork/DEVCS_to_SOACS/osb-demo2/.git/
remote: Counting objects: 211, done
remote: Finding sources: 100% (211/211)
remote: Getting sizes: 100% (87/87)
remote: Total 211 (delta 74), reused 208 (delta 74)
Receiving objects: 100% (211/211), 137.25 KiB, done.
Resolving deltas: 100% (74/74), done.
```

```
bash-4.1$ pwd
/scratch/kishork/DEVCS_to_SOACS
bash-4.1$ ls -ltr
total 4
drwxr-xr-x 4 kishork dba 4096 Oct 27 02:23 installation
bash-4.1$ git clone https://wd-ops-soacs_ww%40oracle.com@developer.us2.oraclecloud.com/developer34301-orclwdasd14soa/s/developer34301-orclwdasd14soa_osb-demo2_21358/scm/osb-demo2.git
Initialized empty Git repository in /scratch/kishork/DEVCS_to_SOACS/osb-demo2/.git/
remote: Counting objects: 211, done
remote: Finding sources: 100% (211/211)
remote: Getting sizes: 100% (87/87)
remote: Total 211 (delta 74), reused 208 (delta 74)
Receiving objects: 100% (211/211), 137.25 KiB, done.
Resolving deltas: 100% (74/74), done.
bash-4.1$
```

```

bash-4.1$ pwd
/scratch/kishork/DEVCS_to_SOACS
bash-4.1$ ls -ltr
total 4
drwxr-xr-x 4 kishork dba 4096 Oct 27 02:23 installation
bash-4.1$ git clone https://wd-ops-soacs.wv440oracle.com/developer.us2.oraclecloud.com/developer34381-orclwdsdz14soa/s/developer34381-orclwdsdz14soa_osb-demo2_21358/scm/osb-demo2.git
Initialized empty Git repository in /scratch/kishork/DEVCS_to_SOACS/osb-demo2/.git/
remote: Counting objects: 211, done
remote: Finding sources: 100% (211/211)
remote: Getting sizes: 100% (87/87)
remote: Total 211 (delta 74), reused 208 (delta 74)
Receiving objects: 100% (211/211), 137.25 KiB, done.
Resolving deltas: 100% (74/74), done.
bash-4.1$ ls
installation  osb-demo2
bash-4.1$ █

```

3. If you are behind a proxy, then you must tell Git about the proxy information using the following commands:


```
% git config --global http.proxy myproxy.mycompany.com : 80
```

```
[git config --global proxyUrl:proxyPort ]
```
4. The clone command creates a subdirectory called `osb-demo2` that acts as your local repository. Any change that you make in this repository must be pushed to the remote repository in Oracle Developer Cloud Service in order for the change to be reflected and be able to build and deploy from Oracle Developer Cloud Service.

6. Populate Local Maven Repository

There are multiple ways in which the local Maven repository can be populated. The Maven repository is the location that contains the JAR files required for Maven to compile and package your source code.

This section describes how to populate the local Maven repository from the `MW_HOME` directory of your Oracle Service Bus installation. This is the easiest way to populate your local Maven repository.

Use the Oracle Maven Synchronization Plug-in to synchronize the Maven repository with `MW_HOME`. This plug-in introspects `MW_HOME` and uploads any `pom.xml` files and referenced binaries into the repository. To perform this operation, follow these steps:

1. Use the following commands to install the Oracle Maven Synchronization Plug-in from your `MW_HOME` directory.

Notes: Your version may be different -- 12.2.1 is used here. Also note the directory change. You should be in a directory without a POM or with a bland POM otherwise Maven attempts to evaluate it:

```
% cd osb-demo2/
```

The following `echo` commands show how `MW_HOME`, `JAVA_HOME` and `M2_HOME` might be set before executing the `mvn` command:

```
% echo $MW_HOME
```

```

/scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS
% echo $JAVA_HOME
/scratch/kishork/DEVCS_to_SOACS/installation/jdk1.8.0_141/
% echo $M2_HOME
/scratch/kishork/DEVCS_to_SOACS/installation/apache-maven-3.5.0/

```

Here's the Maven command:

```

% mvn install:install-file
-DpomFile=${MW_HOME}/oracle_common/plugins/maven/com/oracle/
maven/oracle-maven-sync/12.2.1/oracle-maven-sync-12.2.1.pom
-Dfile=${MW_HOME}/oracle_common/plugins/maven/com/oracle/maven/
oracle-maven-sync/12.2.1/oracle-maven-sync-12.2.1.jar

```

```

bash-4.1$ ls
Installation osb-demo2
bash-4.1$ cd osb-demo2/
bash-4.1$ echo $MW_HOME
/scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS
bash-4.1$ echo $JAVA_HOME
/scratch/kishork/DEVCS_to_SOACS/installation/jdk1.8.0_141/
bash-4.1$ echo $M2_HOME
/scratch/kishork/DEVCS_to_SOACS/installation/apache-maven-3.5.0/
bash-4.1$ mvn install:install-file -DpomFile=${MW_HOME}/oracle_common/plugins/maven/com/oracle/maven/oracle-maven-sync/12.2.1/oracle-maven-sync-12.2.1.pom -Dfile=${MW_HOME}/oracle_common/pl
ugins/maven/com/oracle/maven/oracle-maven-sync/12.2.1/oracle-maven-sync-12.2.1.jar
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
[INFO]
[INFO] --- maven-install-plugin:2.4:install-file (default-cli) @ standalone-pom ---
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/oracle_common/plugins/maven/com/oracle/maven/oracle-maven-sync/12.2.1/oracle-maven-sync-12.2.1.jar to /scratch
/kishork/repository/com/oracle/maven/oracle-maven-sync/12.2.1-2-0/oracle-maven-sync-12.2.1-2-0.jar
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/oracle_common/plugins/maven/com/oracle/maven/oracle-maven-sync/12.2.1/oracle-maven-sync-12.2.1.pom to /scratch
/kishork/repository/com/oracle/maven/oracle-maven-sync/12.2.1-2-0/oracle-maven-sync-12.2.1-2-0.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.873 s
[INFO] Finished at: 2017-11-14T02:03:29-08:00
[INFO] Final Memory: 8M/217M
[INFO] -----
bash-4.1$

```

2. Once the plug-in is installed, run it with the Maven command.

Before you run the Maven command, check content of the <version> element in the file:

```

${MW_HOME}/oracle_common/plugins/maven/com/oracle/maven/oracle-
maven-sync/12.2.1/oracle-maven-sync-12.2.1.pom

```

If the content of the <version> element is “12.2.1-1-0”, then the Maven command is:

```

% mvn com.oracle.maven:oracle-maven-sync:12.2.1-1-0:push
-DoracleHome=${MW_HOME}

```

If the content of the <version> element is “12.2.1-2-0”, then the Maven command is:

```

% mvn com.oracle.maven:oracle-maven-sync:12.2.1-2-0:push
-DoracleHome=${MW_HOME}

```

```

[INFO] -----
[INFO]
[INFO] --- oracle-maven-sync:12.2.1-2-0:push (default-cli) @ standalone-pom ---
[INFO] -----
[INFO] ORACLE MAVEN SYNCHRONIZATION PLUGIN - PUSH
[INFO] -----
[INFO] Found 496 location files in /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/plugins/maven
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/server/lib/osgi.jar to /scratch/kishork/repository/com/oracle/weblogic/osgi/12.2.1-2-0/osgi-12.2.1-2-0.jar
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/plugins/maven/com/oracle/weblogic/osgi/12.2.1/osgi-12.2.1.pom to /scratch/kishork/repository/com/oracle/weblogic/osgi/12.2.1-2-0/osgi-12.2.1-2-0.pom
[INFO] Found 496 location files in /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/plugins/maven
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/server/lib/osgi.jar to /scratch/kishork/repository/com/oracle/weblogic/osgi/12.2.1-2-0/osgi-12.2.1-2-0.jar
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/plugins/maven/com/oracle/weblogic/osgi/12.2.1/osgi-12.2.1.pom to /scratch/kishork/repository/com/oracle/weblogic/osgi/12.2.1-2-0/osgi-12.2.1-2-0.pom
[INFO] Found 496 location files in /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/modules/javax.security.auth.message.jar to /scratch/kishork/repository/com/oracle/weblogic/javax.security.auth.message/12.2.1-2-0/javax.security.auth.message-12.2.1-2-0.jar
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/wlserver/plugins/maven/com/oracle/weblogic/javax.security.auth.message/12.2.1/javax.security.auth.message-12.2.1.pom to /scratch/kishork/repository/com/oracle/weblogic/javax.security.auth.message/12.2.1-2-0/javax.security.auth.message-12.2.1-2-0.pom

```

This command takes a while to run because it copies large JAR files from MW_HOME to the local Maven repository.

```

[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/OPatch/plugins/maven/com/oracle/oracle.glm.opatchauto.core.oracle.glm.opatchauto.core.classpath/13.9.1/oracle.glm.opatchauto.core.oracle.glm.opatchauto.core.classpath-13.9.1.pom to /scratch/kishork/repository/com/oracle/oracle.glm.opatchauto.core.oracle.glm.opatchauto.core.classpath/13.9.1-0-0/oracle.glm.opatchauto.core.oracle.glm.opatchauto.core.classpath-13.9.1-0-0.pom
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/OPatch/plugins/maven/com/oracle/oracle.glm.opatchauto.core/13.9.1-0-0/oracle.glm.opatchauto.core-13.9.1-0-0.pom to /scratch/kishork/repository/com/oracle/oracle.glm.opatchauto.core/13.9.1-0-0/oracle.glm.opatchauto.core-13.9.1-0-0.pom
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/OPatch/plugins/maven/com/oracle/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath/13.9.1-0-0/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath-13.9.1-0-0.pom to /scratch/kishork/repository/com/oracle/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath/13.9.1-0-0/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath-13.9.1-0-0.pom
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/OPatch/plugins/maven/com/oracle/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath/13.9.1-0-0/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath-13.9.1-0-0.pom to /scratch/kishork/repository/com/oracle/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath/13.9.1-0-0/oracle.glm.opatch.common.api.oracle.glm.opatch.common.api.classpath-13.9.1-0-0.pom
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/OPatch/plugins/maven/com/oracle/oracle.glm.opatchauto.fmw.actions.classpath/13.9.1-0-0/oracle.glm.opatchauto.fmw.actions.classpath-13.9.1-0-0.pom to /scratch/kishork/repository/com/oracle/oracle.glm.opatchauto.fmw.actions.classpath/13.9.1-0-0/oracle.glm.opatchauto.fmw.actions.classpath-13.9.1-0-0.pom
[INFO] Installing /scratch/kishork/Oracle/Middleware/Oracle_Home_DEVCS_2_SOACS/OPatch/plugins/maven/com/oracle/oracle.glm.opatchauto.fmw.actions.classpath/13.9.1-0-0/oracle.glm.opatchauto.fmw.actions.classpath-13.9.1-0-0.pom to /scratch/kishork/repository/com/oracle/oracle.glm.opatchauto.fmw.actions.classpath/13.9.1-0-0/oracle.glm.opatchauto.fmw.actions.classpath-13.9.1-0-0.pom
[INFO] SUMMARY
[INFO] -----
[INFO] PUSH SUMMARY - ARTIFACTS PROCESSED SUCCESSFULLY
[INFO] -----
[INFO] Number of artifacts pushed: 2250
[INFO] -----
[INFO] No issues encountered.
[INFO] -----
[INFO] IMPORTANT NOTE
[INFO] This operation may have added/updated archetypes in your repository.
[INFO] To update your archetype catalog, you should run:
[INFO] 'mvn archetype:crawl -Dcatalog=$HOME/.m2/archetype-catalog.xml'
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:47 min
[INFO] Finished at: 2017-11-14T02:17:57-08:00
[INFO] Final Memory: 9M/228M
[INFO] -----
bash-4.1$

```

- Once the sync completes, you should be able to run the following commands to create a service bus archive on the local machine:

```

% cd /scratch/kishork/DEVCS_to_SOACS/osb-demo2/test-application
% mvn package -DoracleHome=$MW_HOME

```

```

bash-4.1$ pwd
/scratch/kishork/DEVCS_to_SOACS/osb-demo2/test-application
bash-4.1$ ls -ltr
total 20
-rw-r--r-- 1 kishork dba 7458 Nov 7 07:25 pom.xml
drwxr-xr-x 2 kishork dba 4096 Nov 13 00:37 certificates
drwxr-xr-x 3 kishork dba 4096 Nov 13 02:05 System
drwxr-xr-x 4 kishork dba 4096 Nov 13 02:17 default
bash-4.1$

```

```

drwxr-xr-x 4 kishork dba 4096 Nov 13 02:17 default
bash-4.1$ mvn package -DoracleHome=$PWD_HOME
[INFO] Scanning for projects...
[WARNING] Some problems were encountered while building the effective model for test-application:default:sbar:1.0
[WARNING] 'parent.relativePath' of POM test-application:default:1.0 (/scratch/kishork/DEVCS-to_SOACS/osb-demo2/test-application/default/pom.xml) points at test-application:app instead of co
m.oracle.servicebus:sbar-project-common, please verify your project structure @ line 13, column 13
[WARNING] Some problems were encountered while building the effective model for test-application:System:sbar:1.0
[WARNING] 'parent.relativePath' of POM test-application:System:1.0 (/scratch/kishork/DEVCS-to_SOACS/osb-demo2/test-application/System/pom.xml) points at test-application:app instead of com.
oracle.servicebus:sbar-system-common, please verify your project structure @ line 13, column 13
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO] -----
[INFO] Reactor Build Order:
[INFO]
[INFO] System
[INFO] default
[INFO] app
[INFO] -----
[INFO] Building System 1.0
[INFO] --- oracle-servicebus-plugin:12.2.1-2-0:package (default-package) @ System ---
[INFO] -----
[INFO] Building default 1.0
[INFO] --- oracle-servicebus-plugin:12.2.1-2-0:package (default-package) @ default ---
[INFO] -----
[INFO] Building app 1.0
[INFO] -----
[INFO] Reactor Summary:
[INFO] System ..... SUCCESS [ 43.576 s]
[INFO] default ..... SUCCESS [ 13.937 s]
[INFO] app ..... SUCCESS [ 0.003 s]
[INFO] BUILD SUCCESS

```

The next section describes how you can populate the remote Maven repository. The Oracle Developer Cloud Service remote Maven repository has to be populated only once; you can then use the Oracle Developer Cloud Service remote repository to download the JAR files required to build and deploy on local machines.

7. Populate Oracle Developer Cloud Service Maven Repository

There are multiple ways to populate the Oracle Developer Cloud Service remote Maven repository, this example uses a previously installed Oracle Maven Synchronization Plug-in to populate the Oracle Developer Cloud Service remote Maven repository.

To use the Oracle Maven Synchronization Plug-in with remote servers, first define a <repository> and <server> tag in settings.xml. Include a <server> tag in the Maven settings.xml file. An example is shown here:

```

<server>
  <id>MyMavenRepository</id>
  <username>clara.coder</username>
  <password>{qV6wv5700ZfrKjzSpXxrlNgJ}</password>
</server>

```

You may enter the password as it is or encrypt it using Maven encryption. The recommended approach is to use Maven to encrypt it.

Add a profile in Maven's settings.xml file. This profile is used to populate the Oracle Developer Cloud Service remote Maven repository. A sample of the profile is given here:

```

<profile>
  <id>RemoteMavenRepository</id>
  <activation>
    <activeByDefault>>false</activeByDefault>

```

```

</activation>

<repositories>
  <repository>
    <id>MyMavenRepository</id>
    <name>Remote Maven Repository for your DevCS project</name>
    <!-- CHANGE ME! -->
    <url>URL to your ORACLE DEVELOPER CLOUD SERVICE Maven
    Repository. This can be obtained from Oracle Developer Cloud
    Service Landing Page</url>
    <layout>default</layout>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>MyMavenRepository</id>
    <name>Remote Maven Plugin Repository for your DevCS
    project</name>
    <!-- CHANGE ME! -->
    <url>URL to your ORACLE DEVELOPER CLOUD SERVICE Maven
    Repository. This can be obtained from Oracle Developer
    Cloud Service Landing Page</url>
    <layout>default</layout>
  </pluginRepository>
</pluginRepositories>
</profile>

```

Note that the `<id>` used in `<repository>` and `<pluginRepository>` should match that of the `<id>` used in the `<server>` tag.

Run the Oracle Maven Synchronization Plug-in again using the profile described above:

```

% mvn -s /scratch/daxia/osb/settings.xml -P RemoteMavenRepository
com.oracle.maven:oracle-maven-sync:12.2.1-2-0:push -
DoracleHome=${MW_HOME}
-DserverId=MyMavenRepository

```

Notes:

- The `serverId` you specify should match the one you specify in `settings.xml`.
- This command might take many hours to complete.

The Oracle Developer Cloud Service Maven repository can be accessed without a proxy if your company's network policy permits. Accessing Oracle Developer Cloud Service without using a proxy can enhance performance while populating the Maven remote repository. To access the repository without

a proxy, add the Oracle Developer Cloud Service Maven repository URL to `<nonProxyHosts>` in `settings.xml`. For example:

```
<proxy>
  <id>optional</id>
  <active>true</active>
  <protocol>http</protocol>
  <host>YOUR_ORGANIZATION_PROXY_HOST</host>
  <port>PROXY_PORT</port>
  <nonProxyHosts>DEV_CS_MAVEN_REPOSITORY_URL</nonProxyHosts>
</proxy>
```

Any change that you make on your local machine must be pushed to the remote repository in Oracle Developer Cloud Service in order for the change to be reflected and for it to be built and deployed from Oracle Developer Cloud Service.

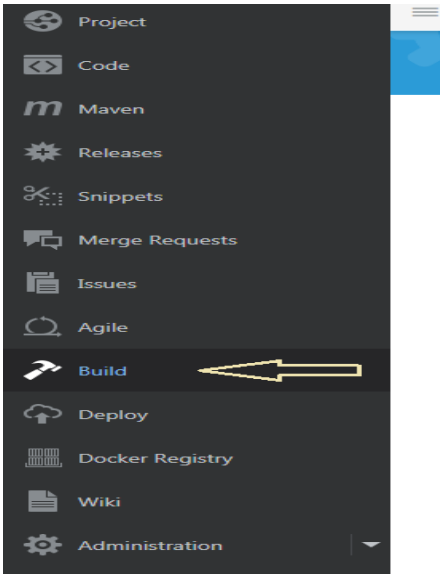
Use the following commands to push your local project changes to Oracle Developer Cloud Service project:

```
git commit -a    (provide your comment in popup window)
git push -u origin branch_specifier
```

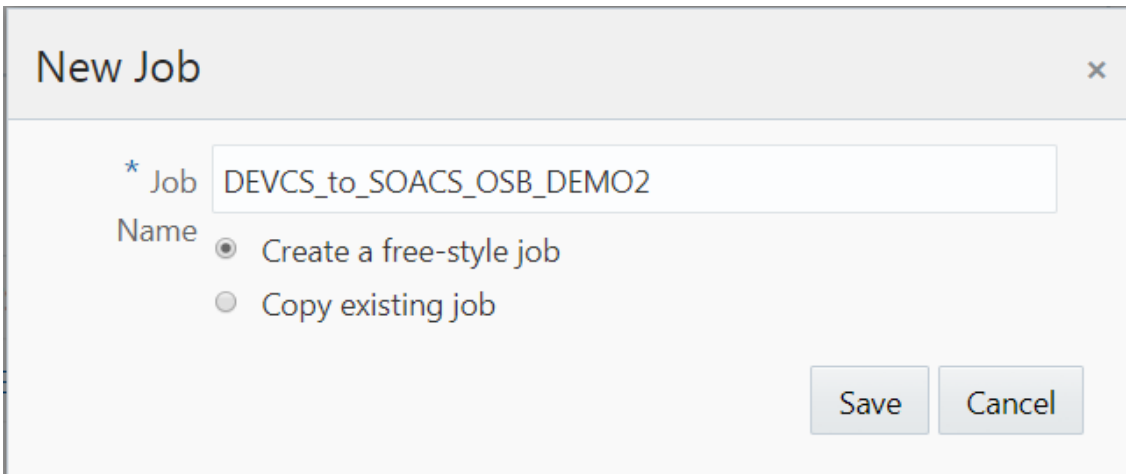
8. Build Your Project on Oracle Developer Cloud Service and Deploy it to Oracle SOA Cloud Service

This step requires you to use Hudson to configure a build job. The following steps guide you thru this configuration.

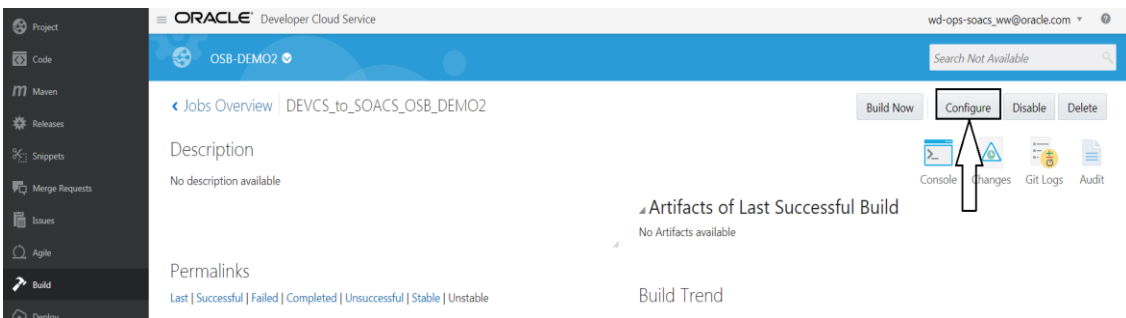
1. From the Oracle Developer Cloud Service landing page, navigate to the **Build** tab



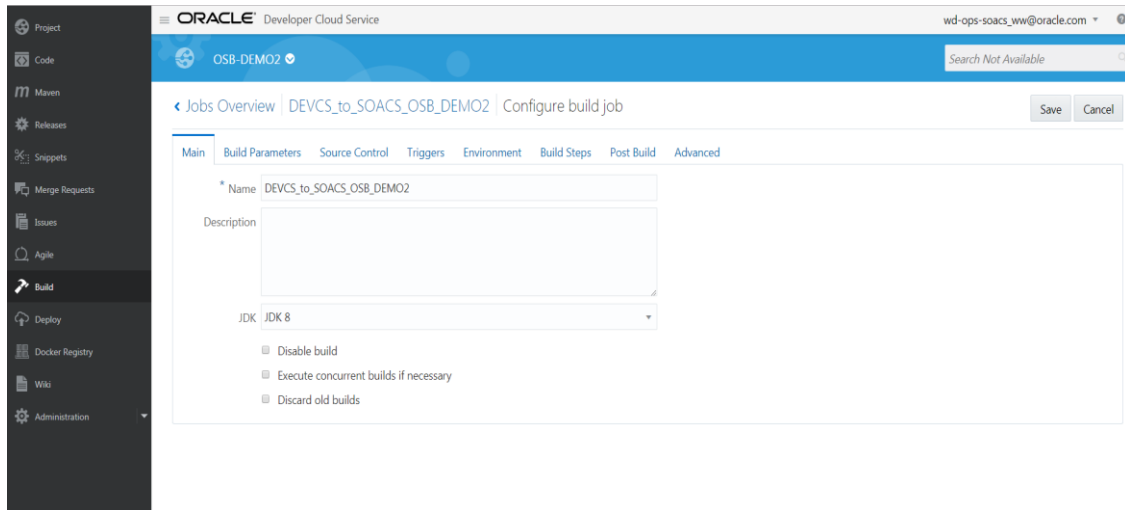
2. Click **New Job**. Name the new job. For example, “DEVCS_to_SOACS_OSB_DEMO2”. Click **Save**.



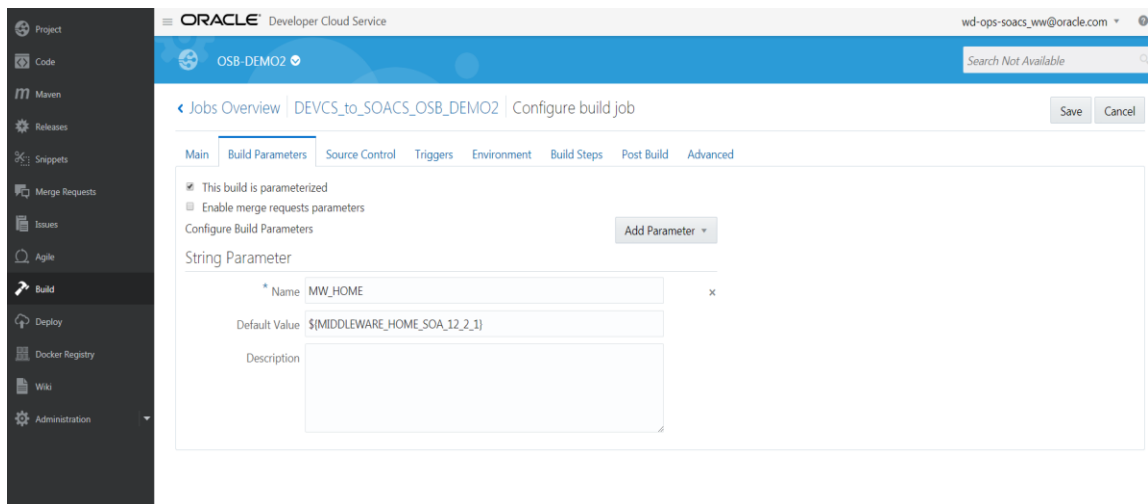
3. Click **Configure**.



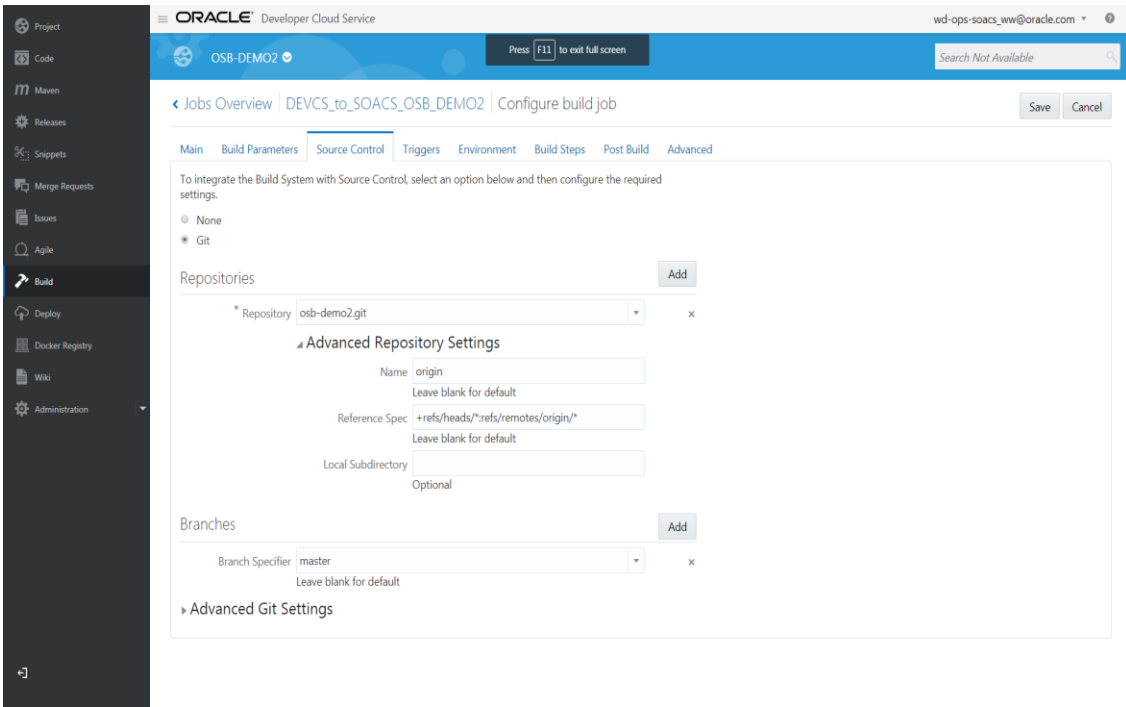
4. In the **Main** tab select **JDK 8** from the **JDK** dropdown menu.



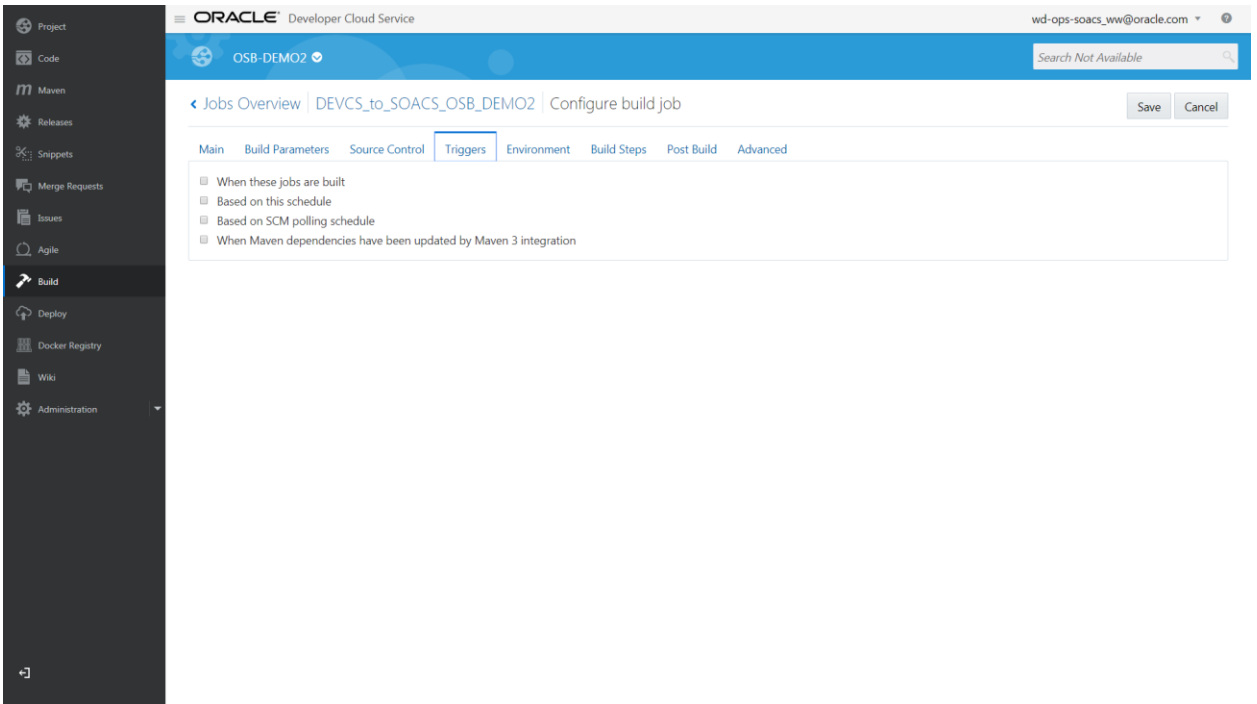
In the **Build Parameters** tab, click **Add Parameter**, and choose **String Parameter**. Enter **MW_HOME** as the name and **\${MIDDLEWARE_HOME_SOA_12_2_1}** as the value. This value is required by `pom.xml`.



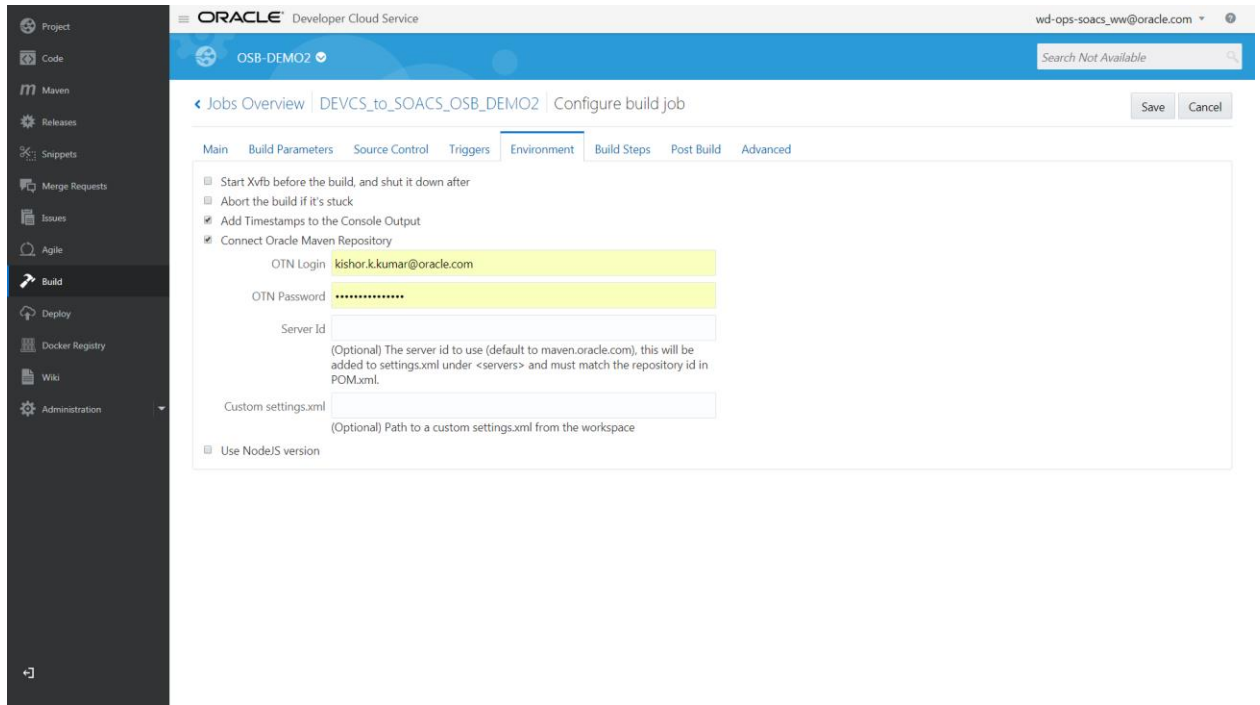
5. In the **Source Control** tab, click **Git** and choose the only repository in the drop down menu. There should only be one branch called `master`.



6. Leave the default settings in the **Triggers** tab. You will manually trigger this job for the time being.



7. In the **Environment** tab, click the **Connect Oracle Maven Repository** check box and enter your OTN login and OTN Password.



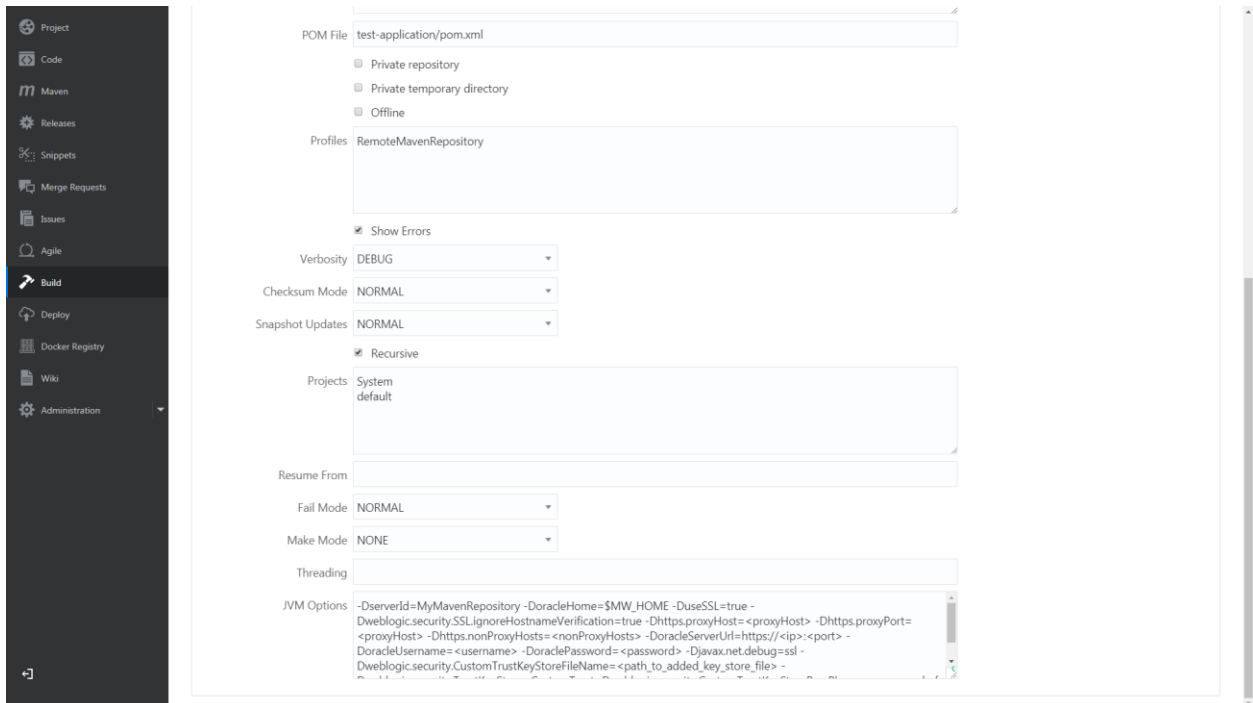
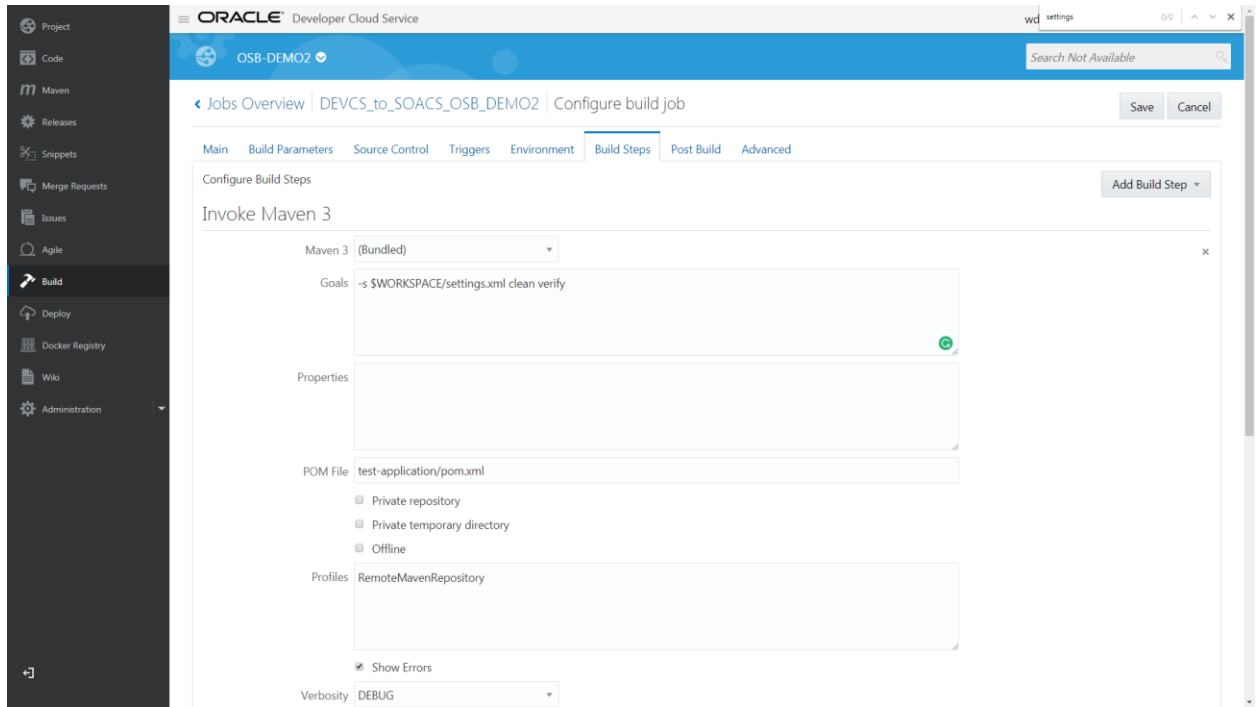
8. You can use the `$WORKSPACE` environment variable to access the workspace in order to navigate and use to any of the files or directories inside the project. In this example the project name is "OSB-DEMO2".

For example if you can use:

```
$WORKSPACE/settings.xml
```

in the **Goals** field in the **Build Steps** tab instead of:

```
/home/c2c/hudson/workspace/developer34301-orclwdasd14soa_osb-  
demo2_21358.DEVCS_to_SOACS_OSB_DEMO2
```

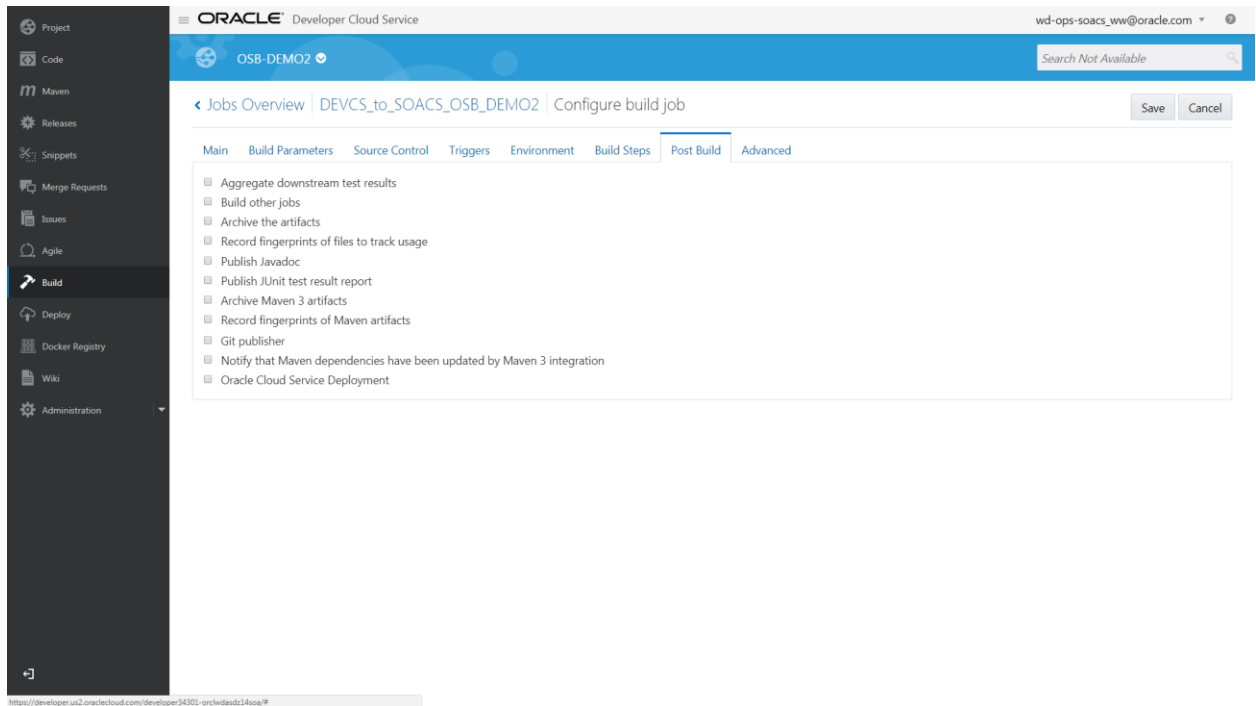


The JVM option:

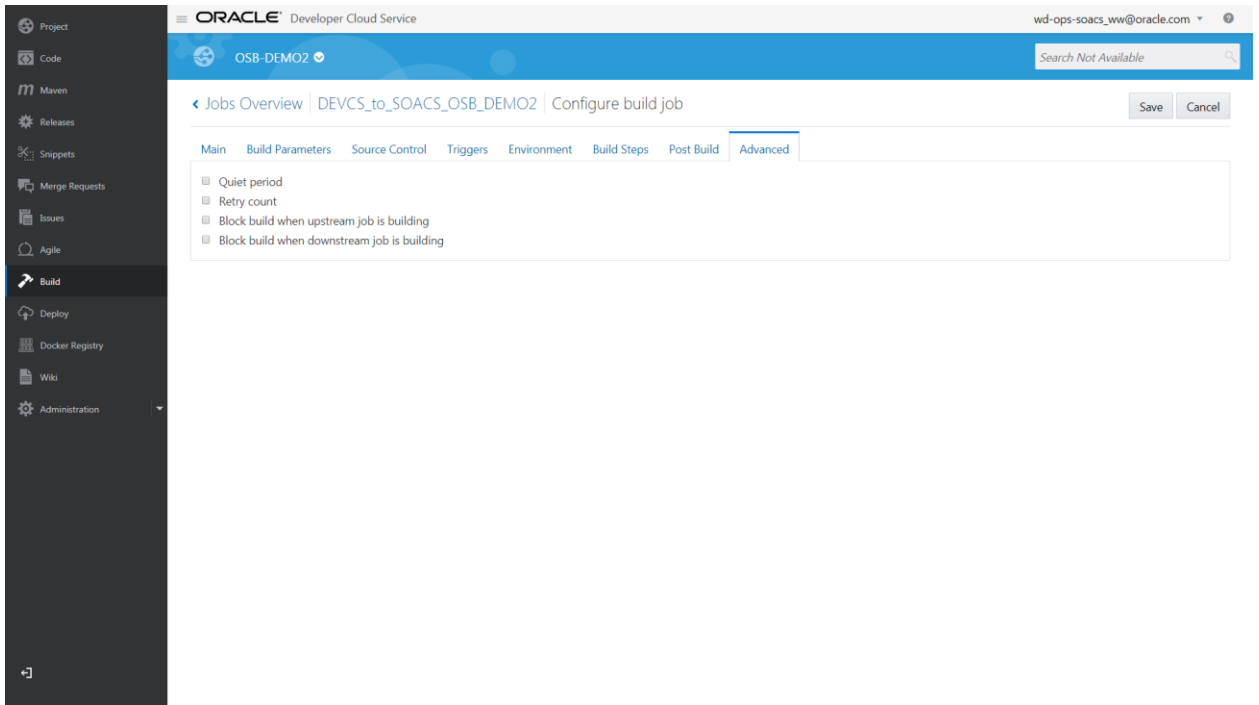
```
-DserverId=MyMavenRepository -DoracleHome=$MW_HOME -DuseSSL=true
-Dweblogic.security.SSL.ignoreHostnameVerification=true
-Dhttps.proxyHost=<proxyHost> -Dhttps.proxyPort=<proxyPort>
```

```
-Dhttps.nonProxyHosts=<nonProxyHosts>
-DoracleServerUrl=https://<ip>:<port>
-DoracleUsername=<username> -DoraclePassword=<password>
-Djavax.net.debug=ssl -
Dweblogic.security.CustomTrustKeyStoreFileName=<path_to_added_key
_store_file> -Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStorePassPhrase=<password of
keystore> -Dweblogic.security.CustomTrustKeyStoreType=jks
```

9. Leave the default settings in the **Post Build** tab.

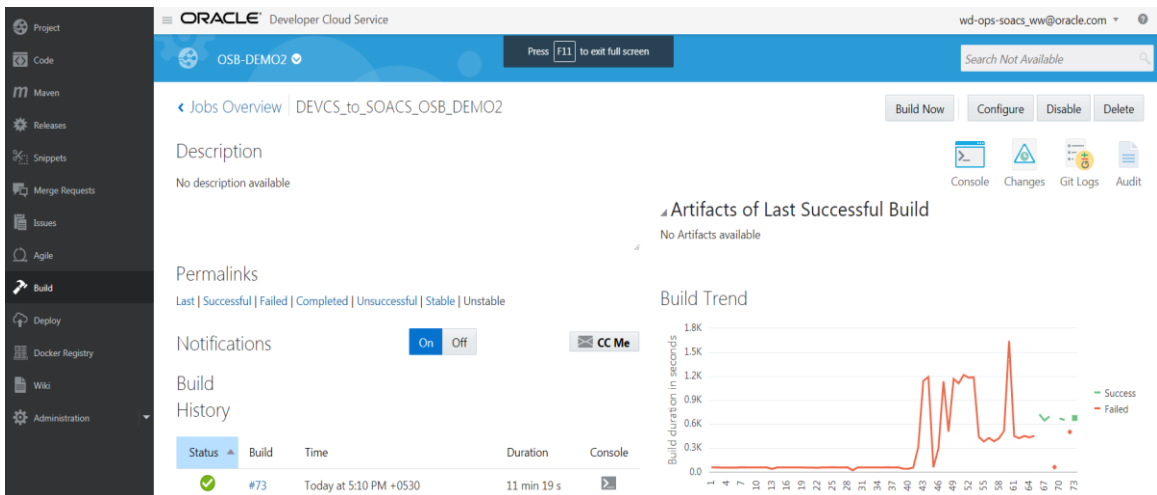


10. Leave the default settings in the **Advanced** tab.

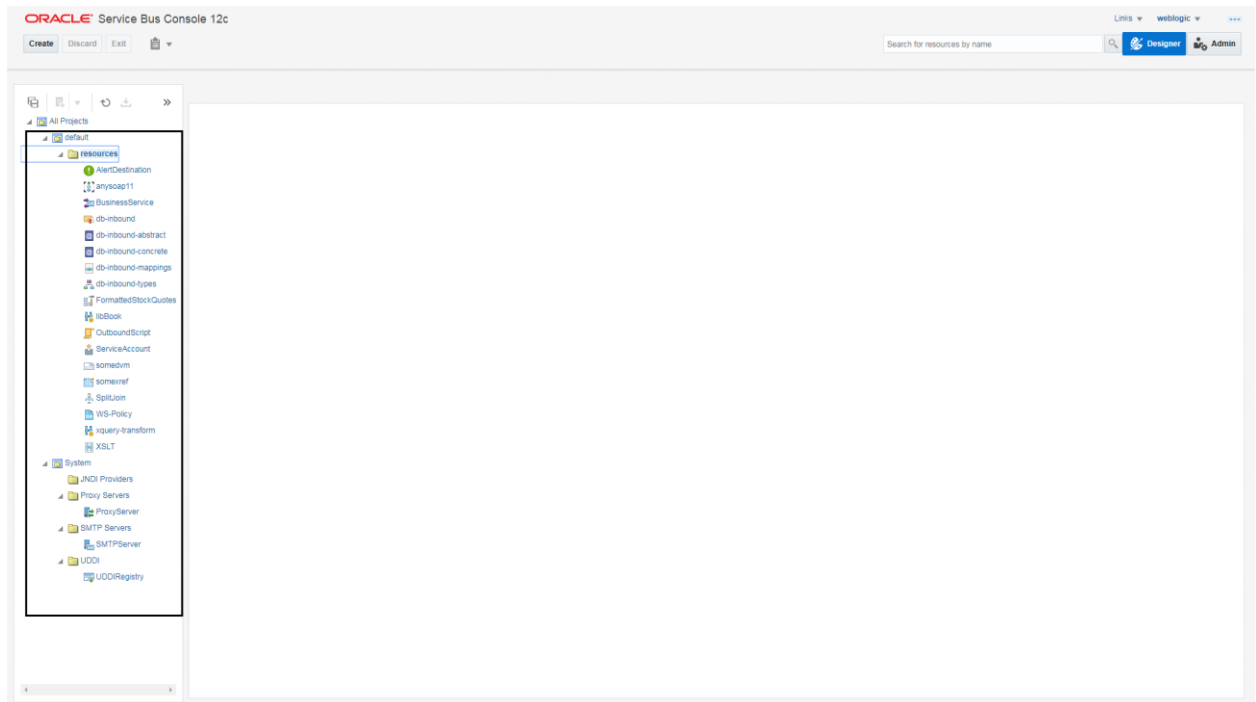


11. Click **Save** and **Build Now**.

After some time the job will complete with a status of either “Success” or “Failed”. The screenshots below show the status of build job number 73 and its console log.



12. The following screenshot of the Oracle Service Bus Console shows that projects have deployed on Oracle SOA Cloud Service from an Oracle Developer Cloud Service build job.



9. Issues With Certificates While Executing a Build Job

If the Oracle SOA Cloud Service server is running over HTTPS and requires a certificate to be installed on the client side, then it has to be added in a `cacerts` file that's located at `test-application/certificates/cacerts`.

The certificate from the Oracle SOA Cloud Service server has to be imported into the `cacerts` file and the file has to be pushed to Oracle Developer Cloud Service Git repository.

The absolute location of this file has to be provided as a parameter to the key `weblogic.security.CustomTrustKeyStoreFileName` in the **JVM Options** field described in the previous section.