

ONAP SO Debug Env Setup

Chuanyu Chen 2017.8.24

Seshu Kumar M 2017.9.24

1、 Get All Artifacts of SO

Build all projects of SO, we can get the following artifacts of SO from each target folder:

asdc-controller-1.1.0-SNAPSHOT.war

mso-api-handler-infra-1.1.0-SNAPSHOT.war

mso-catalog-db-adapter-1.1.0-SNAPSHOT.war

mso-network-adapter-1.1.0-SNAPSHOT.war

mso-requests-db-adapter-1.1.0-SNAPSHOT.war

mso-sdnc-adapter-1.1.0-SNAPSHOT.war

mso-tenant-adapter-1.1.0-SNAPSHOT.war

mso-vnf-adapter-1.1.0-SNAPSHOT.war

mso-workflow-message-adapter-1.1.0-SNAPSHOT.war

MSOCockpit-1.1.0-SNAPSHOT.war

MSOCommonBPMN-1.1.0-SNAPSHOT.war

MSOInfrastructureBPMN-1.1.0-SNAPSHOT.war

MSOMockServer-1.1.0-SNAPSHOT.war

2、 Prepare the docker build folder

2.1 Copy dockerFiles

Copy docker-files folder of the packages project as the working path.

共享 ▾ 新建文件夹			
名称	修改日期	类型	大小
chef-configs	2017/8/4 17:26	文件夹	
jboss-configs	2017/8/4 17:26	文件夹	
scripts	2017/8/4 17:26	文件夹	
Dockerfile.aria	2017/8/23 15:37	ARIA 文件	2 KB
Dockerfile.jacoco	2017/8/23 15:37	JACOCO 文件	1 KB
Dockerfile.mso-arquillian	2017/8/4 17:26	MSO-ARQUILLIA...	4 KB
Dockerfile.mso-chef-final	2017/8/4 17:26	MSO-CHEF-FINA...	4 KB
Dockerfile.ubuntu-16.04-update	2017/8/4 17:26	04-UPDATE 文件	1 KB
Dockerfile.wildfly-10	2017/8/4 17:26	WILDFLY-10 文件	2 KB
settings.xml	2017/8/4 17:26	XML 文档	2 KB

2.2 Copy Configurations to working path.

1. Copy docker volume configs to the working path.

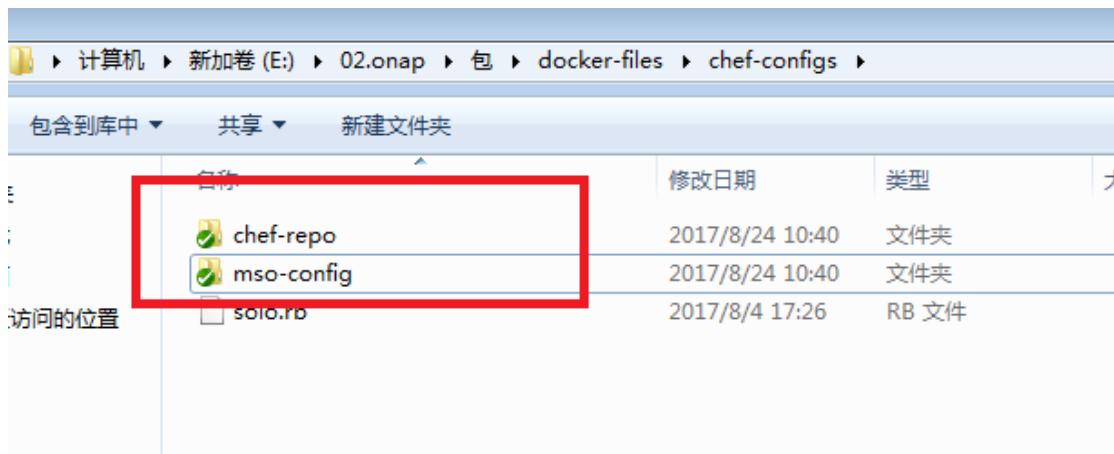
Source: docker-config\volume of docker-config project.

只 打开 新建文件夹				
名称	修改日期	类型	大小	
chef-configs	2017/8/24 10:14	文件夹		
jboss-configs	2017/8/24 10:14	文件夹		
mariadb	2017/8/24 10:27	文件夹		
mso	2017/8/24 10:27	文件夹		
scripts	2017/8/24 10:14	文件夹		
Dockerfile.aria	2017/8/23 15:37	ARIA 文件	2 KB	
Dockerfile.jacoco	2017/8/23 15:37	JACOCO 文件	1 KB	
Dockerfile.mso-arquillian	2017/8/4 17:26	MSO-ARQUILLIA...	4 KB	
Dockerfile.mso-chef-final	2017/8/4 17:26	MSO-CHEF-FINA...	4 KB	
Dockerfile.ubuntu-16.04-update	2017/8/4 17:26	04-UPDATE 文件	1 KB	
Dockerfile.wildfly-10	2017/8/4 17:26	WILDFLY-10 文件	2 KB	
settings.xml	2017/8/4 17:26	XML 文档	2 KB	

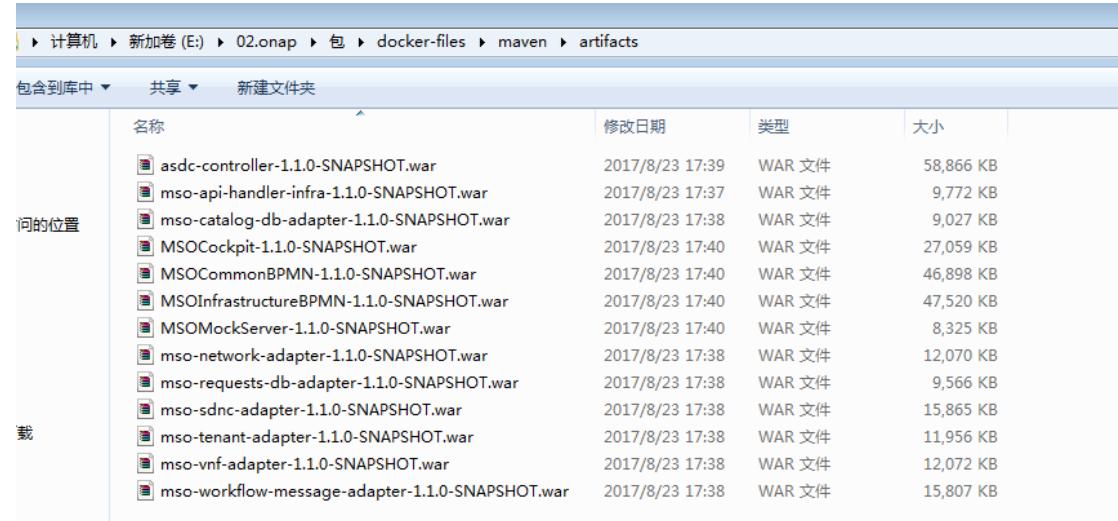
2. Copy the chef-configs files to working_path\chef-configs

Source: so-config project and chef-repo project.

Note: we need to rename so-config to mso-config. Because the DockerFile have not been changed, it use the name as mso-config, I think it is an issue that ATT change mso to so results.



3. Put the artifacts to maven/artifacts under the working path.



4. modify `\scripts\start-jboss-server.sh` for debug as below.

Add –debug for SO jboss start command.

```
$JBoss_HOME/bin/standalone.sh --debug -c standalone-full-ha-mso.xml &
```

```
#!/usr/bin/python

# This script is part of the chef-cookbook-tester project.
# It is used to validate that a .chef/roles directory contains valid
# chef cookbooks.

# Usage: ./check-roles.py <path>
# Example: ./check-roles.py /tmp/test-cookbooks

# This script is part of the chef-cookbook-tester project.
# It is used to validate that a .chef/roles directory contains valid
# chef cookbooks.

# Usage: ./check-roles.py <path>
# Example: ./check-roles.py /tmp/test-cookbooks
```

3、Build Docker Images

3.1 Copy the working path to linux.

```
root@SZX1000124253:~# cd /opt/mso/docker-files/
root@SZX1000124253:/opt/mso/docker-files# ls -l
total 60
drwxr-xr-x 4 root root 4096 Aug 24 10:58 chef-configs
drwxr-xr-x 4 root root 4096 Aug 24 10:58 chef-repo
-rw-r--r-- 1 root root 1140 Aug 24 10:58 Dockerfile.aria
-rw-r--r-- 1 root root 1011 Aug 24 10:58 Dockerfile.jacoco
-rw-r--r-- 1 root root 4020 Aug 24 10:58 Dockerfile.mso-arquillian
-rw-r--r-- 1 root root 3622 Aug 24 10:58 Dockerfile.mso-chef-final
-rw-r--r-- 1 root root 517 Aug 24 10:58 Dockerfile.ubuntu-16.04-update
-rw-r--r-- 1 root root 1042 Aug 24 10:58 Dockerfile.wildfly-10
drwxr-xr-x 4 root root 4096 Aug 24 10:58 jboss-configs
drwxr-xr-x 4 root root 4096 Aug 24 10:58 mariadb
drwxr-xr-x 3 root root 4096 Aug 24 10:58 maven
drwxr-xr-x 3 root root 4096 Aug 24 10:58 mso
drwxr-xr-x 2 root root 4096 Aug 24 10:58 scripts
-rw-r--r-- 1 root root 1675 Aug 24 10:58 settings.xml
drwxr-xr-x 7 root root 4096 Aug 24 10:58 so-config
root@SZX1000124253:/opt/mso/docker-files#
```

3.2 Mariadb

Because we do not need to debug mariadb, so we can use the image from nexus3.onap.org:10001/mariadb

```
root@SZX1000124253:~# docker pull nexus3.onap.org:10001/mariadb
Using default tag: latest
latest: Pulling from mariadb
a8e225cee373: Pull complete
0e1c704443c4: Pull complete
2ea2b04d5bba: Downloading [> 540 kB/74.97 MB
28735c5f20f6: Download complete
58f5b3b3c375: Download complete
2473877f1349: Download complete
26ac5660158b: Download complete
5c045c503830: Download complete
69e8d29cd0ae: Download complete
9a337be08e55: Download complete
]
root@SZX1000124253:~# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       VIRTUAL SIZE
<none>            <none>   9e1c704443c4    2 days ago    142.6 MB
onap/mso           latest   3f322a95c781    8 days ago    1.552 GB
onap/wildflynew    latest   37d39a45afe3    8 days ago    632.6 MB
<none>            <none>   d49a5e57ce88    9 days ago    633.6 MB
ubuntu              latest   5e4fe687078     13 days ago   120.1 MB
nexus3.onap.org:10001/mariadb  latest   4bd9aef9970f    3 weeks ago   397 MB
<none>            latest   3e30ca73a8d8    3 months ago  213.7 MB
```

3.3 Download ubuntu image

```
root@SZX1000124253:~# docker pull ubuntu
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
knopfe>	knopfe>	0e1e7044e3ct	2 days ago	142.6 MB
snappy/mcc	latest	3f322a95c781	8 days ago	1.552 GB
snappy/wildflynew	latest	37d34e45ca89	8 days ago	632.6 MB
knopfe>	knopfe>	d59af57ce08	9 days ago	633.8 MB
ubuntu	latest	58a4fe6a7878	13 days ago	120.1 MB
nexus3.onap.org:10001/mariadb	latest	4bd9deff9970f	3 weeks ago	397 MB
jstest2	latest	3e3dcda79a6d	3 months ago	213.7 MB
jstest	latest	b9f4c9007811	3 months ago	213.7 MB
ubuntu-with-vi-dockerfile	latest	b9f4c9007811	3 months ago	213.7 MB
jstest1	latest	b9f4c9007811	3 months ago	213.7 MB
ubuntu-with-vi	latest	0ce76a9a2ac5b	3 months ago	237 MB
onap/wildfly	latest	b6d705f3ad1c	3 months ago	639.8 MB
registry	latest	bb602e0824bd	3 months ago	33.2 MB
centos	latest	dt35079dc2ee	5 months ago	191.8 MB
jinxinjob1	latest	8e9761f70075	10 months ago	1.093 MB
job1	latest	7e8513992f72	10 months ago	1.093 MB
jinxin1983/busybox	v1	9967c5ab88de	10 months ago	1.093 MB
registry.example.net:5001/jinxin1983/busybox	v1	9967c5ab88de	10 months ago	1.093 MB
busybox	latest	9967c5ab88de	10 months ago	1.093 MB
programm/stress	latest	873c2e292d23	3 years ago	281.8 MB

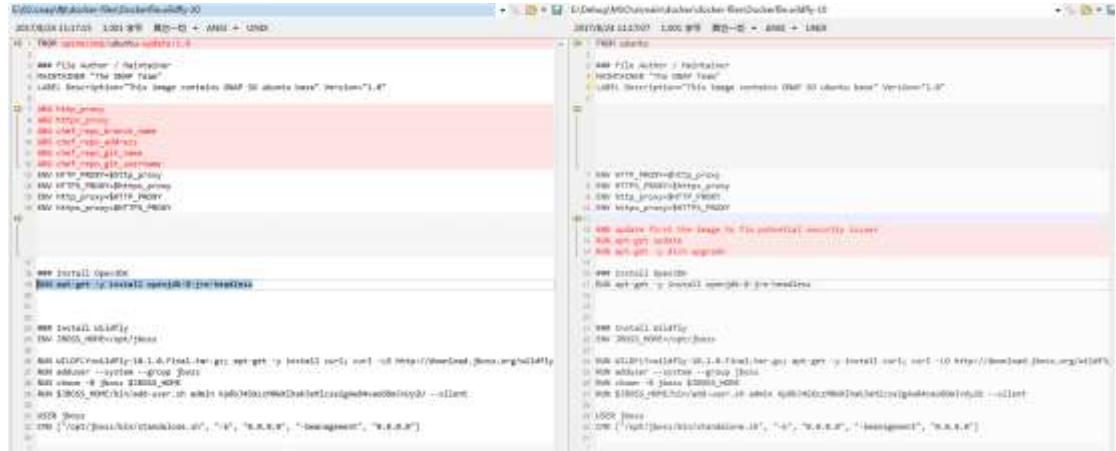
3.4 Build wildfly image

- A. First of all, we need to modify “Dockerfile.wildfly-10” as below.
 - a) From ubuntu, because we download ubuntu image, we need to make sure the name same as that we download
 - b) Remove ARG, this is because my docker version is old , ARG command is not supported and here these ARGs is not used in this docker file.
 - c) Add below lines for install jre, this is from MSO repo’s wildfly DockerFile, if there are no these lines, “RUN apt-get -y install openjdk-8-jre-headless” will be failed.

update first the image to fix potential security issues

RUN apt-get update

```
RUN apt-get -y dist-upgrade
```



- B. And then rename “Dockerfile.wildfly-10” as “Dockerfile”.

```

root@SZX1000124253:/opt/mso/docker-files# ls -l
total 60
drwxr-xr-x 4 root root 4096 Aug 24 10:58 chef-configs
drwxr-xr-x 4 root root 4096 Aug 24 10:58 chef-repo
-rw-r--r-- 1 root root 892 Aug 24 11:02 Dockerfile
-rw-r--r-- 1 root root 1140 Aug 24 10:58 Dockerfile.aria
-rw-r--r-- 1 root root 1011 Aug 24 10:58 Dockerfile.jacoco
-rw-r--r-- 1 root root 4020 Aug 24 10:58 Dockerfile.mso-arquillian
-rw-r--r-- 1 root root 3622 Aug 24 10:58 Dockerfile.mso-chef-final
-rw-r--r-- 1 root root 517 Aug 24 10:58 Dockerfile.ubuntu-16.04-update
drwxr-xr-x 4 root root 4096 Aug 24 10:58 jboss-configs
drwxr-xr-x 4 root root 4096 Aug 24 10:58 mariadb
drwxr-xr-x 3 root root 4096 Aug 24 10:58 maven
drwxr-xr-x 3 root root 4096 Aug 24 10:58 mso
drwxr-xr-x 2 root root 4096 Aug 24 10:58 scripts
-rw-r--r-- 1 root root 1675 Aug 24 10:58 settings.xml
drwxr-xr-x 7 root root 4096 Aug 24 10:58 so-config
root@SZX1000124253:/opt/mso/docker-files#

```

C. Build docker image. Here I name it as onap/wildfly1 , because onap/wildfly already exists.

```

root@SZX1000124253:/opt/mso/docker-files# docker build -t onap/wildfly1 .
Sending build context to Docker daemon 292.7 MB
Step 1 : FROM ubuntu
--> 56a4fe6a7878
Step 2 : MAINTAINER "The ONAP Team"
--> Running in 432482f8d509
--> d30c69371745
Removing intermediate container 432482f8d509
Step 3 : LABEL Description "This image contains ONAP SO ubuntu base" Version "1.0"
--> Running in 783fa38f7c40
--> 89f73cad4e17
Removing intermediate container 783fa38f7c40
Step 4 : ENV HTTP_PROXY $http_proxy
--> Running in 0e78030b32f9
--> 332ba5acf64e
Removing intermediate container 0e78030b32f9
Step 5 : ENV HTTPS_PROXY $https_proxy
--> Running in b94d8c2f7dac

```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
onap/wildfly1	latest	3d5412e06c65	21 seconds ago	632.6 MB
maxos3.onap.org:10001/mariadb	latest	3e337be06e55	2 days ago	397.2 MB
onap/mso	latest	3f322a95c781	5 days ago	1.552 GB
onap/wildflynew	latest	37d1fa55a8e3	8 days ago	632.6 MB
<none>	<none>	d49a5f57ce88	9 days ago	633.8 MB
ubuntu	latest	36a1fe6a7878	13 days ago	120.1 MB
<none>	<none>	4bd5def9970f	3 weeks ago	397 MB
jatext2	latest	3e3ddaa79a6d	3 months ago	213.7 MB
jatext	latest	39f4c9007811	3 months ago	213.7 MB
ubuntu-with-vi-dockerfile	latest	39f4c9007811	3 months ago	213.7 MB
jatext1	latest	39f4c9007811	3 months ago	213.7 MB
ubuntu-with-vi	latest	0cc76a9a265b	3 months ago	237 MB
onap/wildfly	latest	b8d705f3ad1c	3 months ago	633.8 MB
registry	latest	b9612e0824bd	3 months ago	93.2 MB
centos	latest	d4550798c2ee	8 months ago	191.8 MB
jinxin/job1	latest	de976f7d0755	10 months ago	1.093 MB
job1	latest	7e8113892c72	10 months ago	1.093 MB
busybox	latest	996705ad80de	10 months ago	1.093 MB
jinxin1983/busybox	v1	996705ad80de	10 months ago	1.093 MB
registry.example.net:3001/jinxin1983/busybox	v1	996705ad80de	10 months ago	1.093 MB
programm/stress	latest	879c26292d23	3 years ago	281.6 MB

3.5 Build SO image

1. Modify “Dockerfile.mso-chef-final” as below .

Change from image to onap/wildfly1 which we have build.

Remove ARG block because our docker version does not support ARG command and it is not used.

Add module.xml at line 70. I think it is a issue.



```
diff --git a/Dockerfile.boss-chef-final b/Dockerfile.m2-chef-final
index 3e0a2d0..f3a2a2c 100755
--- a/Dockerfile.boss-chef-final
+++ b/Dockerfile.m2-chef-final
@@ -1,10 +1,10 @@
 FROM alpine:latest
-MAINTAINER "The ONSP Team"
+MAINTAINER "The ONSP Team"
-LABEL Description="This image contains the ONSP SO" Version="1.0"
+LABEL Description="This image contains the ONSP SO" Version="1.0"

-#Hadoop proxy
-#HDFS proxy
-#HTTP proxy
-#HTTPS proxy
-#S3 proxy
-#Kafka proxy
-#MQTT proxy
-#Redis proxy
-#MongoDB proxy

 ENV CHEF_REPO_NAME="chef-repo"
 ENV CHEF_CONFIG_NAME="esb-config"

@@ -14,7 +14,7 @@ Downloading dependencies
 HADOOP none
```

```
diff --git a/Dockerfile.boss-m2-chef-final b/Dockerfile.m2-chef-final
index 3e0a2d0..f3a2a2c 100755
--- a/Dockerfile.boss-m2-chef-final
+++ b/Dockerfile.m2-chef-final
@@ -1,10 +1,10 @@
 FROM alpine:latest
-MAINTAINER "The ONSP Team"
+MAINTAINER "The ONSP Team"
-LABEL Description="This image contains the ONSP SO" Version="1.0"
+LABEL Description="This image contains the ONSP SO" Version="1.0"

-#Hadoop proxy
-#HDFS proxy
-#HTTP proxy
-#HTTPS proxy
-#S3 proxy
-#Kafka proxy
-#MQTT proxy
-#Redis proxy
-#MongoDB proxy

 ENV CHEF_REPO_NAME="chef-repo"
 ENV CHEF_CONFIG_NAME="esb-config"

@@ -14,7 +14,7 @@ Downloading dependencies
 HADOOP none
```

2. Rename Dockerfile.mso-chef-final to Dockerfile (you can rename the existing DockerFile back to Dockerfile.wildfly-10 first).

- ### 3. Build SO docker image

4 Run Docker Containers

4.1 Run MariaDB

The Command to run mariadb is:

```
docker run -d --name mariadb -h db.mso.testlab.openecomp.org -e  
MYSQL_ROOT_PASSWORD=password -p 3306:3306 -v  
/opt/mso/docker-files/mariadb/docker-entrypoint-initdb.d:/docker-entrypoint-initdb.d -v  
/opt/mso/docker-files/mariadb/conf.d:/etc/mysql/conf.d nexus3.onap.org:10001/mariadb
```

the mariadb's username and password is :root password

```
root@S2X1000124253:~# docker exec -it mariadb /bin/bash  
root@db:/# mysql -uroot -ppassword  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 14538  
Server version: 10.2.7-MariaDB-10.2.7+maria~jessie-log mariadb.org binary distribution  
  
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database      |  
+-----+  
| camundabpmn   |  
| information_schema |  
| mso_catalog    |  
| mso_requests   |  
| mysql          |  
| performance_schema |  
+-----+  
6 rows in set (0.00 sec)  
  
MariaDB [(none)]>
```

4.2 Run SO

The Command to run SO is:

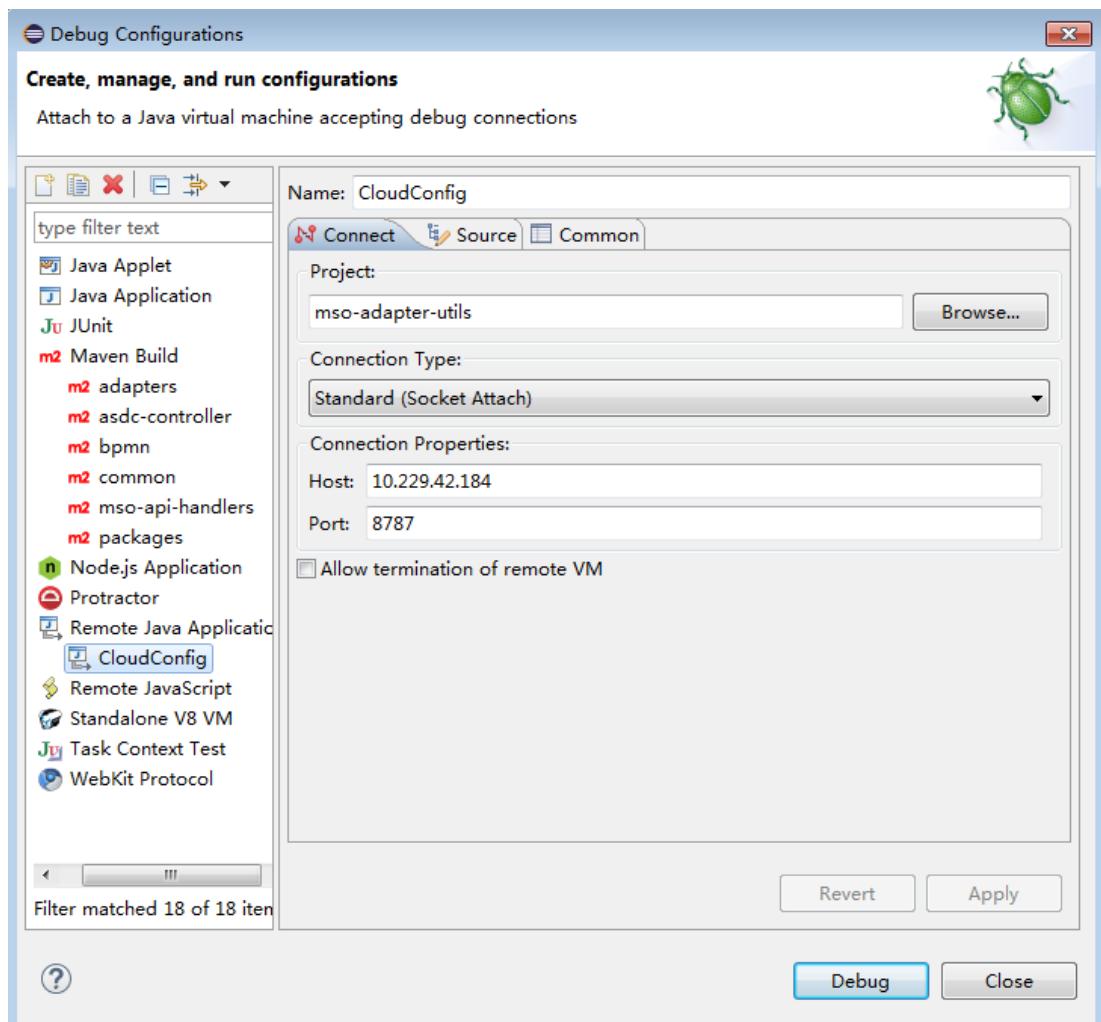
```
docker run -d --name mso -h mso.mso.testlab.openecomp.org -e  
MYSQL_ROOT_PASSWORD=password --link=mariadb:db.mso.testlab.openecomp.org -p  
3904:3904 -p 8787:8787 -p 3905:3905 -p 8080:8080 -p 9990:9990 -v  
/opt/mso/docker-files/mso/chef-config:/shared onap/mso
```

Note:-p 8787 if for debug;

--link=mariadb:db.mso.testlab.openecomp.org it's for link to mariadb instance.

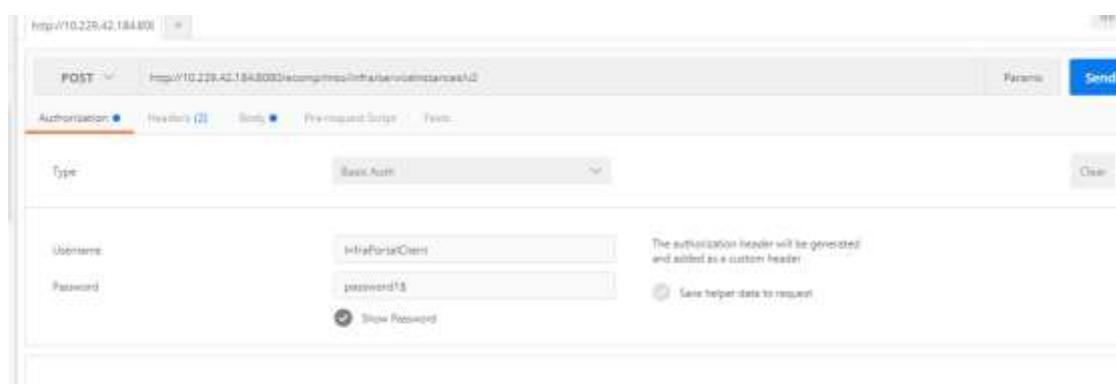
5. Debug Using Eclipse & Postman

1.Add new Remote java application debug configuration as below. The port is 8787.



2. Post message using postman

Note: Basic Auth is needed , username:InfraPortalClient ,password: password1\$



We will get the breakpoint at the eclipse.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Left Side:** Package Explorer view showing various project packages like adapters, arquillian-unit-tests, etc.
- Center:** Editor view displaying Java code for a class named `ServiceInstances`. The code handles requests, uses ObjectMapper for JSON conversion, and logs errors if the request object fails validation.
- Bottom:** Problems view showing a stack trace for a failed request. The stack trace includes frames from `ServiceInstances`, `ServiceInstances$Proxy`, and `NativeMethodAccessorImpl`.

```

private Response serviceInstances(String requestid, Action action, HashMap<String, String> instancevalues)
{
    String instanceId = BUDCheckers.generateInstanceId();
    long starttime = System.currentTimeMillis();
    logger.debug("requestid is " + requestid);
    ServiceInstancesRequest sir = null;
    try{
        ObjectMapper mapper = new ObjectMapper();
        sir = mapper.readValue(requestid, ServiceInstancesRequest.class);
    } catch(Exception e){
        logger.debug("Mapping of request to JSON object failed : " + e);
        Response response = msoRequest.buildServiceErrorResponse(HttpStatus.SC_BAD_REQUEST, MsoException.
                MAPPING_OF_REQUEST_TO_JSON_OBJECT_FAILED, e.getMessage(),
                ErrorNumbers.SVC_BAD_PARAMETER, null);
        if (msoRequest.getRequestId() != null) {
            logger.debug("Mapping of request to JSON object failed");
            msoRequest.createRequestRecord(Status.FAILURE, action);
        }
        logger.error("Message from API REQUEST_VALIDATION_ERROR, MSO_FRESH_APACHE_IN_WA, " + " " + instanceId);
        recordAuditEvent(starttime, logger.StatusCode.FAILED, logger.ResponseCode.SERVICE_UNAVAILABLE);
        logger.debug("End of the transaction, the final response is: " + (String) response.getResponseBody());
        return response;
    }
}

```

Note:

Find additional information on the wiki

<https://wiki.onap.org/display/DW/Development+Environment>