ONAP SO Debug Env Setup

Chuanyu Chen 2017.8.24

# 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

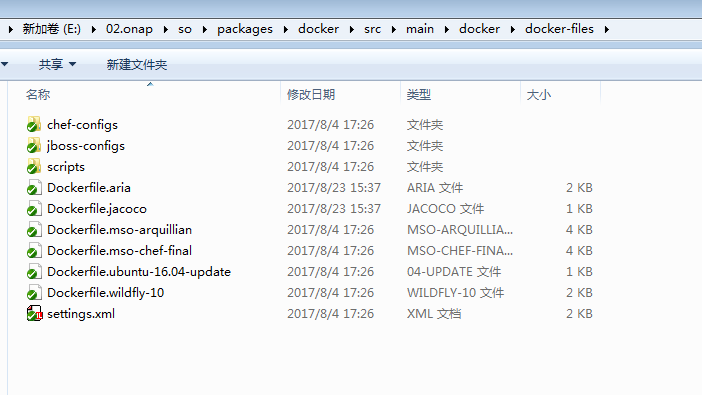
MSOInfrastructureBPMN-1.1.0-SNAPSHOT.war

MSOMockServer-1.1.0-SNAPSHOT.war

# Prepare the docker build folder

## 2.1 Copy dockerFiles

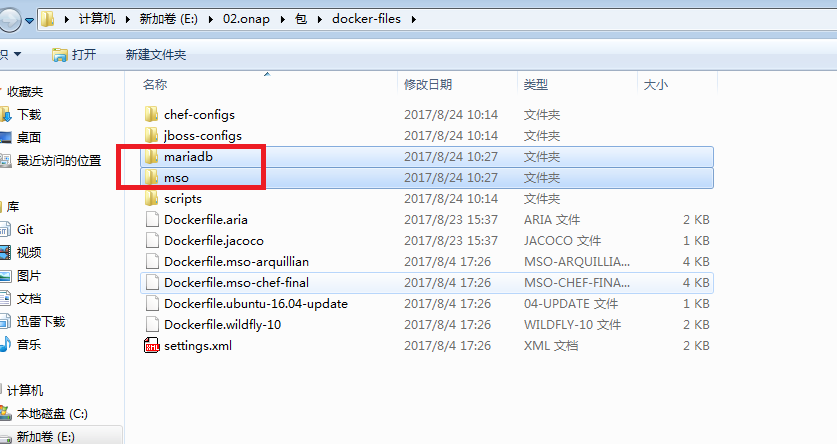
Copy docker-files folder of the packages project as the working path.



## 2.2 Copy Configurations to working path.

1. Copy docker volume configs to the working path.

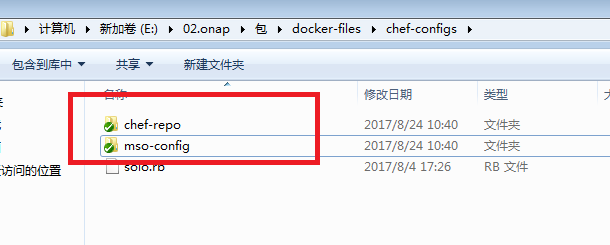
Source: docker-config\volumes of docker-config project.



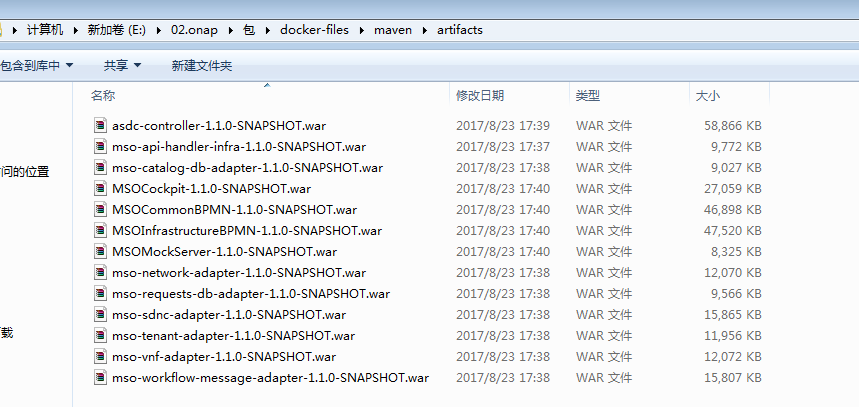
1. Copy the chef-configs files to working\_path\chef-configs

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 resultes.



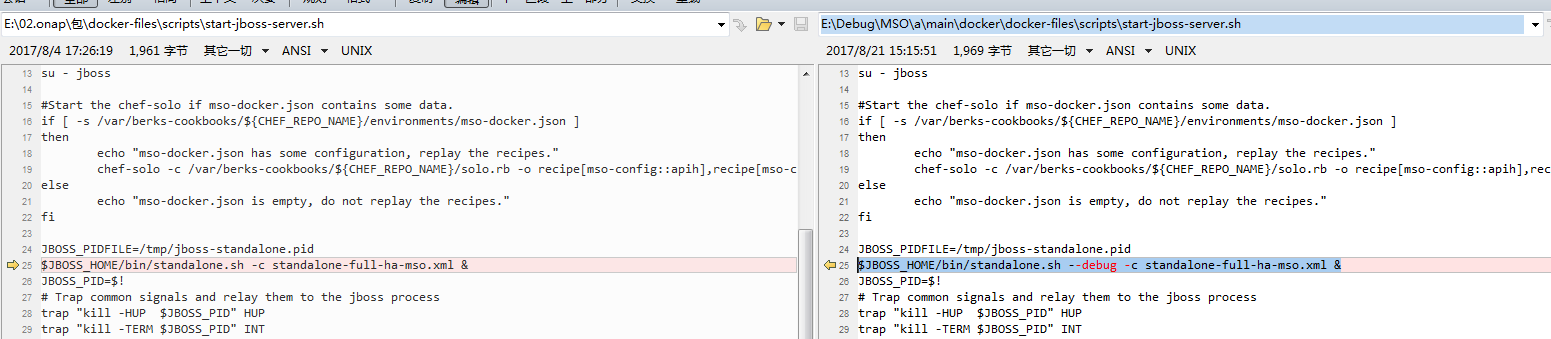
1. Put the artifacts to maven/artifacts under the working path



1. mofidy \scripts\start-jboss-server.sh for debug as below.

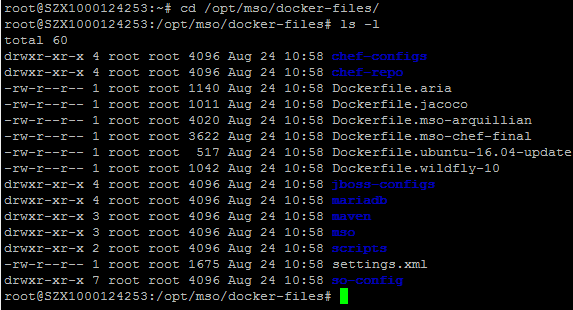
Add –debug for SO jboss start command.

$JBOSS\_HOME/bin/standalone.sh --debug -c standalone-full-ha-mso.xml &



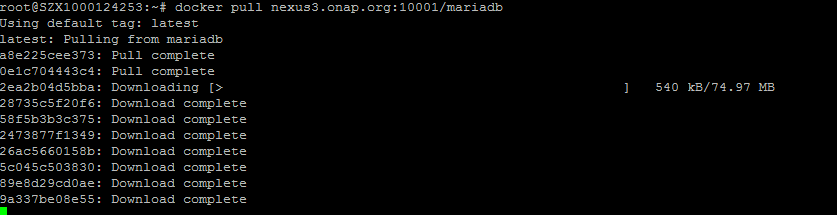
# Build Docker Images

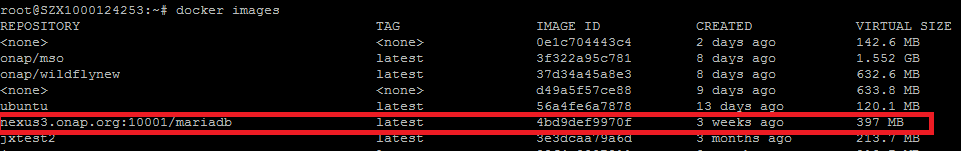
## 3.1 Copy the working path to linux.



## 3.2 Mariadb

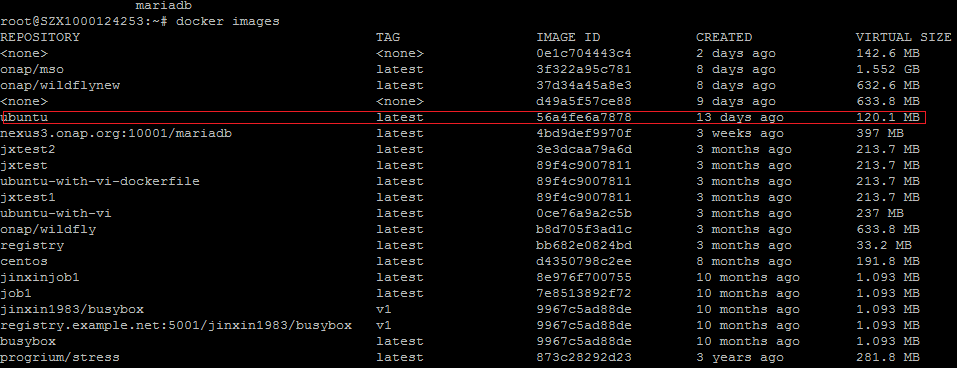
Because we do not need to debug mariadb, so we can use the image from nexus3.onap.org:10001/mariadb





## 3.3 Download ubuntu image





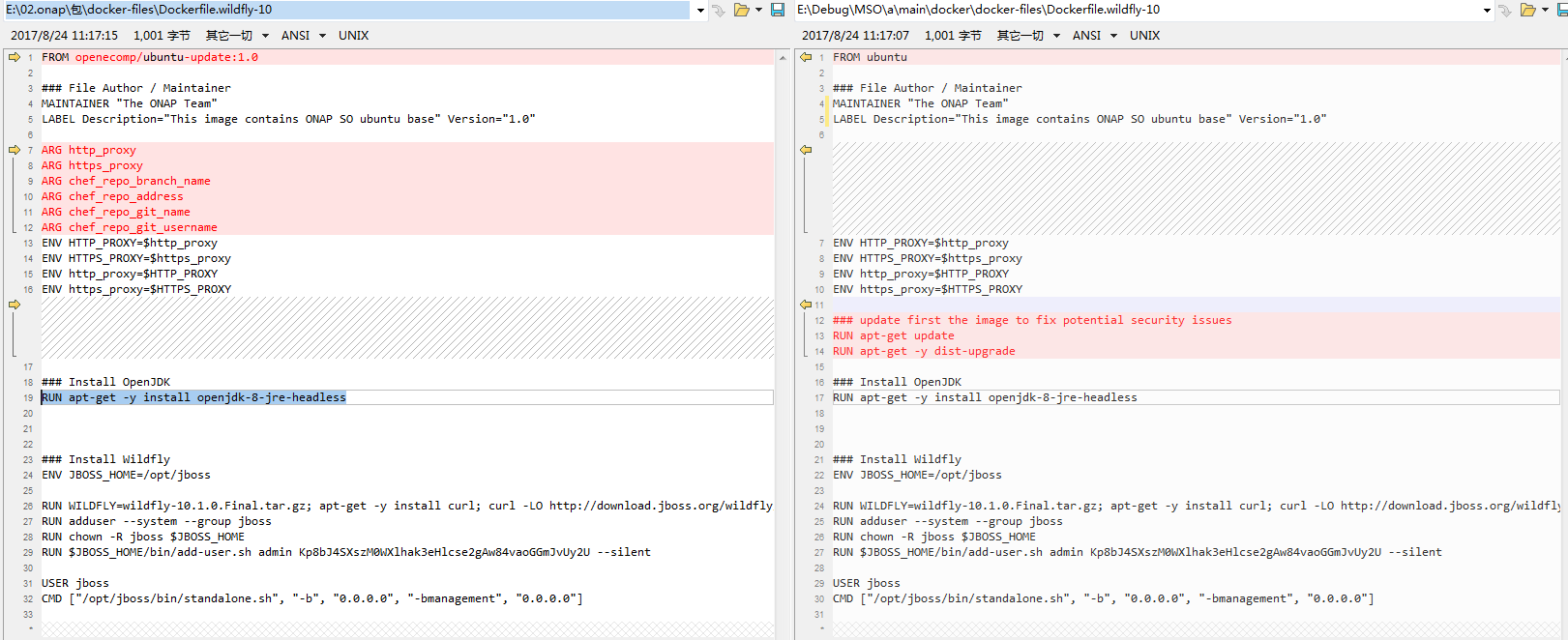
## 3.4 Build wildfly image

1. First of all, we need to modify “Dockerfile.wildfly-10” as below.
2. From ubuntu, because we download ubuntu image, we need to make sure the name same as that we download
3. 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.
4. 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.

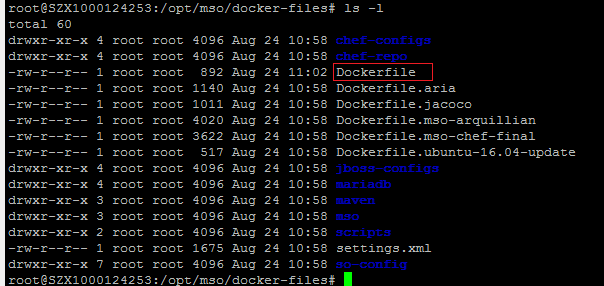
### update first the image to fix potential security issues

RUN apt-get update

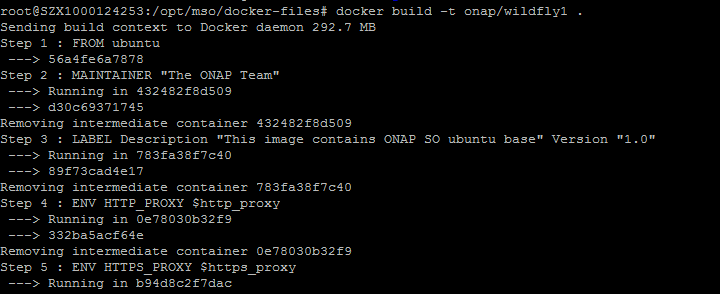
RUN apt-get -y dist-upgrade

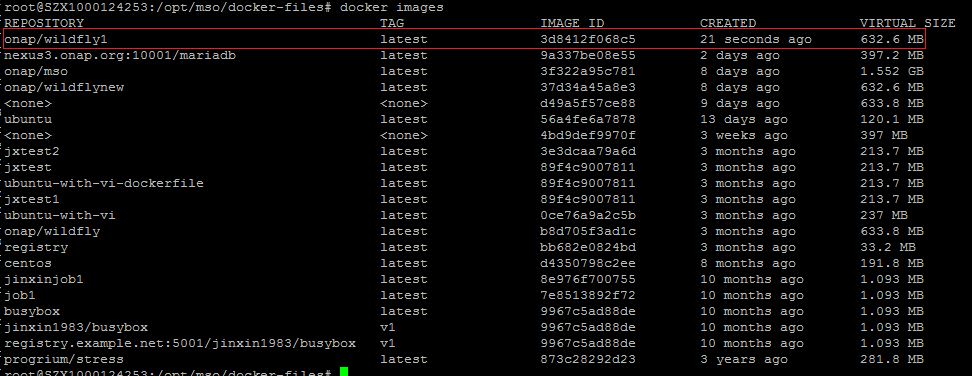


1. And then rename “Dockerfile.wildfly-10” as “Dockerfile”.



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





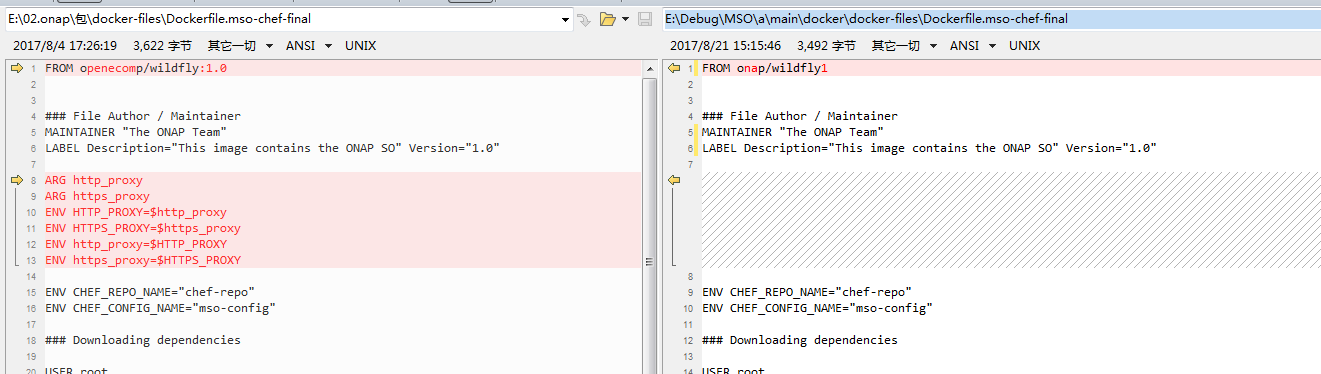
## 3.5 Build SO image

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

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

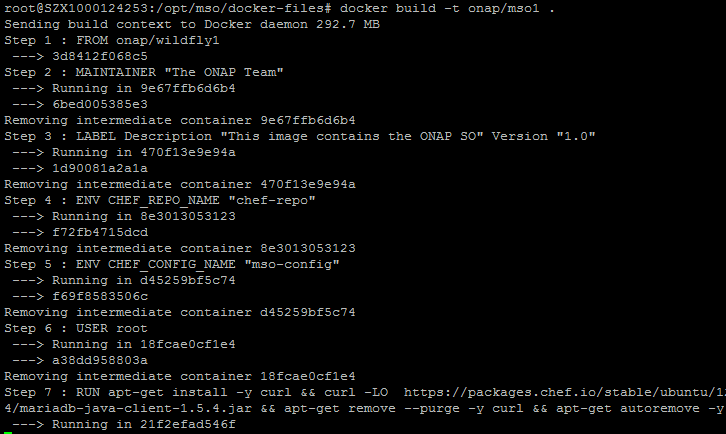
Remove ARG block because our docker version does not support ARG command and it is not used.

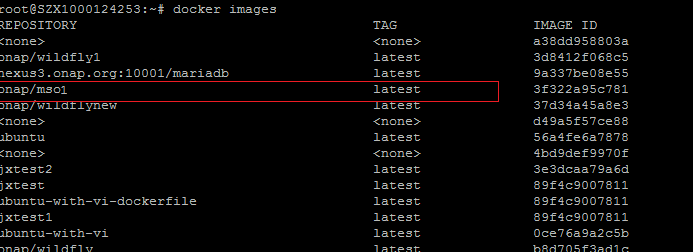
Add module.xml at line 70. I think it is a issue.





1. Rename Dockerfile.mso-chef-final to Dockerfile (you can rename the exsiting DockerFile back to Dockerfile.wildfly-10 first).
2. Build SO docker image





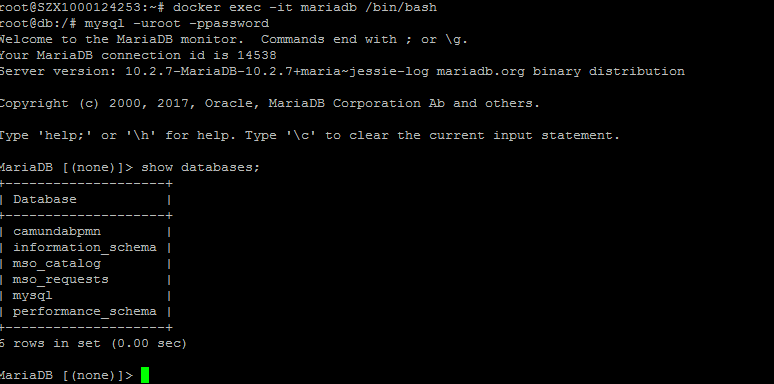
# 4 Run Docker Containers

## 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



## Run SO

The Command to run SO is:

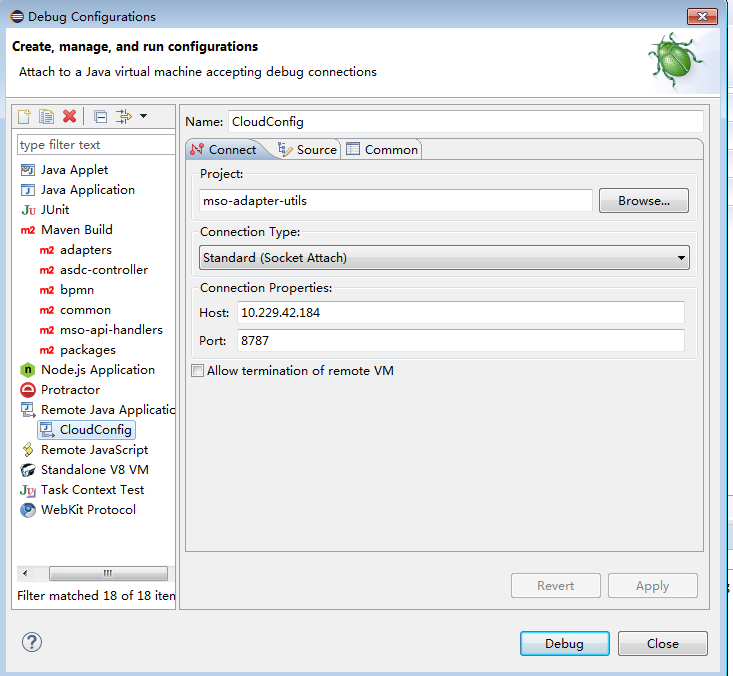
docker run -d --name mso -h mso.mso.testlab.openecomp.org -e MYSQL\_ROOT\_PASSWORD=password --link=mariadb:db.mso.testlab.openecomp.org -p 3904:3904 -p 8787:8787 -p 3905:3905 -p 8080:8080 -p 9990:9990 -v /opt/mso/docker-files/mso/chef-config:/shared onap/mso

Note:-p 8787 if for debug;

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

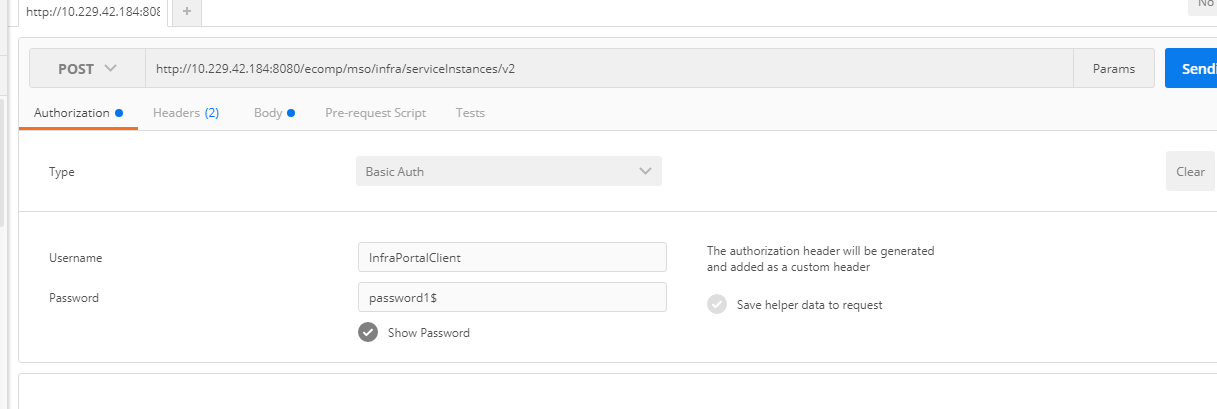
# 5. Debug Using Eclipse & Postman

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.

