**Gather maven/mule commands**

* Deploy Application on Exchange

mvn clean deploy -DclientID=23c7e68d140a4a9588511e7c3bc416bf -DclientSecret=26EAFe4D5D764c9C96941cfb58e1255a -s settings.xml

* Deploy Application on Cloudhub 1.0 using mule maven plugin

mvn clean deploy -DmuleDeploy -Dregion=us-east-2 -Dworkers=1 -DworkerType=MICRO -Denvironment=Sandbox -DappName=ecom-order-ext-eapi -DclientID=23c7e68d140a4a9588511e7c3bc416bf -DclientSecret=26EAFe4D5D764c9C96941cfb58e1255a -Denv=dev -Dkey=1234567812345678 -Danypoint.clientId=f25d3ad0983445a699ceef9b9ee26cfd -Danypoint.clientSecret=EF7cA31B14354cDea42A7C4d30833419 -s settings.xml

* Deploy Application on Cloudhub 2.0 using mule maven plugin

mvn clean deploy -DmuleDeploy -DtargetName=Cloudhub-US-East-2 -Denvironment=Sandbox -Dreplicas=1 -DvCores=0.1 -DappName=ecom-order-ext-eapi -DclientID=23c7e68d140a4a9588511e7c3bc416bf -DclientSecret=26EAFe4D5D764c9C96941cfb58e1255a -Denv=dev -Dkey=1234567812345678 -Danypoint.clientId=f25d3ad0983445a699ceef9b9ee26cfd -Danypoint.clientSecret=EF7cA31B14354cDea42A7C4d30833419 -s settings.xml

* Encrypt yaml files

java -cp secure-properties-tool.jar com.mulesoft.tools.SecurePropertiesTool file encrypt AES CBC 1234567812345678 config-secure-dev.yaml dev-out.yaml

**Gather putty commands**

* Upload jar file to ec2 instance

pscp -scp -i ec2-key.ppk oms-backend-0.0.1-SNAPSHOT.jar [ec2-user@ec2-13-211-167-26.ap-southeast-2.compute.amazonaws.com:~](mailto:ec2-user@ec2-13-211-167-26.ap-southeast-2.compute.amazonaws.com:~)

* Perform write permission on pem key

chmod 400 OmsPemKey.pem

* Connect to ec2 instance

ssh -i OmsPemKey.pem [ec2-user@ec2-13-211-167-26.ap-southeast-2.compute.amazonaws.com](mailto:ec2-user@ec2-13-211-167-26.ap-southeast-2.compute.amazonaws.com)

* Install JDK 17 if instance created for the first time

sudo yum install java-17-amazon-corretto

* Deploy the jar file with output logs enabled

java -jar oms-backend-0.0.1-SNAPSHOT.jar > output.log &

**Gather git commnds**

* Push Existing local repo to remote repo

git remote add origin ""

git branch -M main

git add .

git commit -m "message"

git push --set-upstream origin main

* Change remote repo for existing local repo

git remote set-url origin <https://github.com/OWNER/REPOSITORY.git>

* Create a new repository on the command line

echo "# oms-backend" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/kedarawasthi/oms-backend.git

git push -u origin main

**Gather AWS credentials**

**Iam account ID:** 891376909160

**Root user**

Username: awasthi.kedareshwar@gmail.com

**Iam user**

Console sign-in URL

<https://891376909160.signin.aws.amazon.com/console>

AWS Policy generator: <https://awspolicygen.s3.amazonaws.com/policygen.html>

**Gather SNOW crdentials**

Instance name: **dev191862**

Instance URL: [**https://dev191862.service-now.com/**](https://dev191862.service-now.com/)

Username: **admin**

**Gather Postman collections**

**Mule: ecom-order-ext-eapi (Local,cloudhube) collection url:**

**Others: oms-backend (lambda,ec2,local) collection url:**

**Gather API spec:**

**Raml url:**

**Exchange url:** https://anypoint.mulesoft.com/exchange/portals/deloitte-india-llp/a41e7b82-f610-496e-8898-19374947af3a/ecom-order-ext-eapi/

**Gather Documentation:**

**Document Url:**

**Code URLs**

**Mule: ecom-order-ext-eapi url:**

* https://github.com/kedarawasthi/ecom-order-ext-eapi

**Mule-Aws Lambda DLQ service:**

https://github.com/kedarawasthi/aws-stub-service

**Srpingboot oms-backend url:**

* https://github.com/kedarawasthi/oms-backend

**Java lambda-oms-backend url:**

* <https://github.com/kedarawasthi/lambda-oms-backend>

**Troubleshooting tips:**

* Bug in Runtime version 4.6- use 4.6-java8 or 4.6-java17 as workaround instead of 4.6.2 in app.runtime
* If using mule maven plugin, first deploy mule application to exchange and then deploy to cloud hub. Skipping former will result into deployment failure.