**MDB aPaaS documentation:**

**APAAS V3 DEPLOYMENT FOR MDB APPLICATION:**

**Flow**: Build from Bit Bucket Repo-> Push to Internal Nexus artifact-> aPaaS V3 image build-> Deploy to aPaaS V3-> promotion to external nexus to pull and run image -> deployment via promotion job for https URL projects

After provisioning of projects on aPaaS V3 platform like 9059, 9065. etc., flow can be summarized as:

* Use of Jenkins Job to build MDB code and push artefact to internal Nexus repo.
* Configuring Jenkins build for image building and deploying the application with groovy script.
* MA properties should be properly set for relevant environments via property utils tool.
* Vault secrets have been properly defined for correct deployment.
* Building an image of the nexus artefact on top of JBoss image.
* Performing deployment on aPaaS with the same image.
* After deployment, a route has been created to test application via message push.
* Usage of promotion job for further https URLs project deployment.

**Following are the step by step details for the deployment:**

**Step-1 Pre-requisites: defining proper MQ connections in property utils.**

We need to place certs for every environment for MQ in vault manager for having deployment in CIT-C. CIT-D and so on. and same in case of passwords for DS usernames also.

* Added cert to ma-truststore

e.g, [https://cftjira.barcapint.com/browse/BARCDEVOPS-111649](https://cftjira.barcapint.com/browse/BARCDEVOPS-111649#_blank)

* Added jks file

e.g, [https://cftjira.barcapint.com/browse/BARCDEVOPS-111641](https://cftjira.barcapint.com/browse/BARCDEVOPS-111641#_blank)

* Added jks password

e.g, [https://cftjira.barcapint.com/browse/BARCDEVOPS-111598](https://cftjira.barcapint.com/browse/BARCDEVOPS-111598#_blank)

**Step-2: Image building via Jenkins:**

* We need to build the mdb code and push artefact to Nexus via 9059 Jenkins Job.
* Above job has 4 stages which include pre-flight, checkout, build and push to nexus as artefact.
* Pre-flight is checking and validating all the necessary credentials.
* Checkout stage is used to get source code from Bit Bucket repo.
* Build stage is used to create a build from source code and prepare ear file out of it.
* Push to Nexus is actually pushing ear artefact to internal nexus repo.

**Below job then is used for the deployment to aPaaS:**

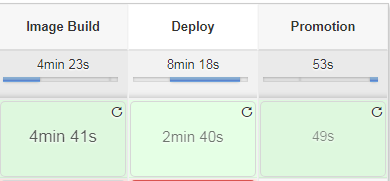
<https://jenkins1a.intranet.barcapint.com:8444/jenkins/job/BUK/job/BPMMDB/job/Imagebuilddeploy9059/>

* This is using MCAs gear configuration files (<https://stash.barcapint.com:8443/projects/GRP_BPM/repos/new_gear_config_files>) which we’ve forked to GRP\_BPM repo to have mqclient.jar in deployments output.
* We are getting uploaded version of ear on internal Nexus (<https://barcorpnexus.barcapint.com:8443/nexus> ) under path [/com/barclays/bpm/mdbpipeline/](https://barcorpnexus.barcapint.com:8443/nexus/service/local/repositories/MIS_Release/content/com/barclays/mis/apps/platforms/businessbanking/04.00.251/businessbanking-04.00.251-wsdls.zip)<version>

e.g.,

<https://barcorpnexus.barcapint.com:8443/nexus/content/repositories/GRP_BPM/com/barclays/bpm/>mdbpipeline

* It has three stages shown as below:



Above job is used to build image from nexus artifact and deploy to 9059 aPaaS.

* After image building, the image used to store in internal aPaaS V3 docker registry as a part of image stream which can be checked through aPaaS V3 platform: Details for docker registry is likewise depends upon the namespace:

e.g., Host-172.17.150.40:5000 (inaccessible through URL).

* After then the same image will be deployed on already existed image catalog (currently 9059) for aPaaS V3 platform (gl and sl) under our project (like 9065, 9059.) as a pod which run container for our application and create a route to access it.
* Then it to push the image to external nexus repo (<https://barclays-registry.barcapint.com:18443>) to make it available for pulling the same configuration for next or other environment deployments
* Currently we’re using 9059 project repo for promoting mdb application to external nexus repo.

* Once it will be deployed to 9059 it will promote the same image to external nexus repo under 9059:

URL for external Nexus;

<https://barclays-registry.barcapint.com:18443/>

**Step 3- Deployments to other environments projects:**

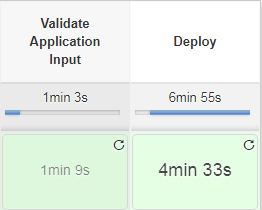
It will then use to deploy the tested image to promote on other environments or to another aPaaS environment.

**For Promotion Job:**

* We are using release/api branch of latest pipeline of MCA tools for this job.
* We have defined separate promotions Jenkins job defined for that inside promotion folder:

<https://jenkins1a.intranet.barcapint.com:8444/jenkins/job/BUK/job/BPMMDB/job/Promotion/job/mdbpipeline/>

* It has following stages:



And need to provide parameters accordingly to build it:

**Step 4- For other projects deployments:**

We have below different projects already setup for deployment of application in GL NP. We need the similar setup for PROD/others as well.

URL for accessing projects:

<https://console-gl.appcloud-np.barcapint.com>

**Important contacts:**

**bchat/DLs**: IT\_aPaaSV3, IT\_Bitbucket, IT\_Jenkins, IT\_JIRA, IT\_DEVTOOLS

**Issues and resolutions:**

1. **There was a vault resolve expression error per below which was resolved by setting up proper passwords for usernames in vault manager:**

08:06:31,787 ERROR [org.jboss.as.controller.management-operation] (ServerService Thread Pool -- 27) JBAS014612: Operation ("enable") failed - address: ([

    ("subsystem" => "datasources"),

    ("data-source" => "DB2\_DS1")

]) - failure description: "JBAS014802: Cannot resolve expression '${VAULT::Db2Ds::UserName::1}'"

1. **If Jenkins throws error like below: Need to retrigger again:**

Pulling image barclays-registry.barcapint.com:5000/bukeng/com.barclays.buk.jboss/jboss-sti-builder:0.1.206 ...

error: build error: failed to pull image: unauthorized: authentication required

error: The build 9059/mdb -8 status is "Failed"

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // ws

[Pipeline] }

[Pipeline] // node

[Pipeline] }

[Pipeline] // stage

[Pipeline] End of Pipeline

ERROR: script returned exit code 1

Finished: FAILURE

Using openshift URL CIPE:  [https://console-gl.appcloud-np.barcapint.com:443](https://console-gl.appcloud-np.barcapint.com/)

:createDeploymentConfig FAILED

FAILURE: Build failed with an exception.

\* Where:

Build file '/home/syspcbjenkins/store/workspace/BUK/BPMMDB/8924-RETAIL-PROCESSING-BPM-BBP-SIT1@2/v3.gradle' line: 531

\* What went wrong:

Execution failed for task ':createDeploymentConfig'.

> Could not get unknown property 'imageGroup' for task ':createDeploymentConfig' of type org.gradle.api.DefaultTask.

\* Try:

Run with --info or --debug option to get more log output.

\* Exception is:

org.gradle.api.tasks.TaskExecutionException: Execution failed for task ':createDeploymentConfig'.

        at org.gradle.api.internal.tasks.execution.ExecuteActionsTaskExecuter.executeActions(ExecuteActionsTaskExecuter.java:84)

        at org.gradle.api.internal.tasks.execution.ExecuteActionsTaskExecuter.execute(ExecuteActionsTaskExecuter.java:55)

        at org.gradle.api.internal.tasks.execution.SkipUpToDateTaskExecuter.execute(SkipUpToDateTaskExecuter.java:62)

        at org.gradle.api.internal.tasks.execution.ValidatingTaskExecuter.execute(ValidatingTaskExecuter.java:58)

        at org.gradle.api.internal.tasks.execution.SkipEmptySourceFilesTaskExecuter.execute(SkipEmptySourceFilesTaskExecuter.java:88)

        at org.gradle.api.internal.tasks.execution.ResolveTaskArtifactStateTaskExecuter.execute(ResolveTaskArtifactStateTaskExecuter.java:46)

**For Below also:**

error:oc tag error

1. **To extracts logs from pod side;**
2. **Go to under terminal tab for apaas environment;**

**Steps**: On aPaaS URL and under 9059/XXXX project

1. Go to business banking application.
2. Go to application ->Pods
3. Click on  terminal tab

1. **For co-event logs**

Type cd /home/jboss

Type cat co-event.log

1. **To check Version:**

sh-4.2$ ls

6.4.19.0

sh-4.2$ pwd

/apps/jboss/jboss-as

1. **To check server.log**

Go to /opt/eap/standalone/logs

1. **To check deployed ear**

Go to /opt/eap/standalone/deployments

1. **To check services.xml**

Go to /config/