**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

CI/CD Walkthrough For DTAC

Author: Gulam Moideen.

Creation Date: Feb 22, 2023

Last Updated: Feb 22, 2023

Version: Initial Document.

# CI

## Prerequisites: -

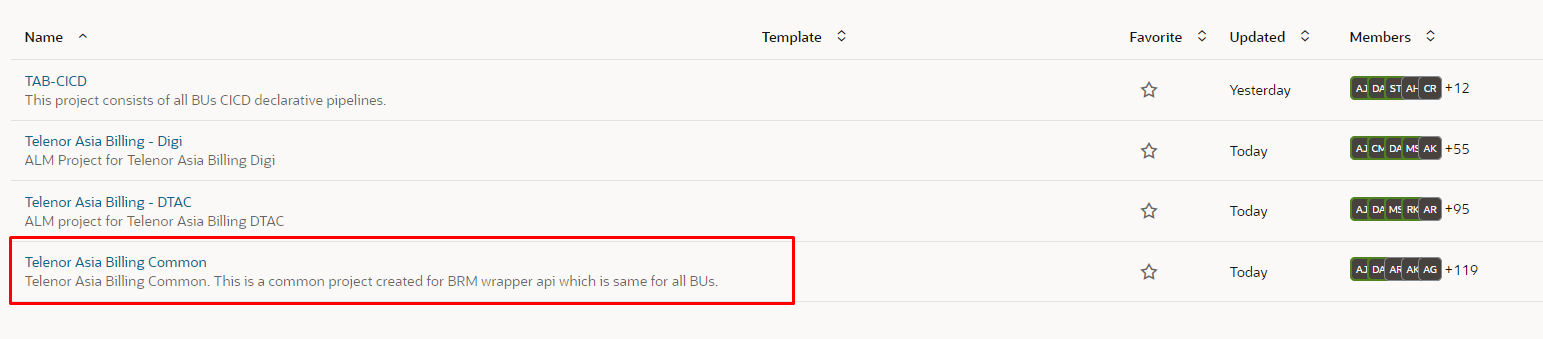
* Allthe latest changes should be commited on ALM by Dev Team
* Makefile and Dockerfile should be updated with latest image tag name by TA Team for **Telenor Asia Billing Common** project in ALM :
  + tab-common-brm
  + tab-common-rest

# LOGIN Credential

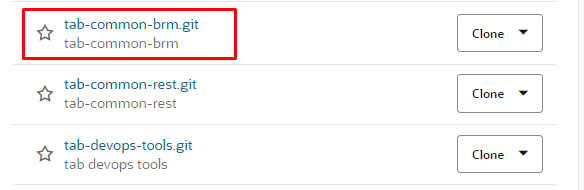
|  |  |  |  |
| --- | --- | --- | --- |
| **Application** | **URL** | **Username** | **Password** |
| Jenkins | http://100.76.194.253:8080/ | **devops** |  |
| ALM | https://alm.oraclecorp.com/cgbu | **Oracle e-mail** |  |

Steps for CI:

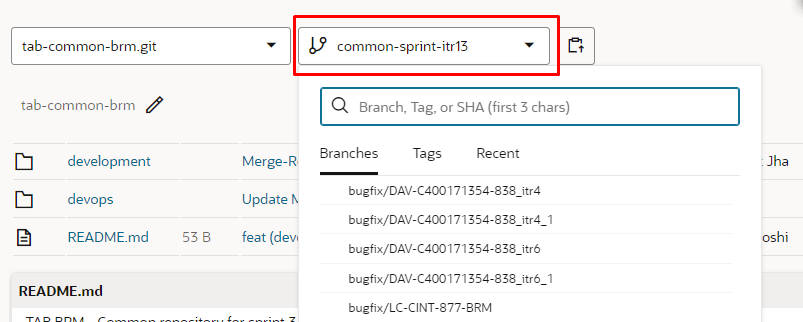
Step1: Login to ALM and select the **Telenor Asia Billing Common** Project.



Step2: Select the tab-common-brm from the listed Option.



Step3: Select the latest IRT’s Branch as given by Release Manager.



Step4: update the both Dockerfile and Makefile for CM under the following location (devop/container/cm)

Example:

Dockerfile:

ARG BASE\_IMAGE=cm

ARG BASE\_TAG=12.0.0.7.0-34916707

Makefile:

IMAGE\_NAME=common-brm-cm

IMAGE\_TAG=common-cm- 13.38.4-b002

Step5: update the both Dockerfile and Makefile for Config-jobs under the following location (devop/container/config-jobs)

Example:

Dockerfile:

ARG BASE\_IMAGE=config\_jobs

ARG BASE\_TAG= 12.0.0.7.0-34763817

Makefile:

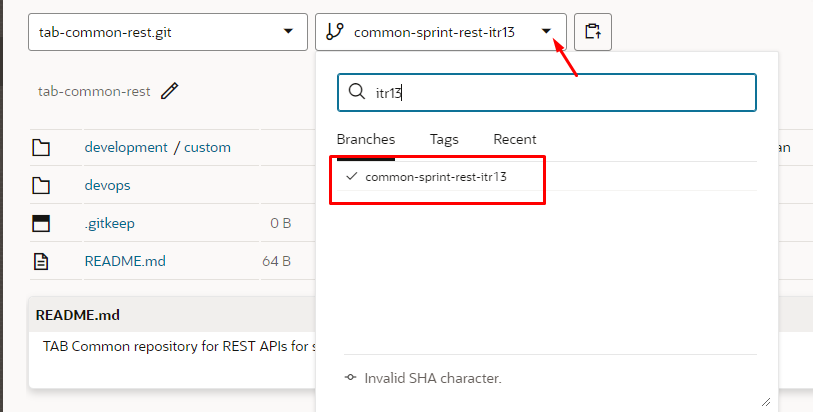
IMAGE\_NAME= common-brm-config\_jobs

IMAGE\_TAG= common-config-jobs- 13.38.4-b002

Note: After making the changes for tab-common-brm section. We’ll continue with tab-common-rest.

Step6: Select the tab-common-rest and latest ITRs option from the drop-down





Step7: update the both Dockerfile and Makefile for Config-jobs under the following location (devop/container/billingcare\_sdk)

Example:

Dockerfile:

ARG BASE\_IMAGE= oracle/billingcare

ARG BASE\_TAG= 12.0.0.7.0-34627512

Makefile:

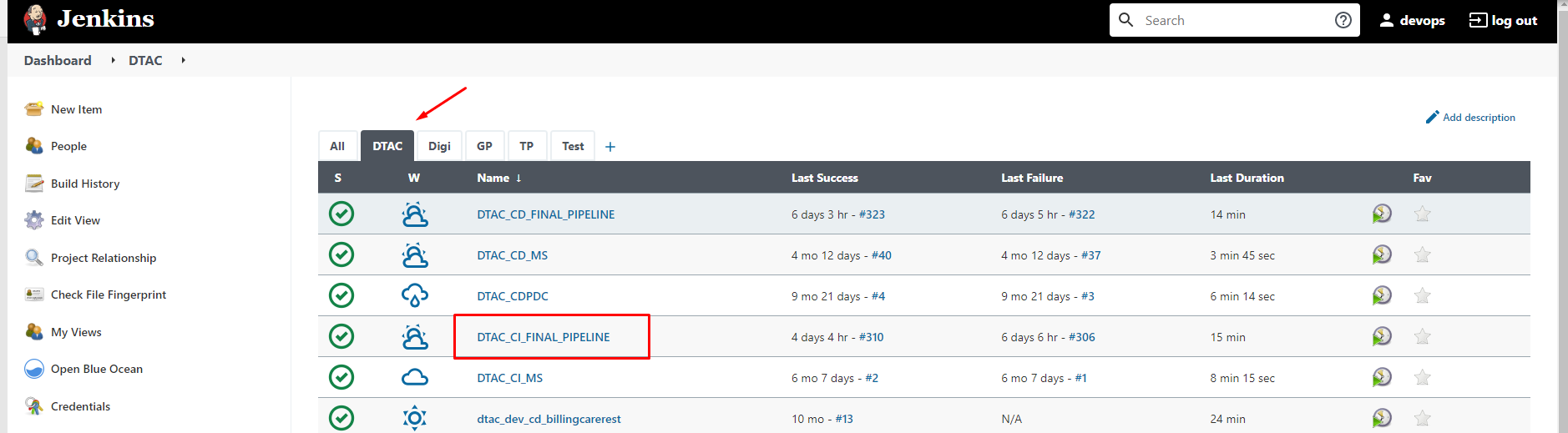
IMAGE\_NAME= common-billingcare\_sdk

IMAGE\_TAG= common-rest-13.38.4-b002

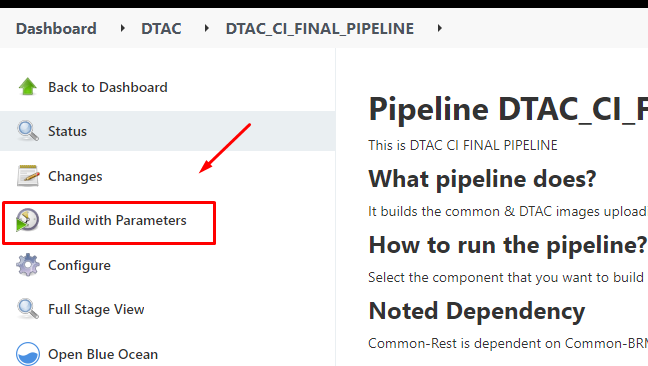
Note: Now we’ve successfully changed the latest images tags for tab-common-brm and tab-commom-rest.

But for tab-dtac-cn-brm which is Bu-Specific we’ll update the image tags directly from Jenkins CI Pipeline.

Step8: login to the Jenkins with the above mentioned URL. Click on DTAC on the Dashboard view and Select the DTAC\_CI\_FINAL\_PIPELINE

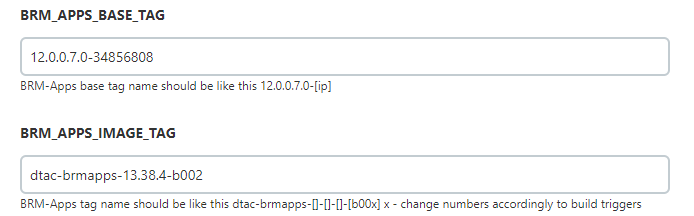


Step9: Select the Build with Parameters option to update the parameter required for CI pipeline.

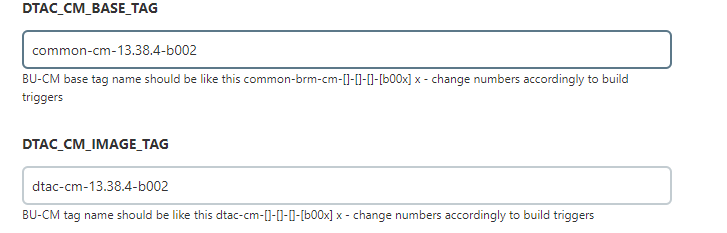


Step10: Update the Base Tag and Image Tag for tab-dtac-cn-brm section directly from Jenkins CI Pipeline

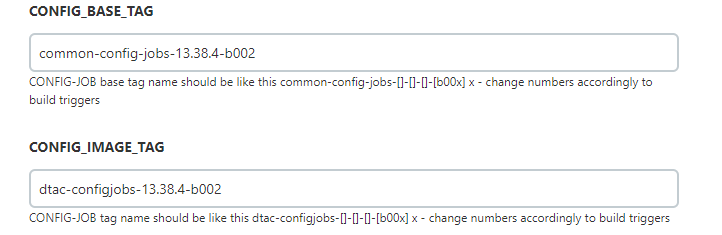
BRM-APP:



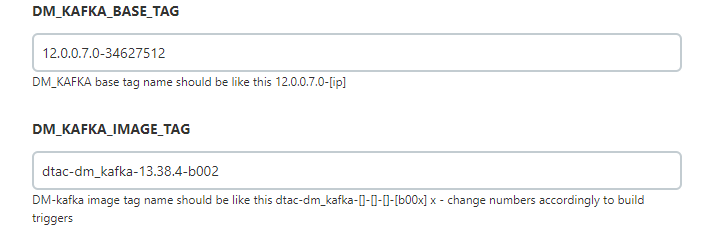
BU-CM:



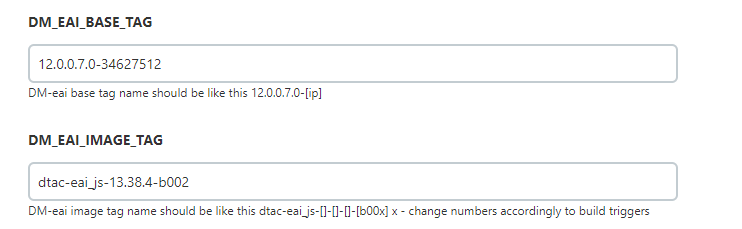
BU-Configjobs:



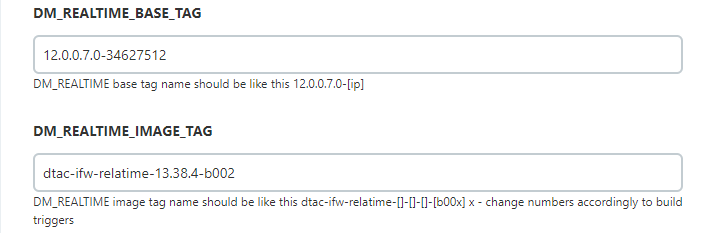
DM-kafka:



Eai-js:

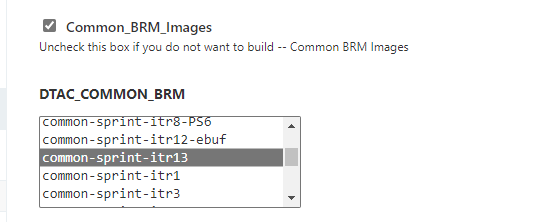


RealtimePipe:

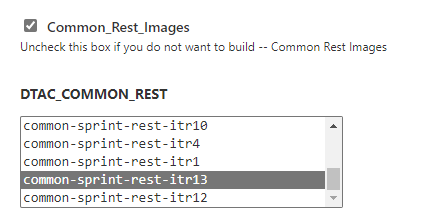


Step11: Select the latest ITR’s Bracnch and Check-in the box for which you would like to create Image.

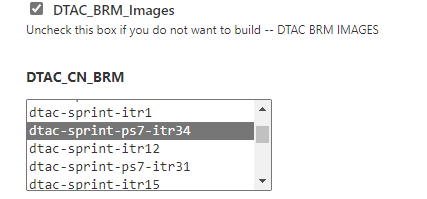
* Common\_BRM\_Images
  + Common-cm
  + Common-configjobs



* Common\_BRM\_Images
  + Custom BCWS

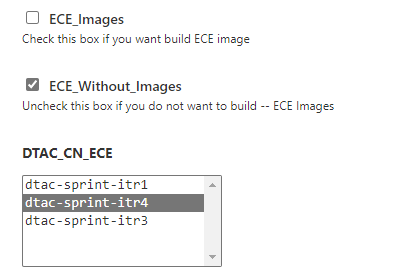


* DTAC\_BRM\_Images
  + Brm-apps
  + BU – CM
  + BU – Config-jobs
  + Dm\_kafka
  + Eai\_js
  + Realtime\_Pipe



* ECE\_Without\_Images

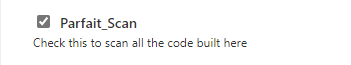
Note: If you would like to create the custom image for ECE, then click on ECE\_Images. Or For just Jars deployment. Select ECE\_Without\_images



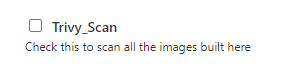
Step12: We’ve integrated other options as well with Jenkins CI Pipeline like

* + Parfait\_scan 🡪 for Code scanning
  + Trivy\_scan 🡪 for Images scanning

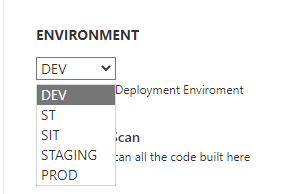
To run the Parfait, Select the Parfait\_Scan option:



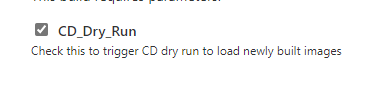
To run the Trivy, Select the Trivy\_Scan option:



Note : Make sure that you’ve selected the right Environment before triggering the Build.



And If you are going deploy the Build in DEV Environment. Select the CD\_Dry\_Run option. It will initiate the Dry run for the DTAC\_CD\_FINAL\_PIPELINE and Will be in a Ready-To-Deploy State.

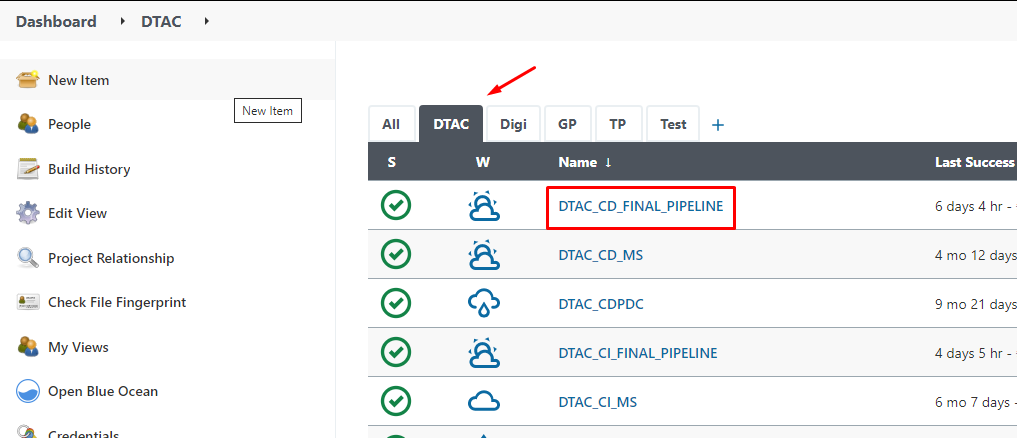


Now we’re good to trigger the DTAC\_CI\_FINAL\_PIPELINE by clicking the Build.

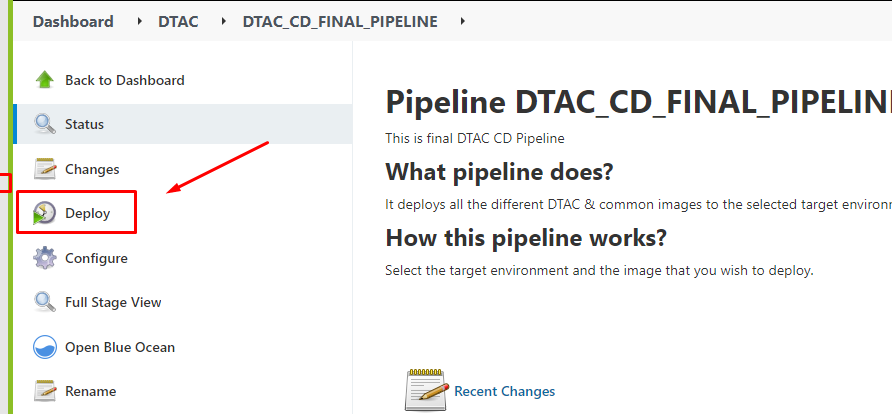


CD walkthrough: DEV

Step1: Open the DTAC\_CD\_FINAL\_PIPELINE from the DTAC Dashborad view.

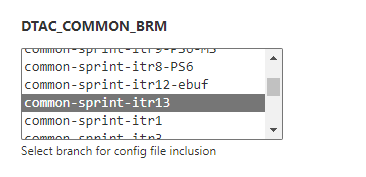


Step2: Click on Deploy option on Jenkins window.

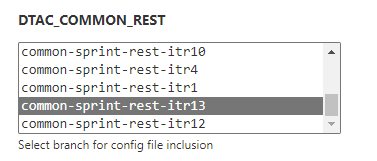


Step3: Select the latest ITR’s for the below sections:

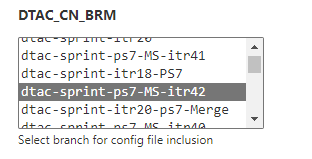
tab-common-brm:



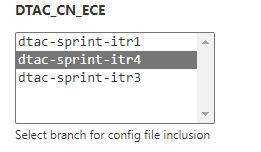
tab-common-rest:



tab-dtac-cn-brm:

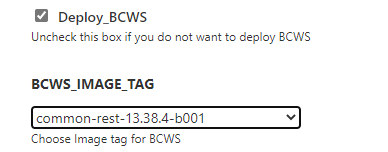


tab-dtac-cn-ece:

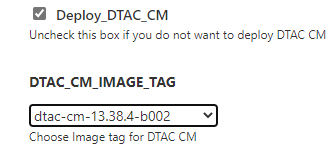


Step 4: Select the latest image tag for the below Applications.

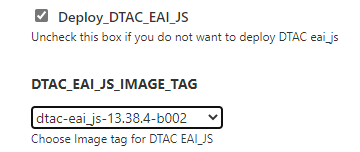
BCWS:



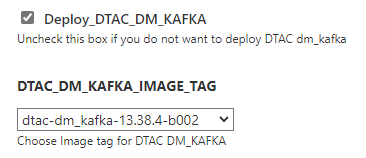
CM:



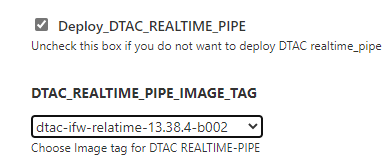
EAI JS:



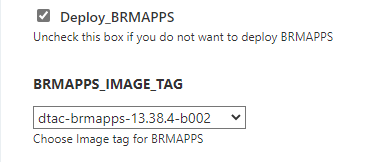
DM\_Kafka:



Realtimepipe:



Brm\_Apps:



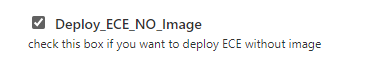
Dtac\_Config:



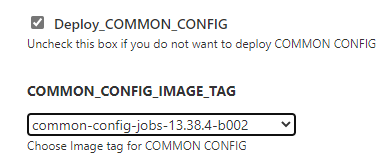
ECE:



ECE\_No\_image:



Common\_config:



After Selecting the Image tag click on Build to trigger the CD Pipeline.

CD walkthrough: Client Environments.

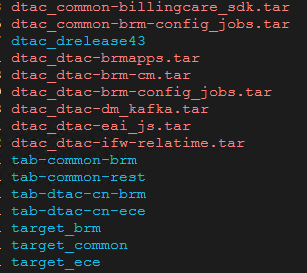
## CD Prerequisites: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application** | **Environment** | **URL** | **Username** | **Password** |
| Jenkins | **SIT & STG** | **https://brm-tb-cicd.dtacnetwork.dev:8080/** | **admin** |  |
| **Prod** | https://brm-cicd.dtacnetwork.co.th:8090/ | **admin** |  |

Step1: Follow the below steps to take the custom Release to the client environment.

* Create the directory in local
  + mkdir dtac\_drelease38 (note: 38 denotes the release number)
* Copy the folders from local Jenkins workspace location to the newly created directory.
  + cd workspace\_location
  + cp –r tab-common-brm tab-common-rest tab-dtac-cn-brm tab-dtac-cn-ece target\_brm target\_common target\_ece path/dtac\_drelease38
  + use the docker save command to save the image in a tar file.
    - docker save -o dtac\_dtac-brm-cm.tar dtac-brm-cm:dtac-cm- 13.38.4-b002
    - docker save -o dtac\_common-brm-config\_jobs.tar common-brm-config\_jobs:common-config-jobs- 13.38.4-b002
    - docker save -o dtac\_common-billingcare\_sdk\_stg.tar common-billingcare\_sdk:common-rest- 13.38.4-b002
    - docker save -o dtac\_dtac-brm-config\_jobs.tar dtac-brm-config\_jobs:dtac-configjobs- 13.38.4-b002
    - docker save -o dtac\_dtac-brmapps.tar dtac-brmapps:dtac-brmapps- 13.38.4-b002
    - docker save -o dtac\_dtac-dm\_kafka.tar dtac-dm\_kafka:dtac-dm\_kafka- 13.38.4-b002
    - docker save -o dtac\_dtac-eai\_js.tar dtac-eai\_js:dtac-eai\_js- 13.38.4-b002
    - docker save -o dtac\_dtac-ifw-relatime.tar dtac-ifw-relatime:dtac-ifw-relatime- 13.38.4-b002

Your folder structure will look like:

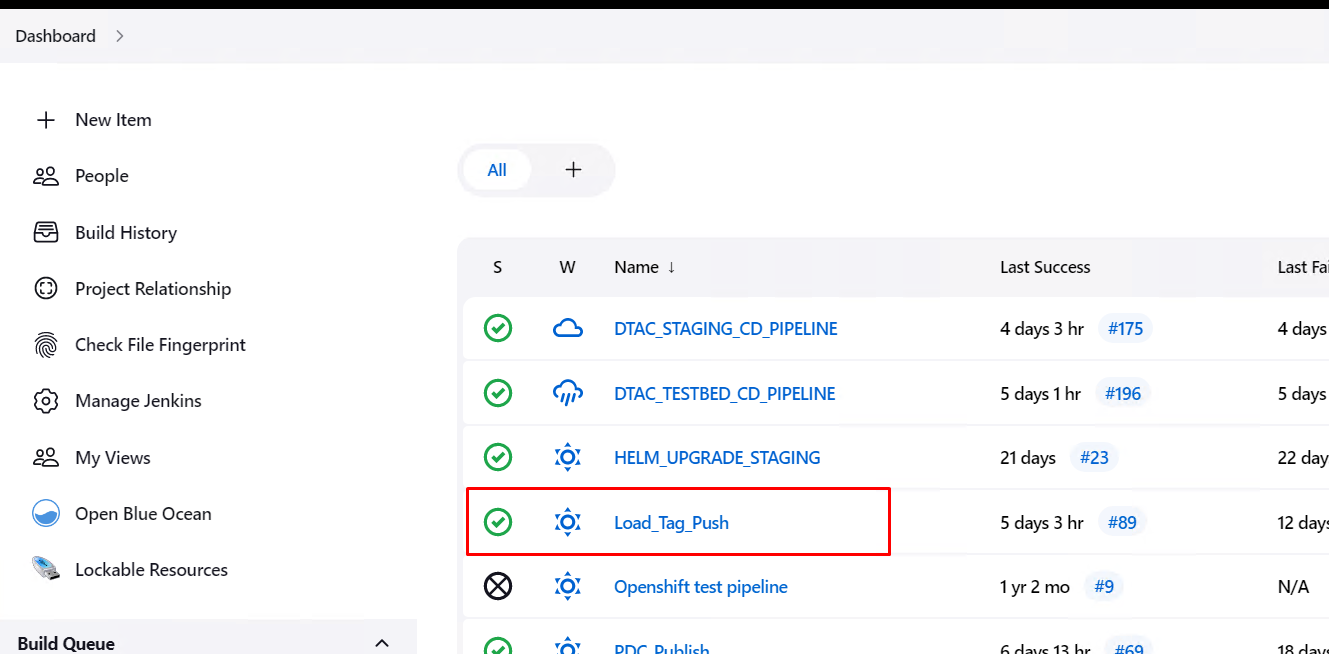


* make the tar of newly created folder which contains workspace’s folders and image tar file.
  + tar –cf dtac\_drelease38.tar dtac\_drelease38
* Use curl commad to push the tar file to OCI Storage.
  + curl -X PUT --data-binary '@dtac\_drelease38.tar' https://objectstorage.uk-london-1.oraclecloud.com/p/hHdw5ayWjHnNTO7ctCsppUgzkh2piRtff8Sh\_kgRH9pcNdshjnadQ1krgFMjb2iX/n/idvvfekwvhut/b/tab\_dtac\_lhrbkt/o/dtac\_drelease38.tar
* After It got uploaded successfully. Take the URL from the above curl command and send it to DTAC via mail. DTAC will download the file in UTL server.
  + https://objectstorage.uk-london-1.oraclecloud.com/p/hHdw5ayWjHnNTO7ctCsppUgzkh2piRtff8Sh\_kgRH9pcNdshjnadQ1krgFMjb2iX/n/idvvfekwvhut/b/tab\_dtac\_lhrbkt/o/dtac\_drelease38.tar
* DTAC will confirm the location of downloaded file in UTL server over the mail.
  + Then we need to connect to UTL server, Then upload the file to CICD server with the help of winscp.
  + We need to place the tar file inside the $REPO\_HOME location. Before that we need to make sure REPO\_HOME is Empty.
  + Place only our tar file inside the REPO\_HOME directory.

Example:



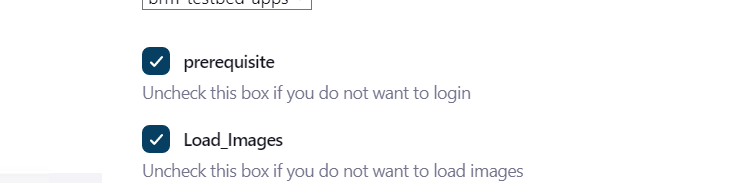
Step 2: Login to Jenkins using the below URL. And Open the Load\_Tag\_Push Pipeline.



Step3: Uncheck all the boxes except {prerequisite & Load\_Images}.

* + Prerequisite 🡪 It will untar the file (dtac\_drelease38.tar)
  + Load\_Images 🡪 It will load the images locally so that later we can tag the image and push to OCR.

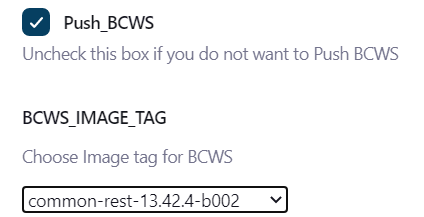
Then trigger the Build.



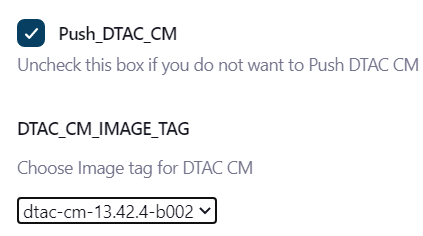
Step4: After the completion of Pipeline. Open the pipeline job again and Uncheck all the boxes and then trigger the Build. (DRY\_RUN).

Step5: After the completion of DRY\_RUN. Open the pipeline job again and Uncheck the {prerequisite & Load\_Images}. Then check and select the boxes and corresponding image tag.

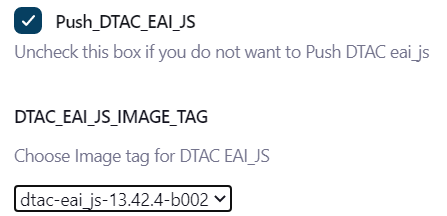
BCWS:



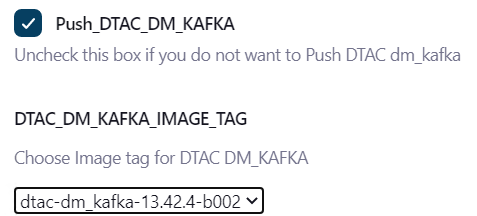
CM:



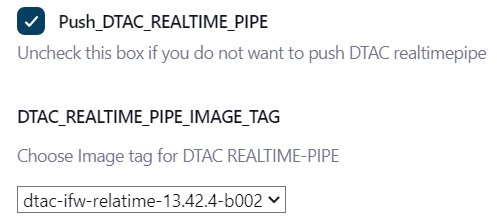
EAI:



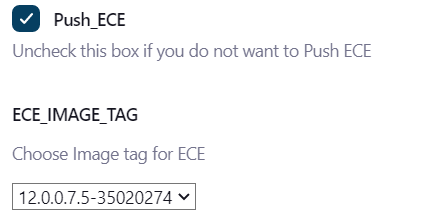
DM-Kafka:



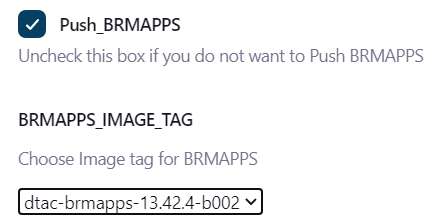
Realtime:



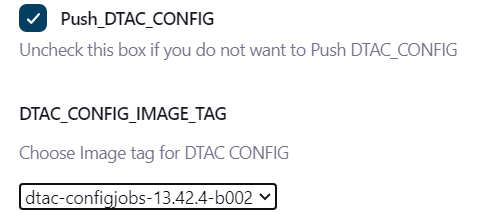
ECE:



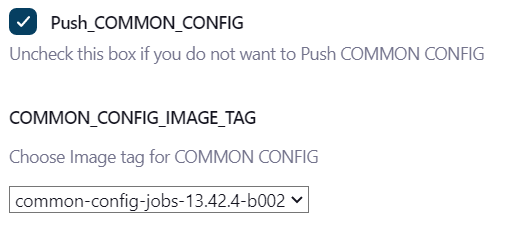
BRM\_Apps:



DTAC\_Config:



Common\_config:

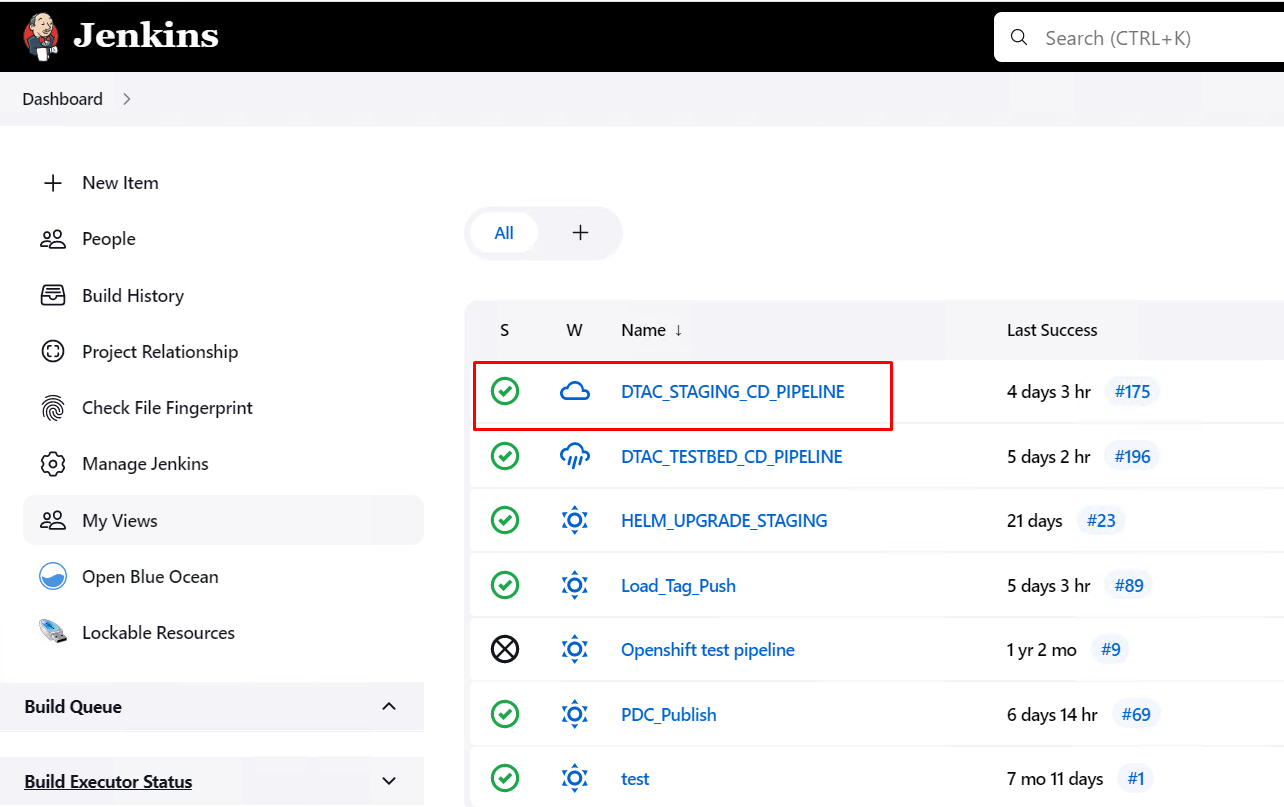


Make sure that you’ve selected the latest image tag from the drop down. And Click on Build to TAG and PUSH the latest image to OCR.

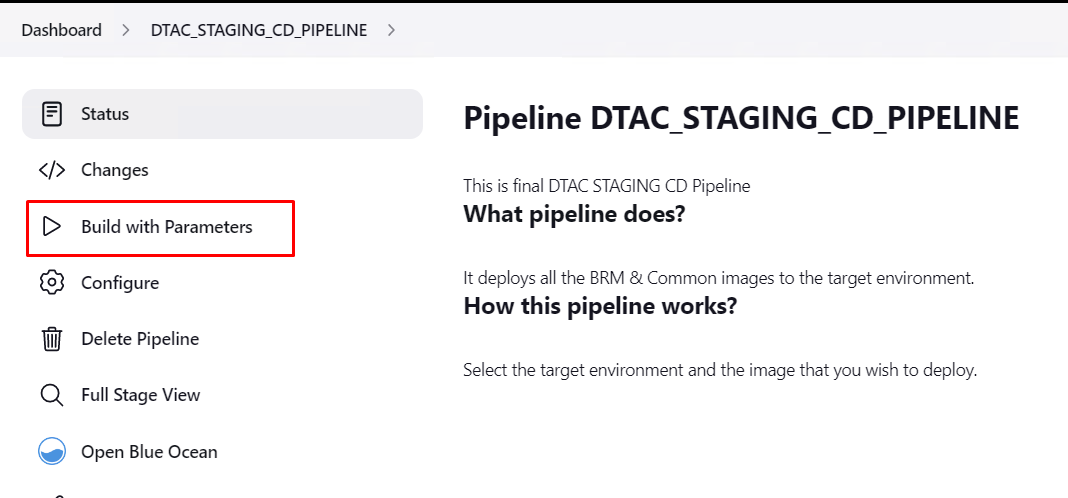
CD:

## Note: Perform the below steps after the completion of CD Prerequisites.

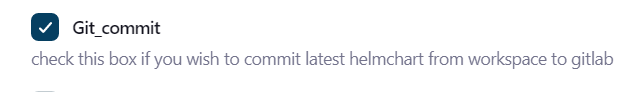
Step1: open the Pipeline job



Step2: Click on the Build with Parameters option to update the parameter required for CD pipeline.



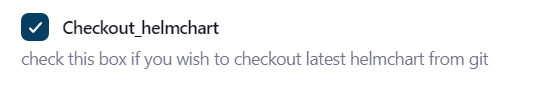
Step3: Check whether your Git has the latest helmchart or Local. If Local has the latest then check the Git\_commit option alone. And Trigger the CD pipeline.



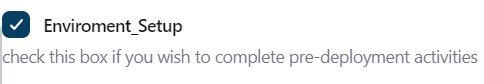
Note: Maintaining the latest helm chart in GitLab is a Good practice.

Step4: check the Checkout\_helmchart ,Environment\_Setup and Disable\_Cronjob options then Trigger the Build.

Checkout\_helmchart: To clone your helmchart from GitLab to Local



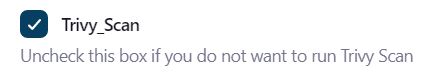
Enviroment\_Setup: Creates the folder structure for the deployment and other pre-deployment activities.



Disable\_Cronjob: To disable the running cron-jobs from the server. Then Trigger the first round of build of CD Pipeline.

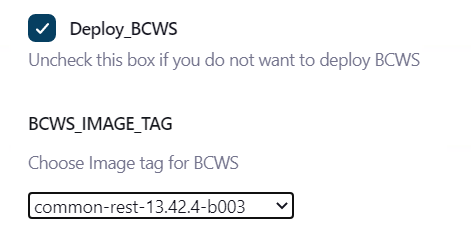
Note: After the completion of first round of build. Reopen the CD pipeline job and follow the below procedures by clicking on Build with Parameters.

Step5: Check the Trivy\_Scan if you want to scan the image before its get deployed.

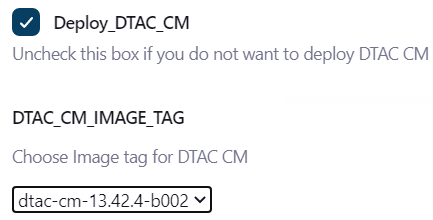


Step6: Check and select the images that needs to be deployed.

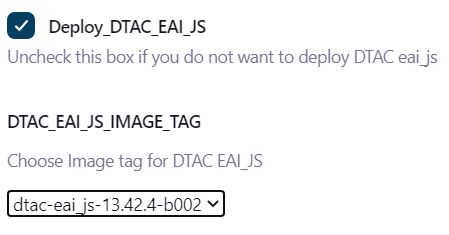
BCWS:



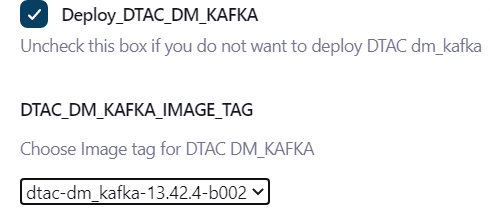
CM:



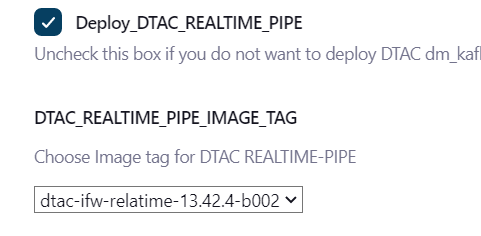
EAI:



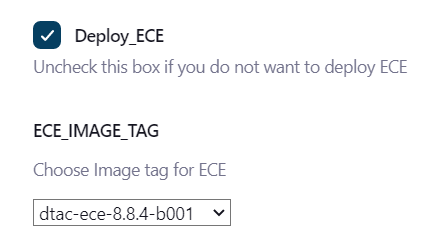
DM-Kafka:



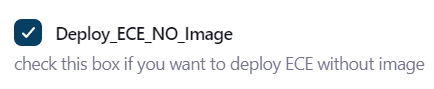
RealtimePipe:



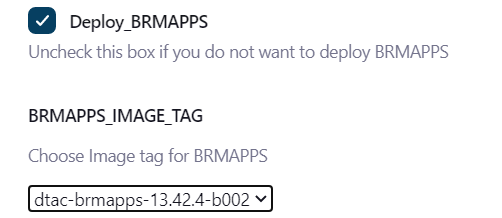
ECE:



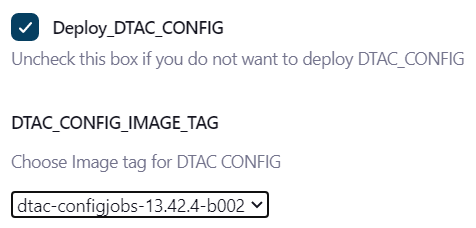
Note: If there is no custom image deployment and need to load the latest jars to ECE then select the below option.



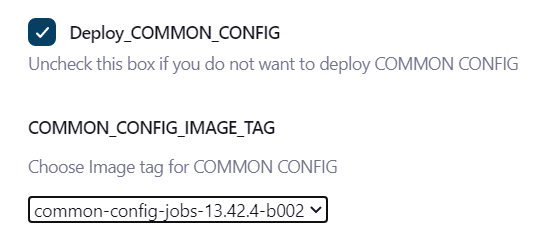
PDC:

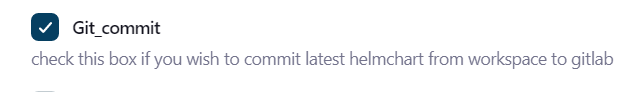


DTAC\_Config:



Common\_Config:

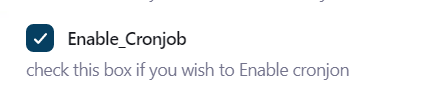




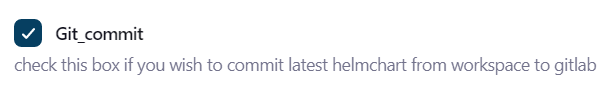
Step 7: Make sure that we’ve selected all the image tag properly and Trigger the Build.

Step 8: After the successful completion of second build. Open the Jenkins Job and Run the following Stages:

Enable cronjob: It will start the cronjob for the available cron entries.



Git\_commit: It is used to commit the latest modified helmchart from workspace to GitLab.



Note: we’ve one more option integrated with Jenkins i.e Gitlab\_backup Select the option to take the backup of entire GitLab if required.