

Webhooks Deployments

**API Container Deployment:**

1. Firstly, get the code from the dev team, it will be in the form of a **ZIP File** where we have python scripts in it.
2. Then login to the **POC jump server (169.63.177.113),** we have a docker file already written to build an image.
3. The docker file is present in the path **/opt/FEDEX/ITX/webhooks-api**
4. Now delete the zip which is already present in the path and upload the zip file to the path **/opt/FEDEX/ITX/webhooks-api .**
5. Now unzip the file and you will get a folder from it.
6. Now copy the **db2jcc4.jar and runcontainer.sh** to the folder which is extracted from the ZIP folder and rename the old, unzipped file to the next number accordingly.
7. Now from the webhooks-api directory build the image with the command:

**docker build -t webhooks-api:<tag> .**

1. Now login to the registry us.icr.io with **“docker login us.icr.io”** , Next will prompt to the API username press enter and it will ask for password. The password is:

**9G-FOmu7XT-FEgVmGluFuzyxWOVyJtCxZ5FZYp4kz4VP**

1. Now tag the image which is built with docker tag.  
     
   **docker tag webhooks-api:<tag> us.icr.io/webhooks- nonprod/webhooks-api:<tag>**
2. Now push the image to the registry  
     
    **docker push us.icr.io/webhooks- nonprod/webhooks-api:<tag>**
3. Now after building the image go to the path **/opt/FEDEX/ITX** , you will find the **api-container.yaml** file.
4. Make the image change in the YAML file which is built recently and save the file.
5. Now be on the same project where the API container is present and execute the below command.  
     
    **oc apply –f api-container.yaml**
6. Then type oc get pods for the API pod to pass from container creating to the running state. And the pod which is their earlier will be terminated.

**NOTE: Please mention the proper TAGS while building an image.**

**Deployment of Delivery Pause and redelivery**

1. Firstly, get the python scripts ZIP file from the DEV team and upload the ZIP file to the POC Jump server  **(169.63.177.113)**
2. There will be Two ZIP files which is for delivery pause and redelivery.
   1. Delivery Pause will contain 10 python Script files which are 6 Advanced scripts and 4 Basic Scripts.
   2. Redelivery will contain 3 python files which are Main Basic, Main Advanced and pause and resume.py files.
3. Now upload the Zip file to the path **/opt/FEDEX/ITX/redelivery/config** delete the existing Python file I.e., .py files and unzip the folder which is uploaded.
4. Move the files from the Unzipped folder to the path **/opt/FEDEX/ITX/redelivery/config** and delete the zip file and unzipped folder from the path.
5. If you have scripts other than 6 advanced and 4 basic scripts, then move Main Basic and Main Advanced to the path to **/opt/FEDEX/ITX/redelivery/PR** or else ignore.
6. Before copying the Main Basic and Advanced to the PR directory delete the scripts in the PR and then copy.
7. Now go the path **/opt/FEDEX/ITX/redelivery** and build a docker image where the docker file is already written.  
     
   **docker build -t us.icr.io/webhooks-nonprod/redelivery:<tag> .**
8. Now login to the registry of “us.icr.io” with docker login us.icr.io and it will prompt to username press enter and then asks for password.  
     
   9G-FOmu7XT-FEgVmGluFuzyxWOVyJtCxZ5FZYp4kz4VP
9. Now push the image to the registry with  
      
   **docker push us.icr.io/webhooks-nonprod/redelivery:<tag>**
10. If you are deploying the image in dev, then switch to the   
    **fedex-dev-itx ,** for the deployment of python-delivery-pause, go to the python-delivery-pause deployment file and edit the name and tag and save the file.  
      
     **oc edit deployment python-delivery-pause**
11. Now the monitor the pods that the old pods will be terminating, and the new pods will be in container creating and should be running state.
12. Same as if you are deploying the image in QA and prod, go the namespace and edit the deployments file with the image name that needs to be updated, then save the file.

**Note:**

If you are deploying the image in the prod, pull the image to the **prod jump server 150.239.80.250** which is already built in the **POC server** and login to the registry and tag with the prod namespace which is **docker tag us.icr.io/webhooks-nonprod/redelivery:<tag> us.icr.io/webhooks-prod/redelivery:<tag>**   
and then push the image to the registry with **docker push us.icr.io/webhooks-prod/redelivery:<tag>.** Now edit the image in the prod deployments wherethe image needs to be updated.

**Maps Deployment**

1. Get the maps from the Maps team and upload the maps to the respective jump server and upload to the respective pod which is instructed.  
     
    **oc cp <filename> <pod name>:/data/maps/**  
     
   Note: The maps should be deployed in the RS pod and should be deployed in the path **/data/maps/**
2. Jars files needs to be updated in the path **/data/extra/**