## **Introduction:**

By using CICD pipelines we are deploying pod to Kubernetes cluster.

## **Prerequisites:**

* A version control system (Azure Git repo).
* A CI/CD tool (Azure DevOps).
* Service Connection for Kubernetes Cluster.
* Self-Hosted agent.
* Variable Group: - "We have used a Variable Group to securely store the username and password for pulling images from the Azure Container Registry (ACR)."
* Helm charts.

**Implementation:**

1. Created new repository with name of **ECM\_comp\_Exstream\_dev** in ECM\_Exstream project under Organization Nedbank-Limited.
2. Created a yaml pipeline (Pull\_&\_Push.yml) to place the Images in ACR **“nsnakscontregecm001.azurecr.io”**.
3. Extracted Tgz files (**Opentext-experience -cloud 24.4.0.tgz and exstream-24.4.6+59452.tgz).**  which were provided by client. Named it as experience\_values.yaml and exstream\_values.yaml .
4. Placed the yaml file into the repository with appropriate values.
5. After that another yaml file is related to SecretProviderClass is placed into the same repo.
6. Created two Yaml files with the name of helm\_install.yaml and exstream\_install.yaml for CICD pipeline setup to deploy pods.

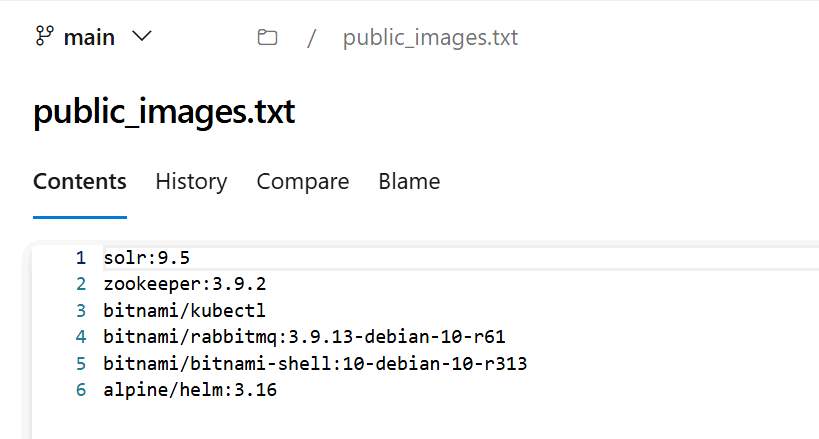
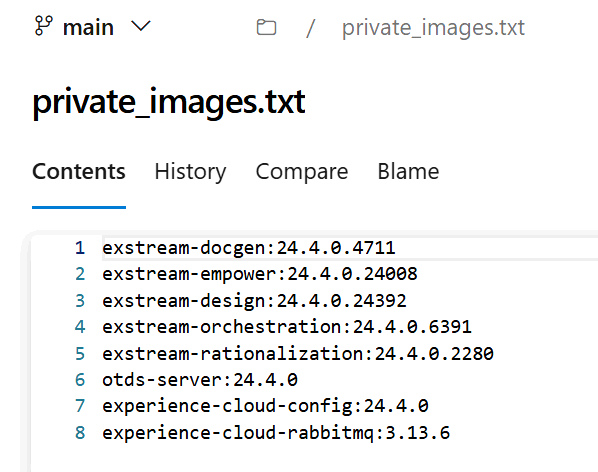
**Detailed Implementation:**

1.Repo Creation:

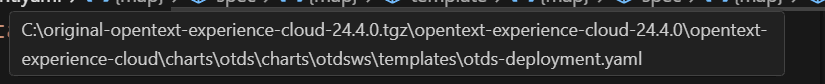
A dedicated repository, **ECM\_comp\_Exstream\_dev**, has been created to facilitate streamlined development and collaboration. This repository will serve as the central source of truth for all changes, ensuring versioning, tracking, and secure storage of code and configuration files.

Link:-

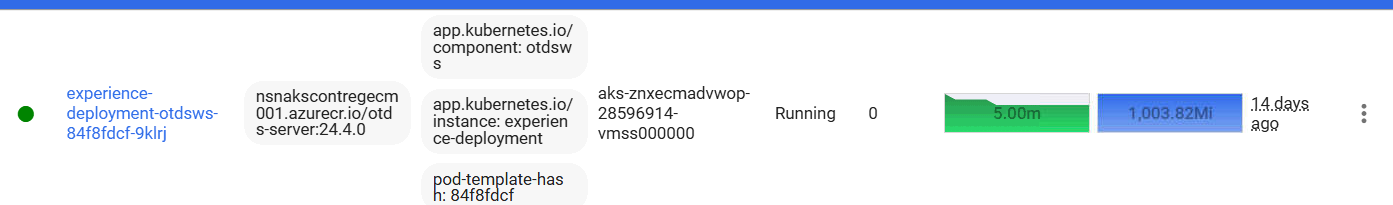
2. Placed the Images in ACR “nsnakscontregecm001.azurecr.io”.

* Created the public and private text files in repo **ECM\_comp\_Exstream\_dev.**
* In the public text file we have 6 images below is the screen shot for reference.
* 
* In private text file we have 8 images below is the screen shot for reference.
* 
* Created a pipeline (Pull\_&\_Push.yml) to Pull and Push the Images to ACR (**“nsnakscontregecm001.azurecr.io”**).
* Link:-
* Once successfully pushed to the ACR moved to the next step.

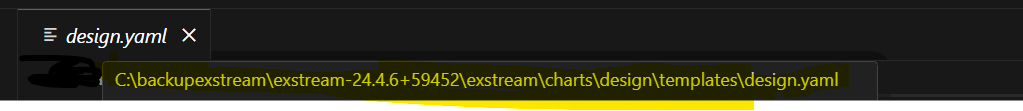
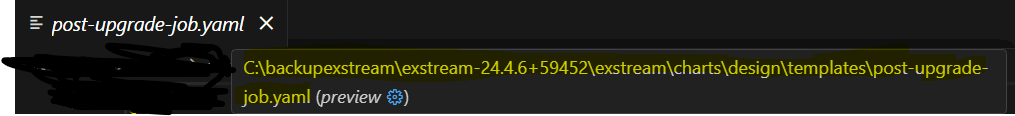
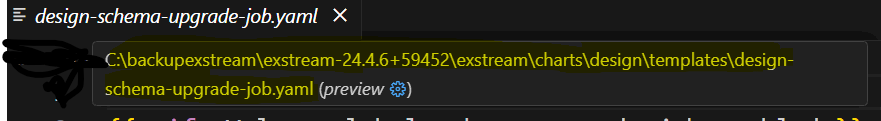
3. Extracted Tgz files (**Opentext-experience -cloud 24.4.0.tgz).**

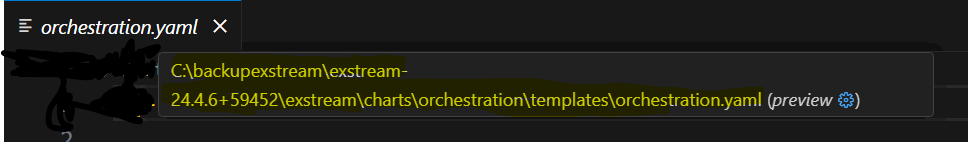
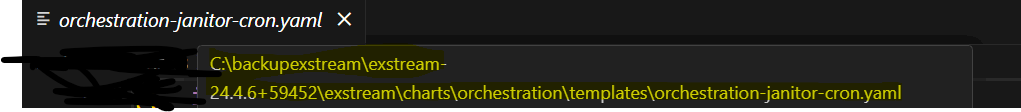
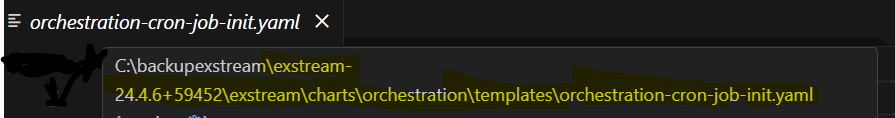
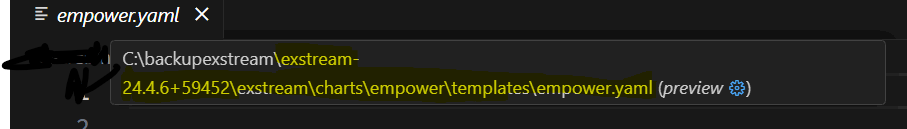
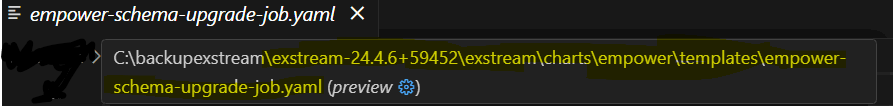
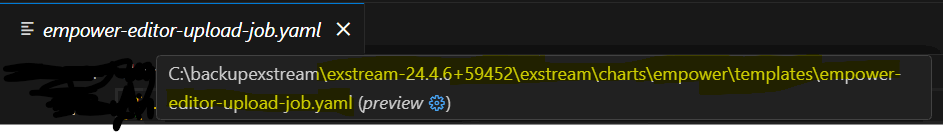
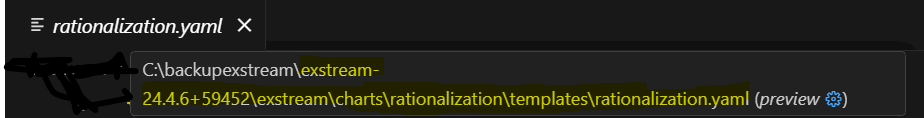
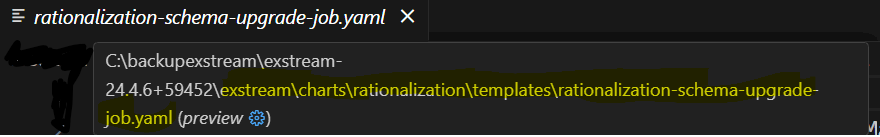
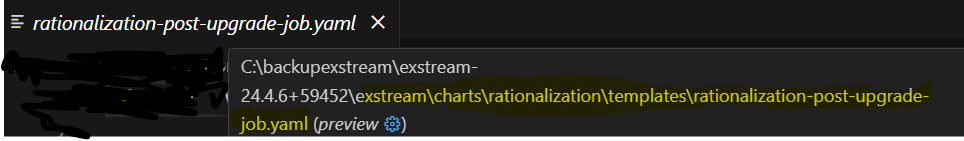
* Extracted the **Opentext-experience -cloud 24.4.0.tgz** in that extracted file we can see multiple folders and files, but we need to focus only on values.yaml.
* We are creating yaml file named experience\_values.yaml by using values.yaml file.
* In the extracted file (**Opentext-experience -cloud 24.4.0.tgz**) we need to update few parameters and again repackage(by using the command helm package exstream) it to **Opentext-experience -cloud 24.4.0.tgz** file and upload it in repo.
* Updating parameter's location referred screen shot below.
* 

4. SecretProviderClass is placed into the same repo.

* We need to create a another yaml file SecretProviderClass.yaml
* Now created Yaml pipeline (helm\_install.yaml) to deploy otds and RabbitMQ pods.
* Once the pipeline is successfully deployed the pods are running up.
* Below are screenshots for reference.
* 

5.Extracted Tgz file **exstream-24.4.6+59452.tgz**

* Extracted the **exstream-24.4.6+59452.tgz** in that extracted file we can see multiple folders and files, but we need to focus only on values.yaml.
* We are creating yaml file named exstream\_values.yaml by using values.yaml file.
* In the extracted file (**exstream-24.4.6+59452.tgz**) we need to update a few parameters and again repackage(by using command helm package exstream) it to **exstream-24.4.6+59452.tgz** file and upload it in repo.
* Exstream consists of multiple schemas, each serving a specific function:
* **Design**
  + Used for **batch processing** and **on-demand execution**.
* **Orchestrator**
  + Handles **workflow orchestration** and **job scheduling**.
* **Empower**
  + Includes:
    - **Shared Schema**: Commonly used across various modules.
    - **System Schema**: Stores system-level configurations.
* **Rationalization**
  + Manages **optimization and data rationalization**.
* We must update the required parameters in the below files. Please refer below screenshots for file locations.
* **Design**
* Design.yaml
* 
* post-upgrade-job.yaml
* 
* design-schema-upgrade-job.yaml
* 

* **Orchestration**
* Orchestration.yaml
* 
* orchestration-janitor-cron.yaml
* 
* orchestration-cron-job-init.yaml
* 
* **Empower**
* Empower.yaml
* 
* empower-schema-upgrade-job.yaml
* 
* empower-editor-upload-job.yaml
* 
* **Rationalization: -**
* Rationalization.yaml
* 
* Rationalization-schema-upgrade-job.yaml
* 
* Rationalization-post-upgrade-job.yaml
* 

. SecretProviderClass is placed into the same repo.

* We need to create a another yaml file SecretProviderClass.yaml.
* Now created Yaml pipeline (exstream\_install.yaml) to deploy Design, Orchestrator, Empower and Rationalization pods
* Once the pipeline is successfully deployed the pods are running up.
* Below are screenshots for reference.
* 