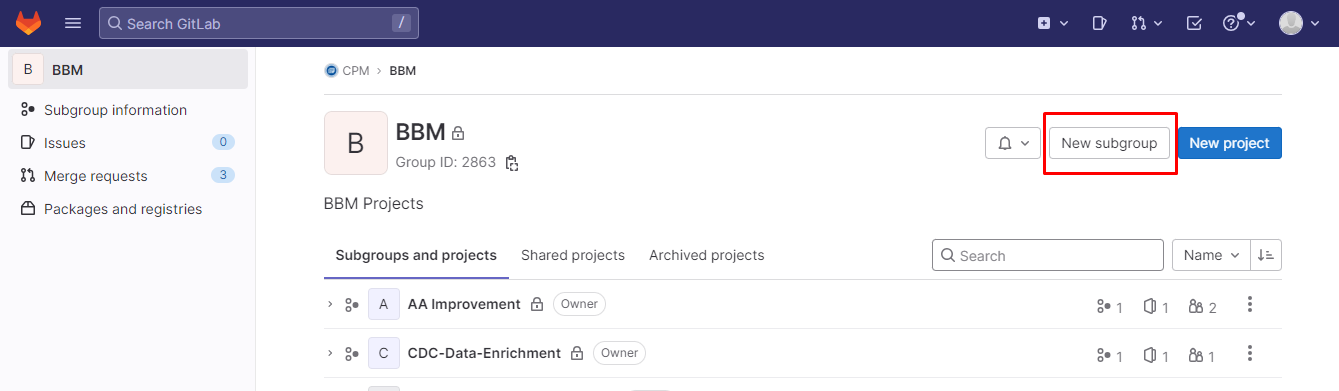
**Environment Setup Doc**

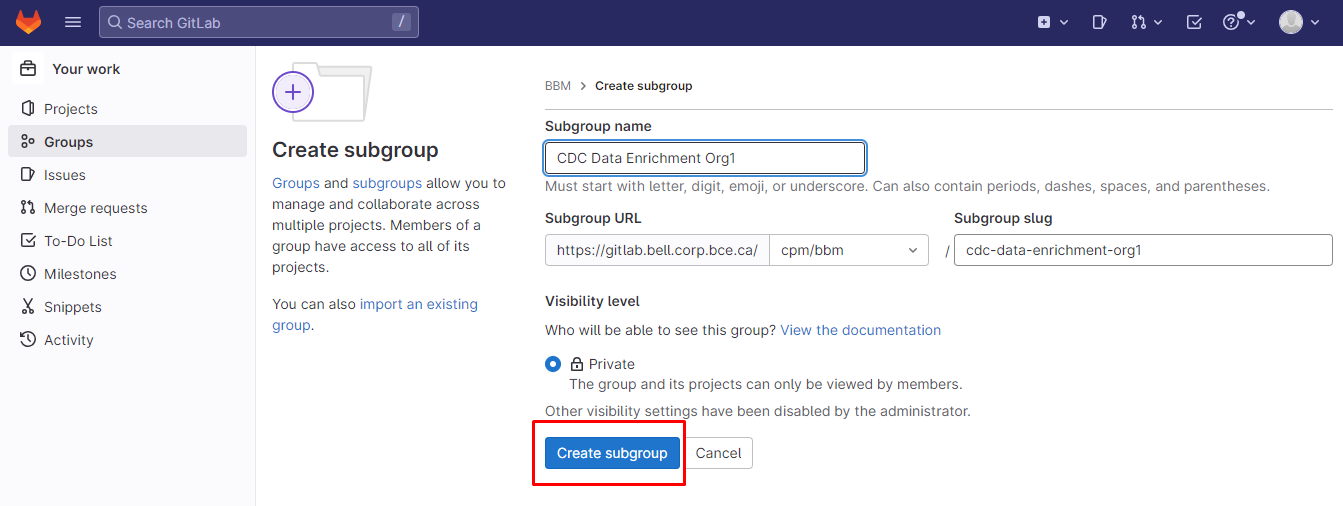
Git Repo (code):

Git: <https://gitlab.bell.corp.bce.ca/cpm>

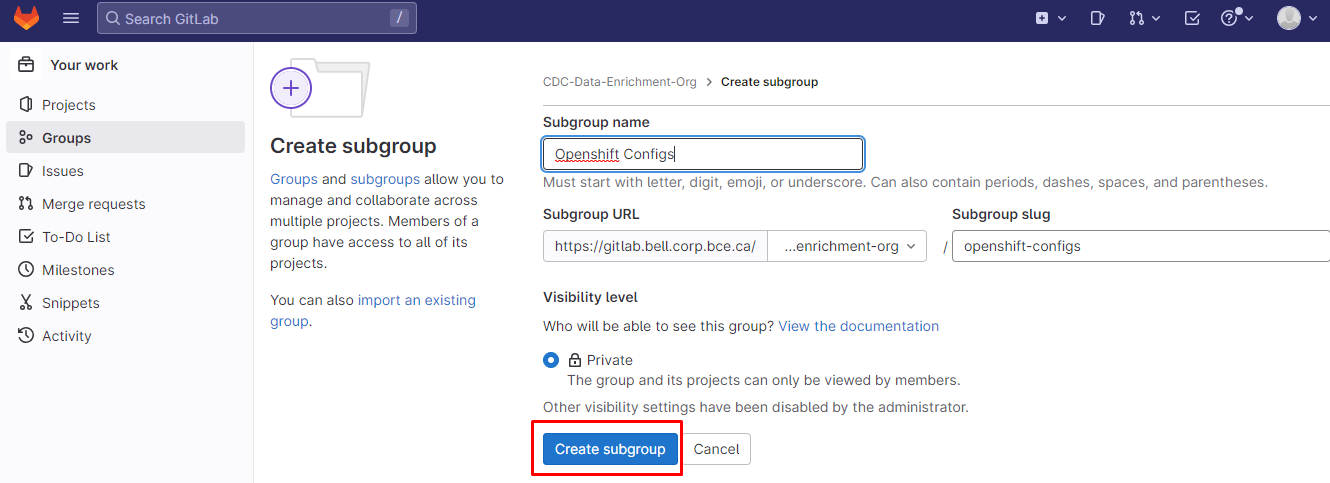
1. Move to the respective branch. Since this project is on BBM. We are moving to BBM branch.
2. Create New Subgroup as show in below screenshot
3. Select new subgroup and create a new subgroup for the project.



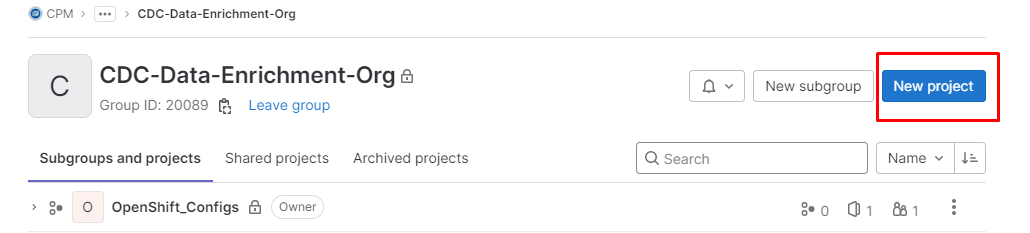
* Enter you **Subgroup Name** and click on **Create subgroup.**



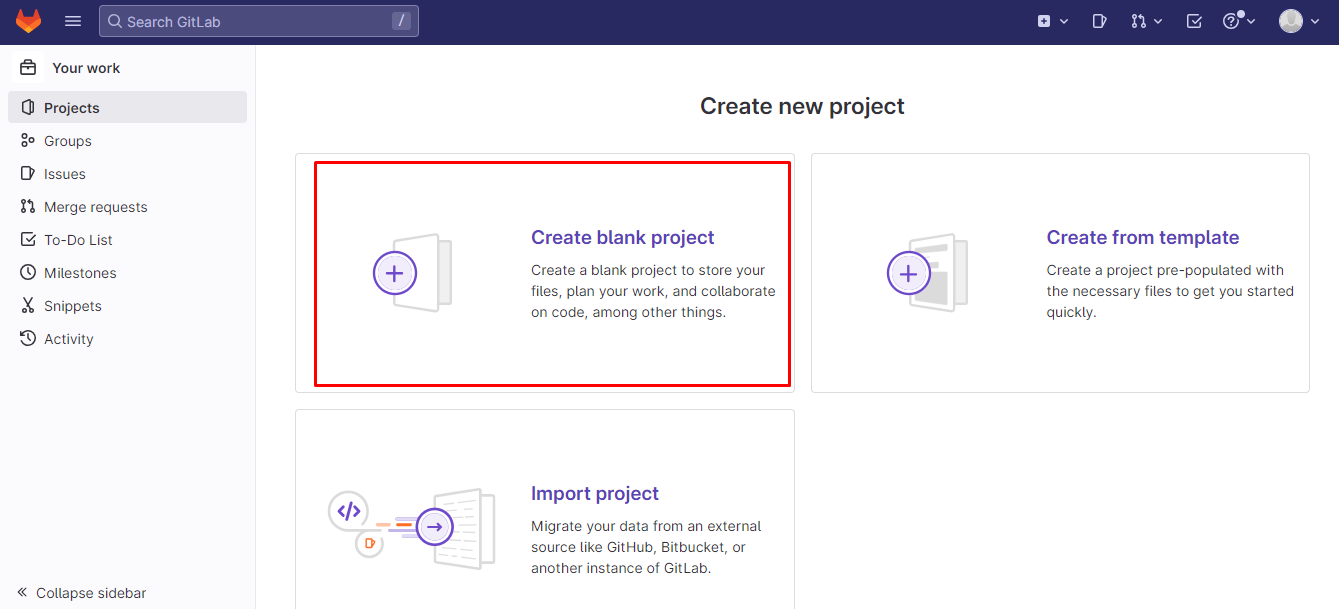
* Click on **Newly Created Repo** and again Click on **New Subgroup** to create **Openshift-config** Repo.
* Fill the **Subgroup Name** and click on **Create Subgroup**.



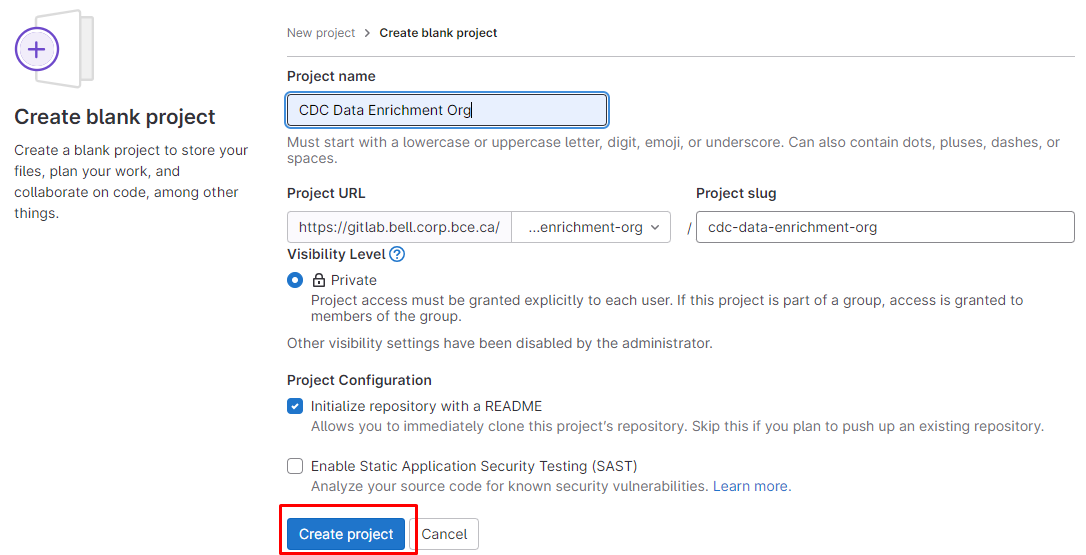
* Click on **New Project** and fill the details



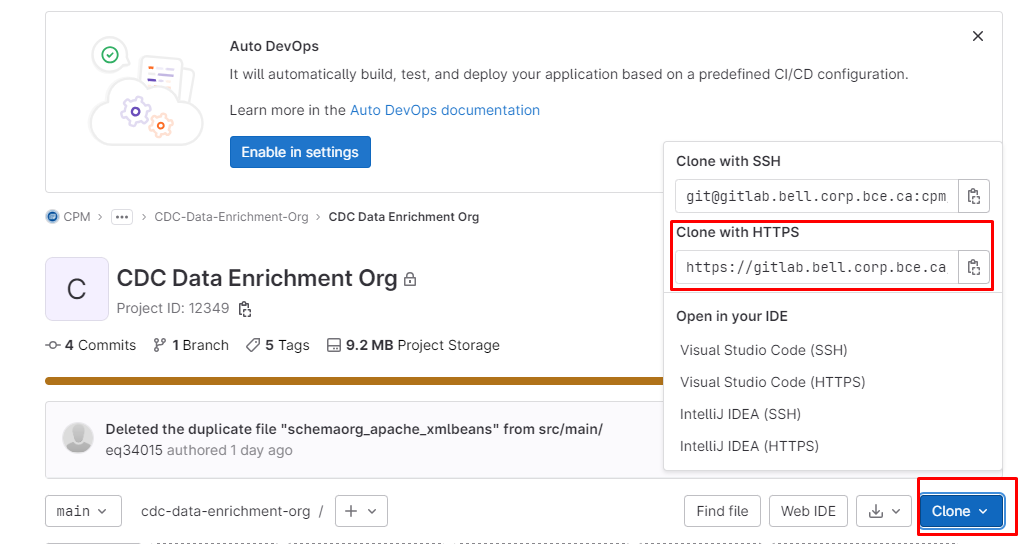
* Click on **Create Black Project**
* Enter the Project name.
* Check **“Initialize repository with README”**.



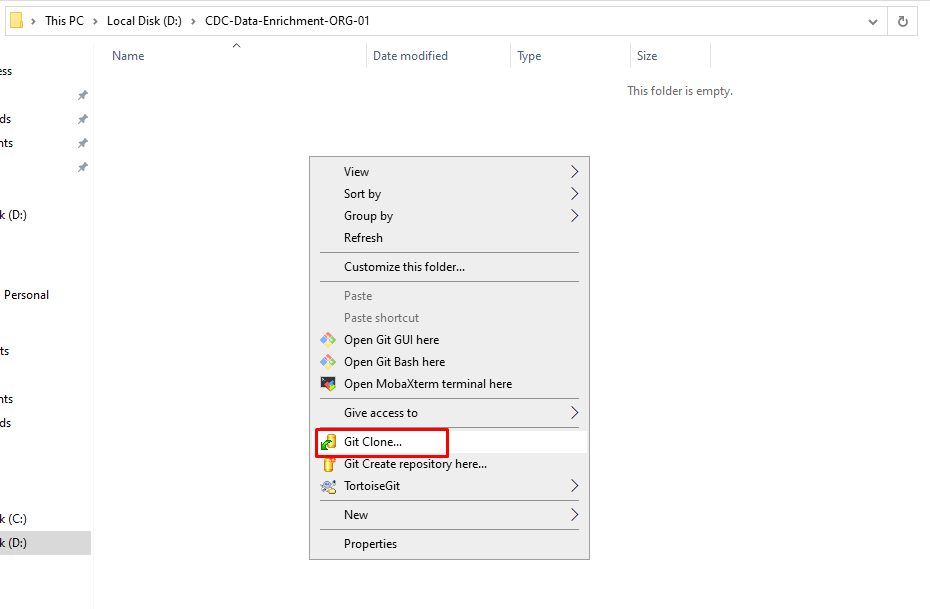
* Fill the details as shown below.



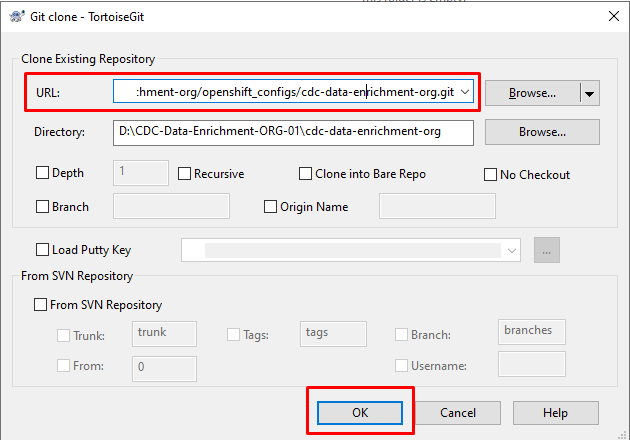
* Click on clone and copy the clone with HTTPs URL and send it to the respectively SPOC.



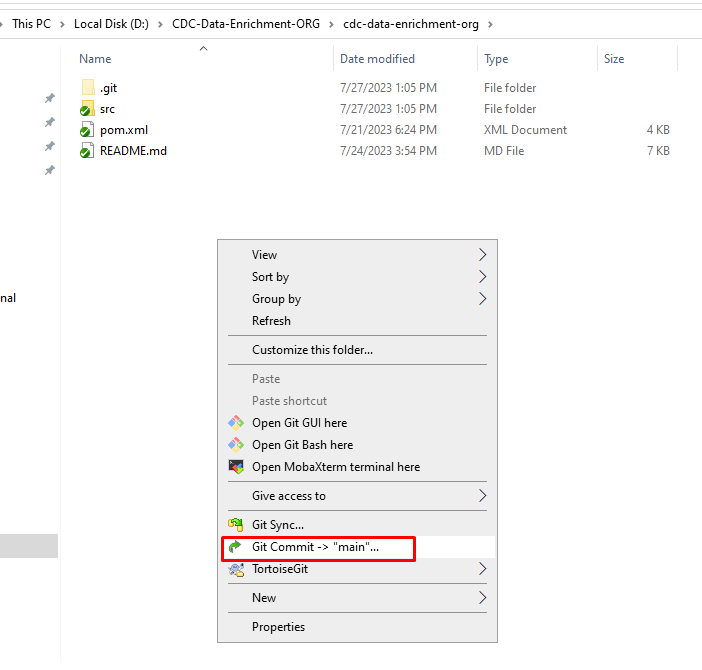
* Create an Empty directory with your Project name and Right click --> Git Clone the project Repo using Tortoise as shown below



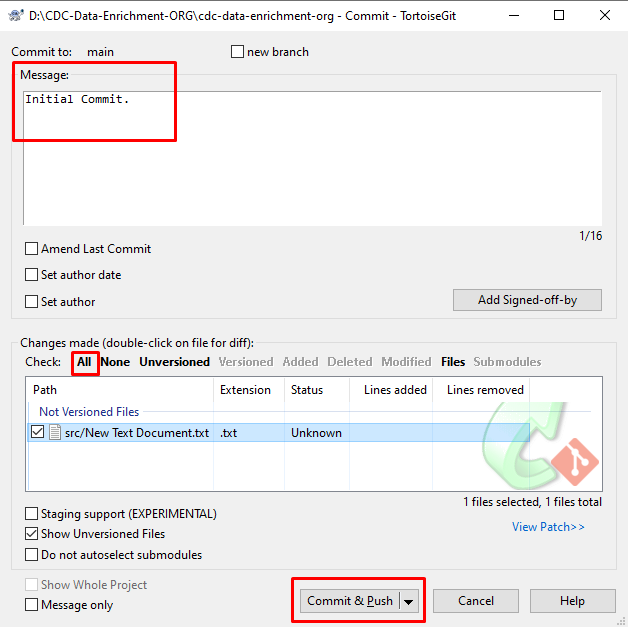
* Paste the Git clone HHTPS URL as shown below and click **OK**
* It will start cloning the Empty Repo
* Paste your Project's src and pom.xml in cloned Repo



* Do Git Commit as shown bellow



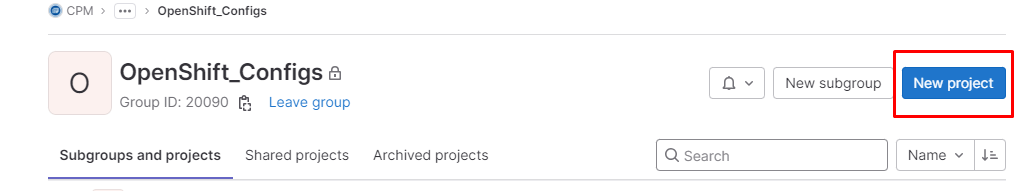
* Add your **Message** and Click on **ALL** and **Commit and Push**

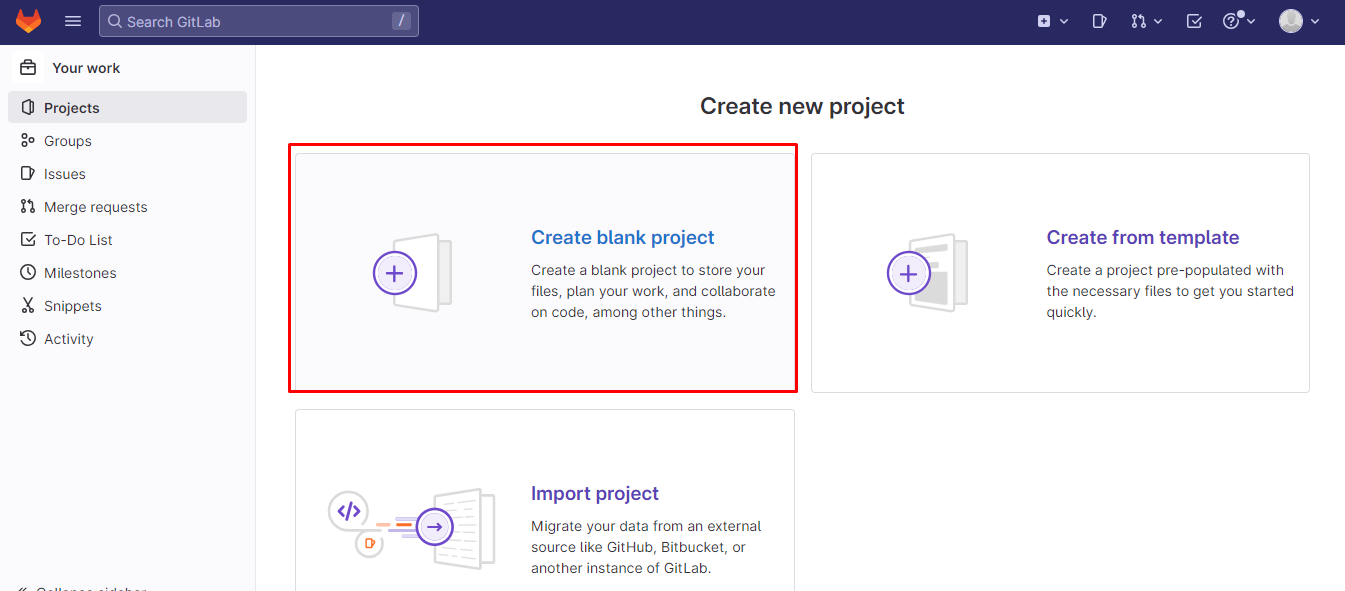


**Now you can see in Gitlab your project is Pushed into Gitlab..**

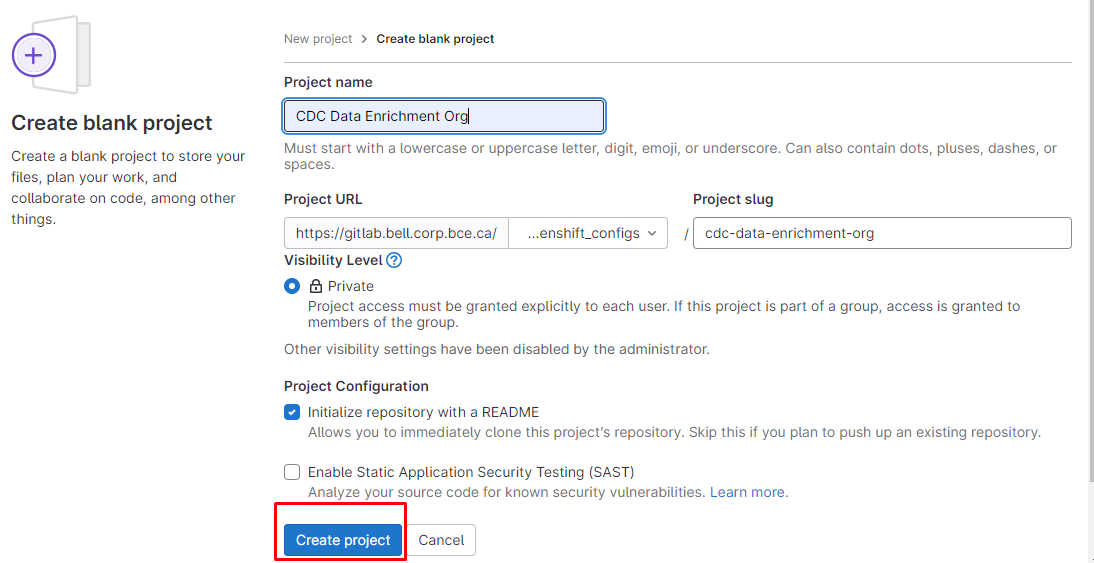
**GIT Repo –Openshift details:**

* Inside newly created subgroup, create another new subgroup for adding the openshift related details.
* Click on **OpenShift\_configs** and create a **New Project** with Same Name (which was created recently)



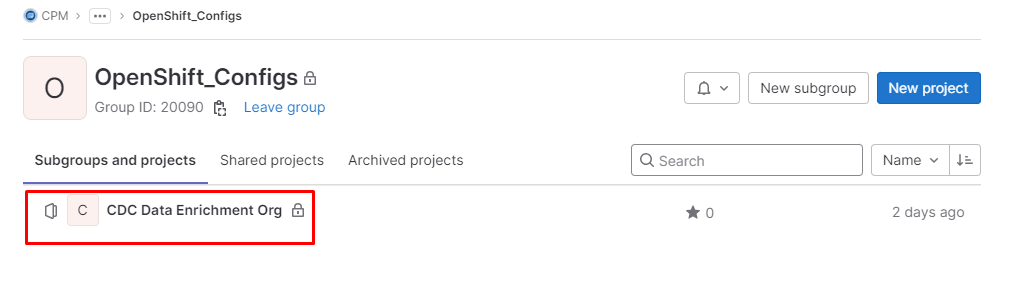


* Fill the required details as shown below.
* Click on create project.

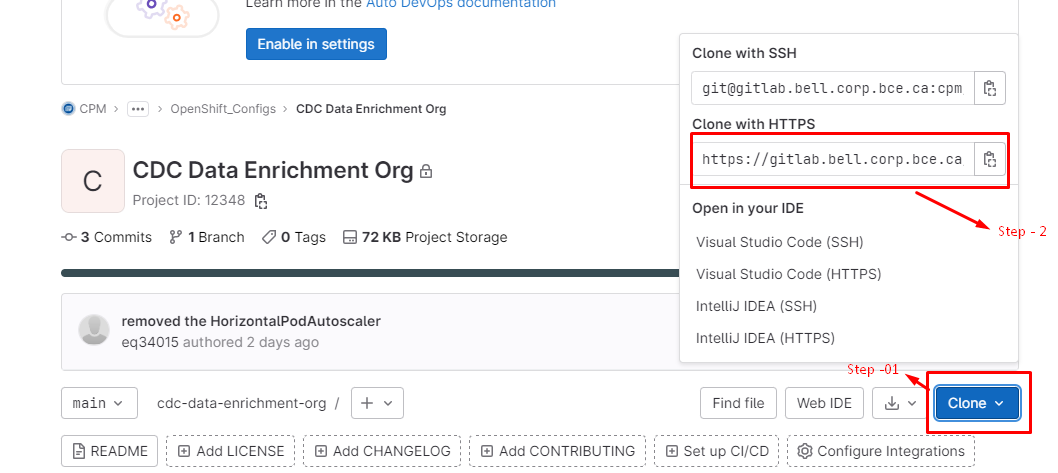


New project is created successfully.

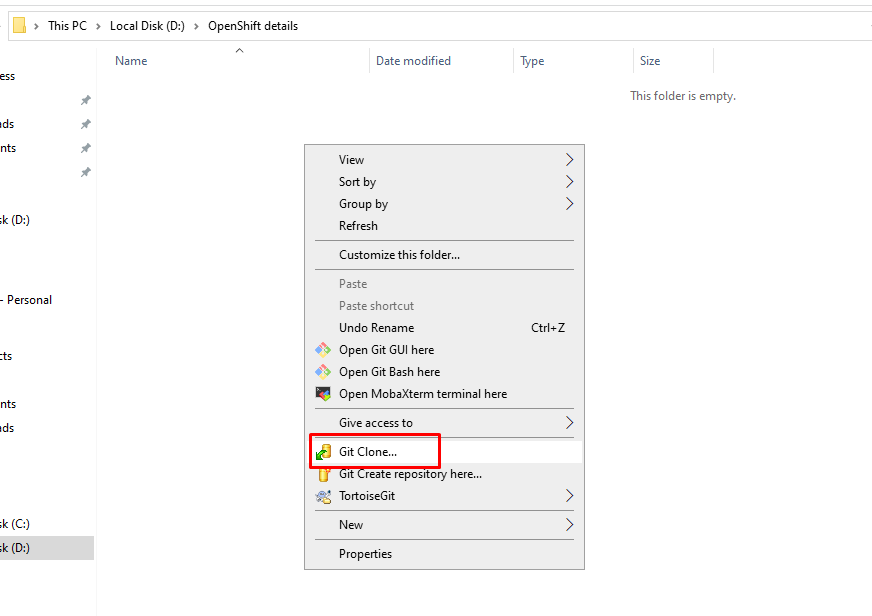
* Go to Openshift\_Configs repo



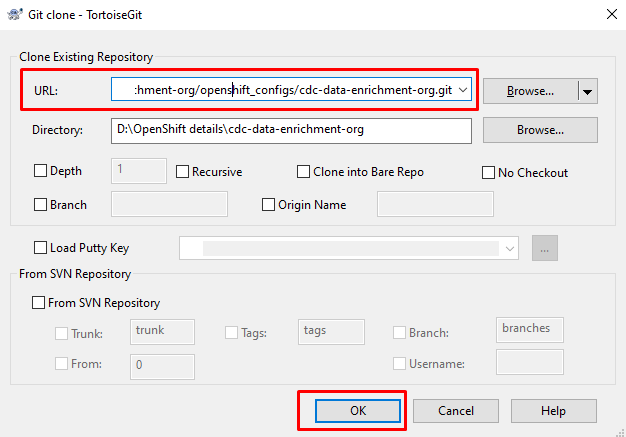
* Click on clone and copy the clone with HTTPs URL and send it to the respectively SPOC.



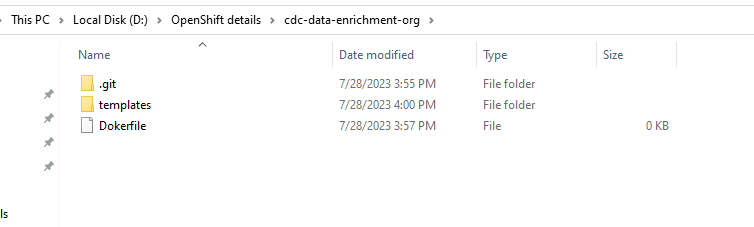
* Create an Empty directory with your Project name
* Right click --> Git Clone the project Repo using Tortoise as shown below



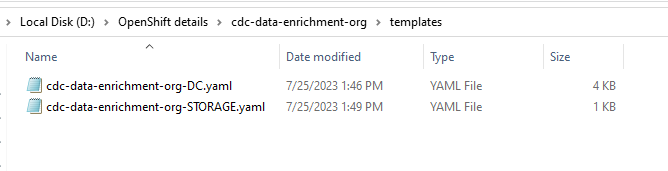
* Paste Your copied **HTTPS** url and Click **OK**



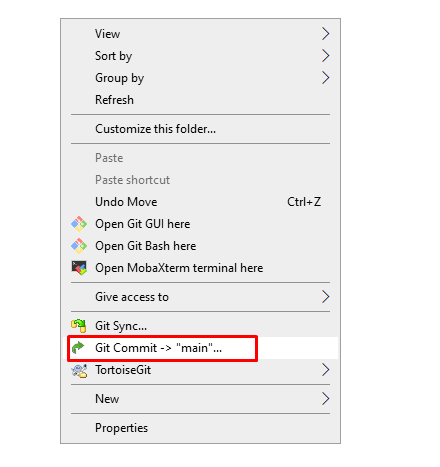
* Inside **Cloned Directory** create another directory with name **templates**
* Paste your OpenShift Config’s **(Deployment-config, Persistent Volume Claim, Service, Route templates) for your project.**



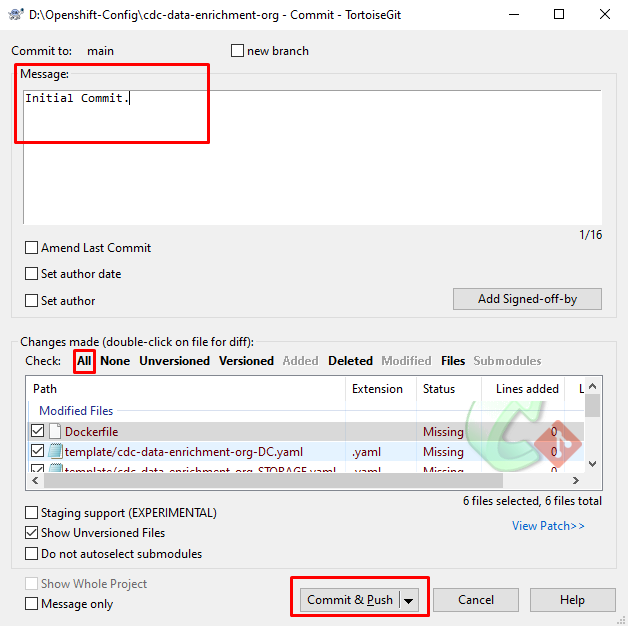
* Inside template folder



* Now come to one step back and commit this repo to **openshift-configs repo**



* Give the **Message** and Select **ALL**
* Click in **Commit & Push**



* Now you should be able see the committed repo in Gitlab.

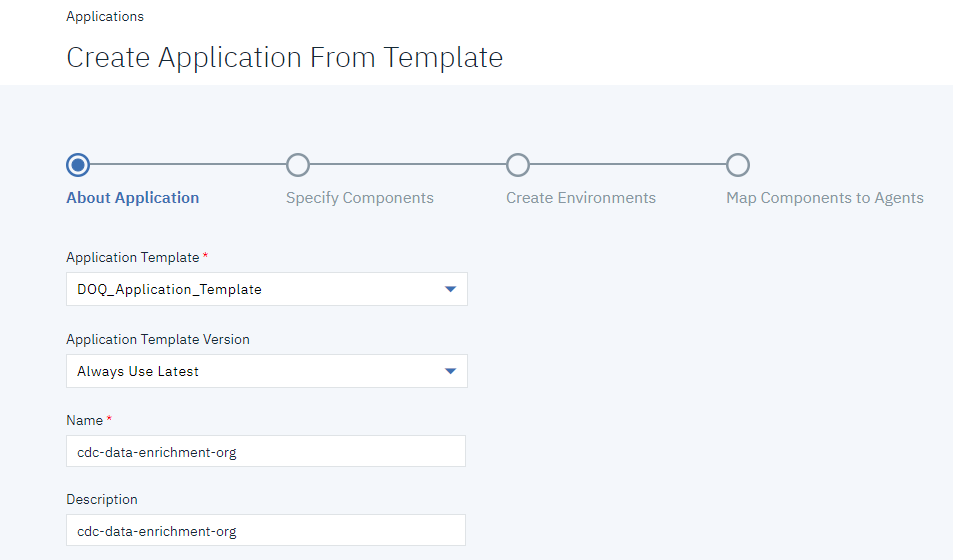
**Udeploy – Application:**

Udeploy URL: <https://udeploy.bell.corp.bce.ca:8443/#applications>

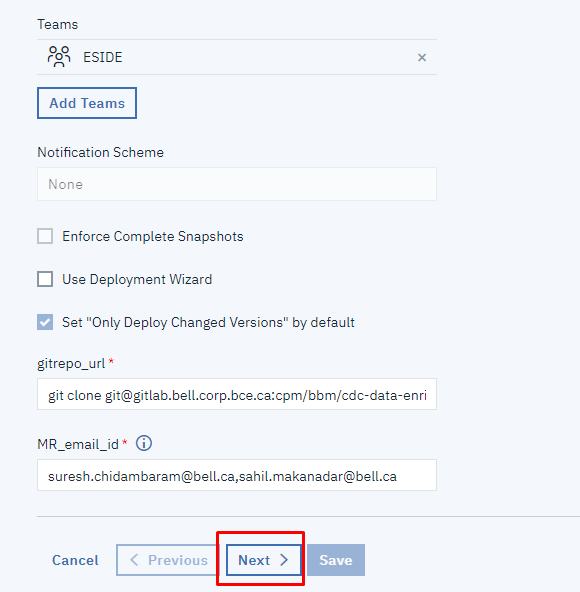
* Go to application tab.
* Select Create application →From Application Template



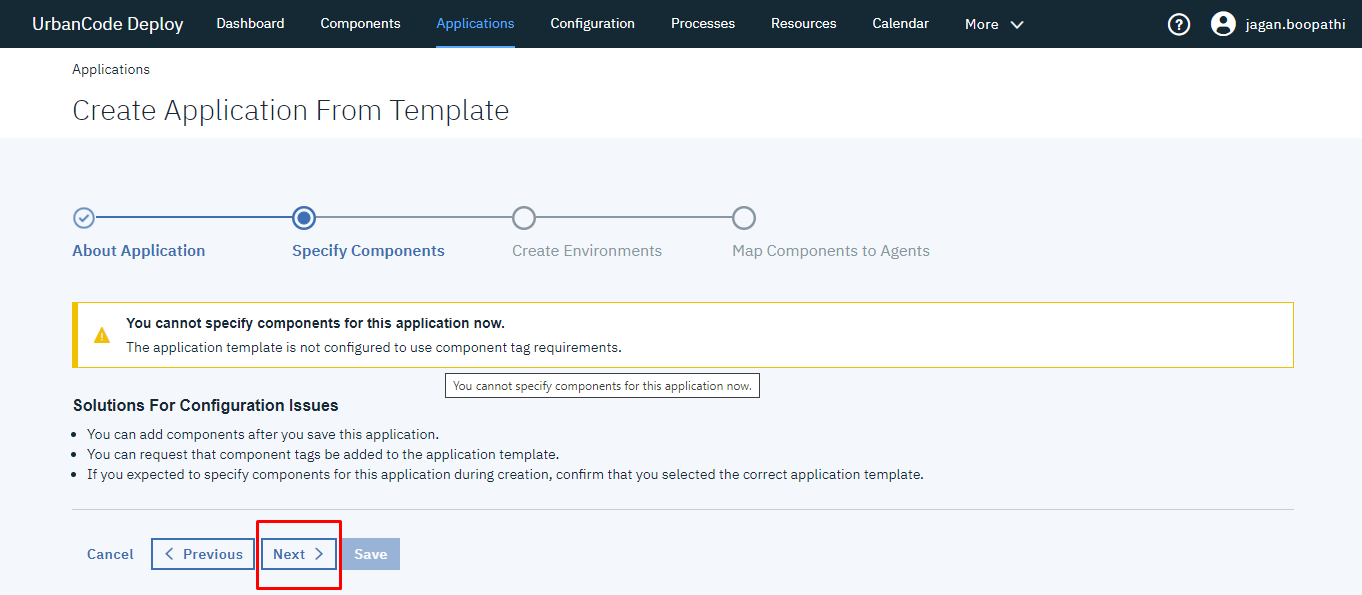
* Fill the details as shown below with respect to Project and click on **NEXT**



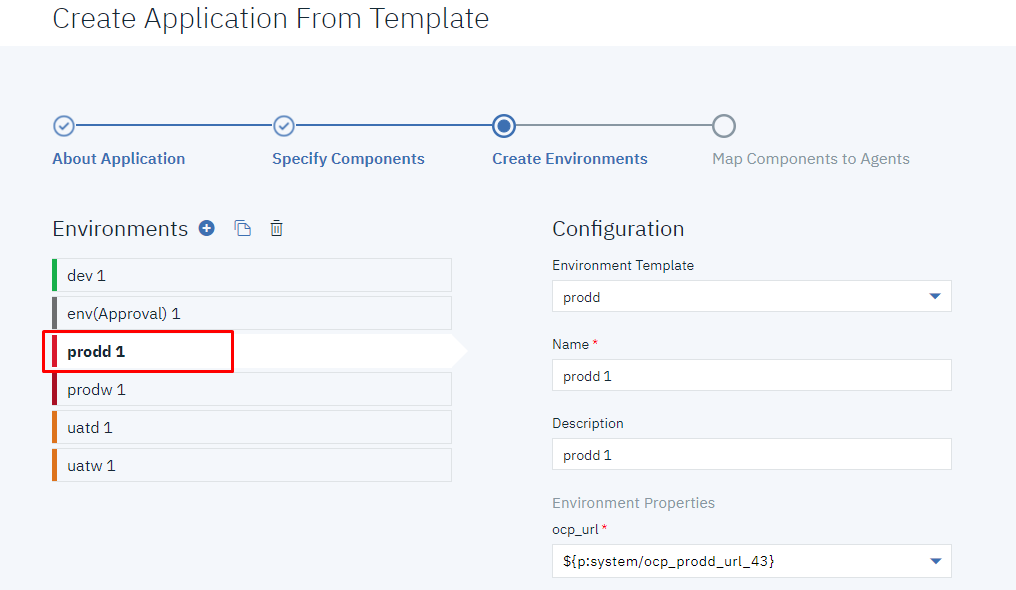
* In **gitrepo\_url** add git clone and then append clone with **SSH link**



* Click **NEXT**, once you get the below page.

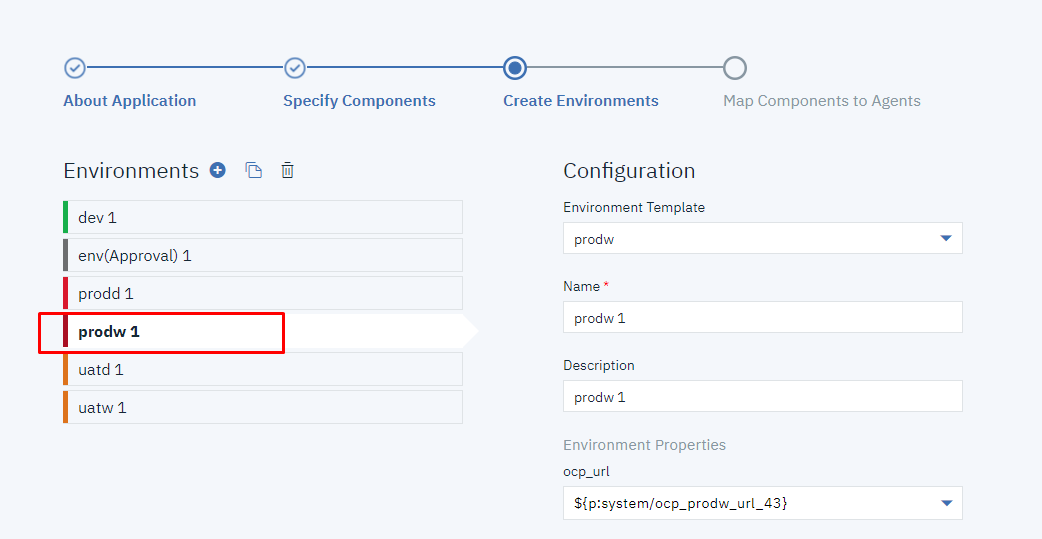


* Create Environment Setup for need ENV's
* Fill the required details as shown below and after configuring for all the environments then click **NEXT.**
* **PRODD1**



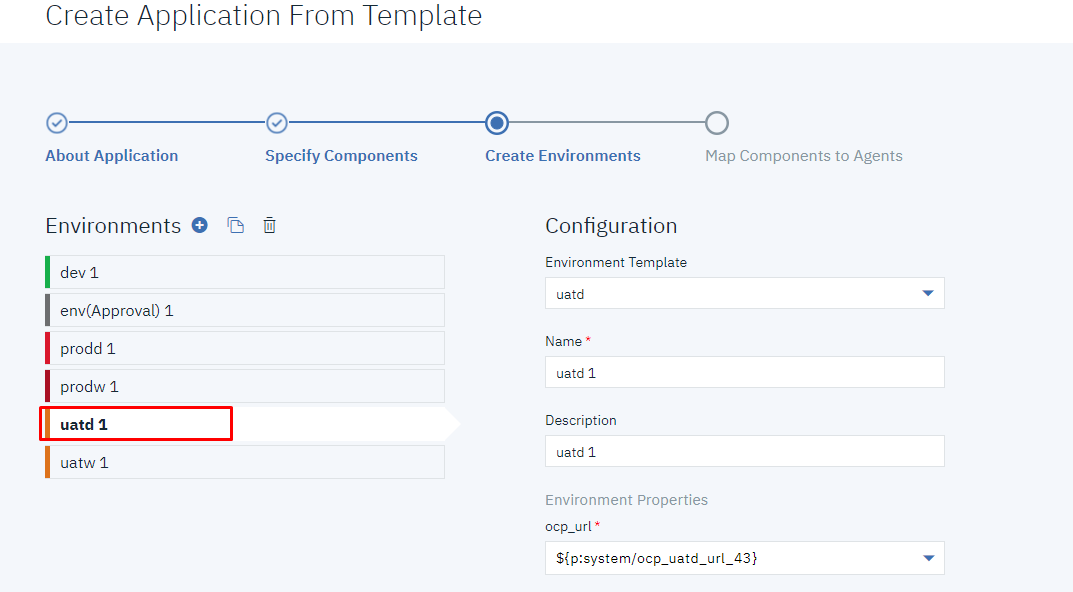


* **PRODW1**



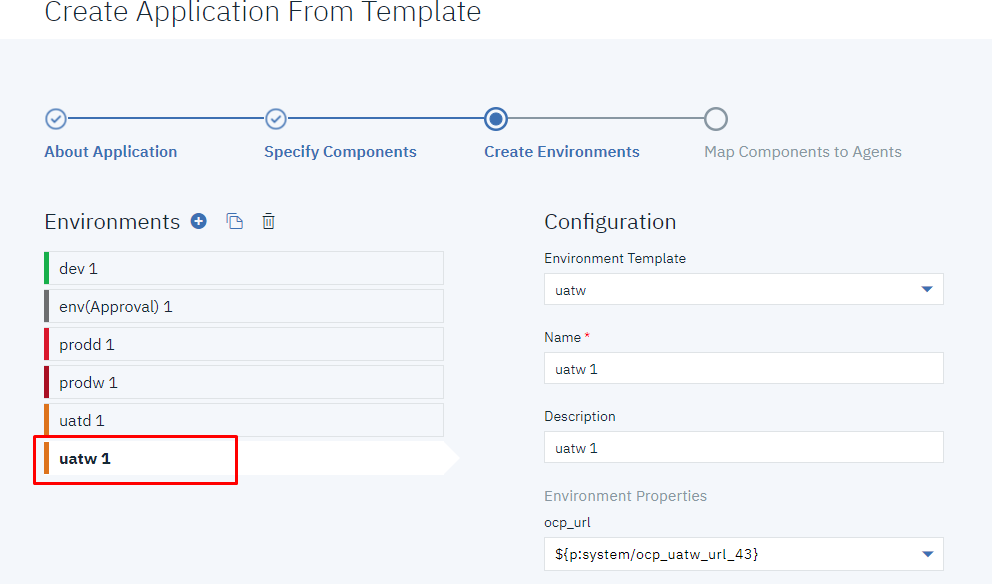


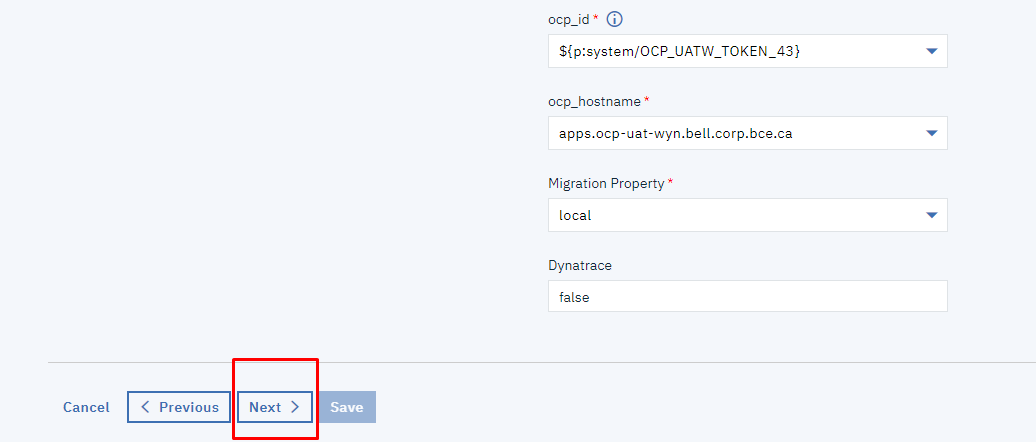
* **UATD1**



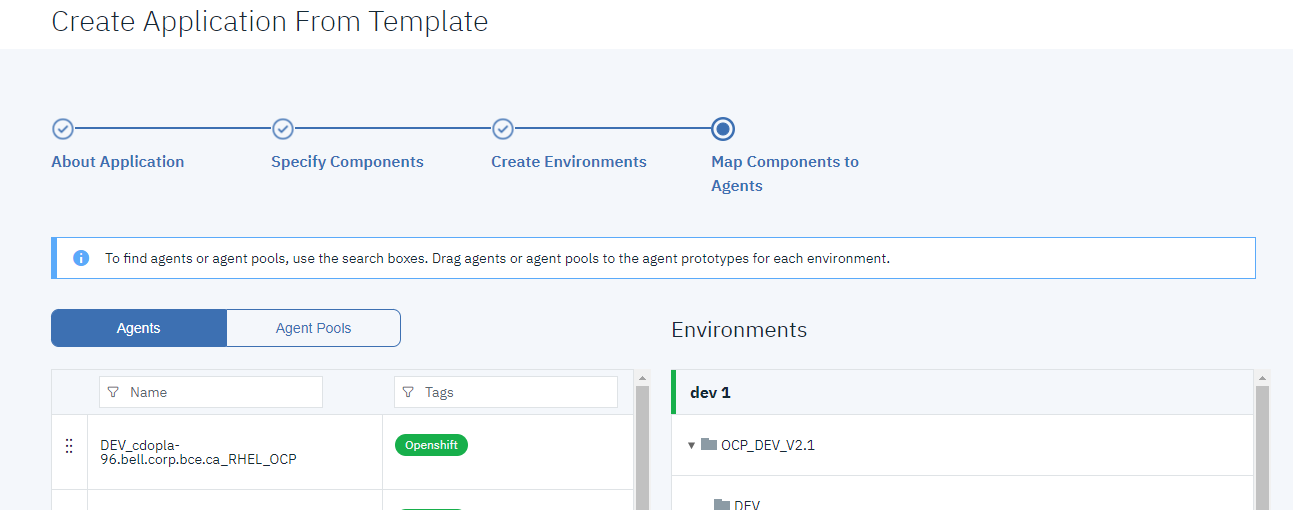


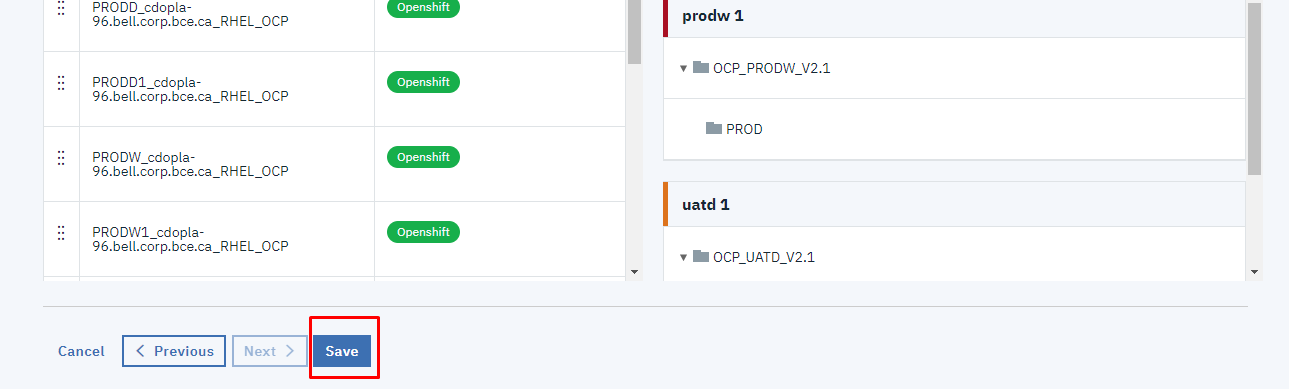
* **UATW1**





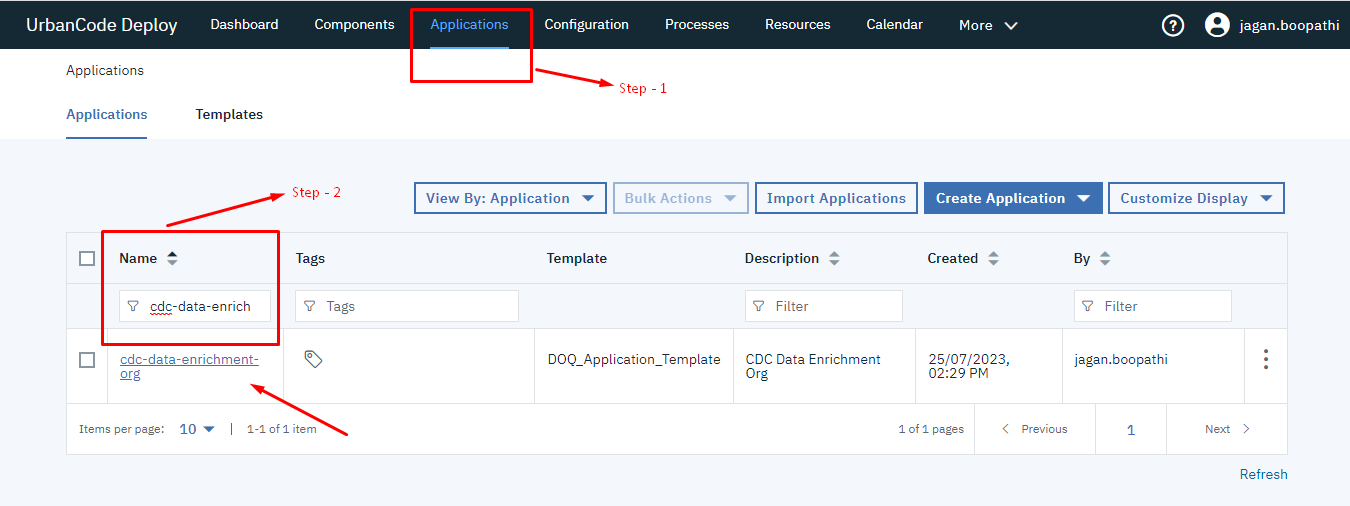
* It will redirect you to the below page then click on **SAVE**.





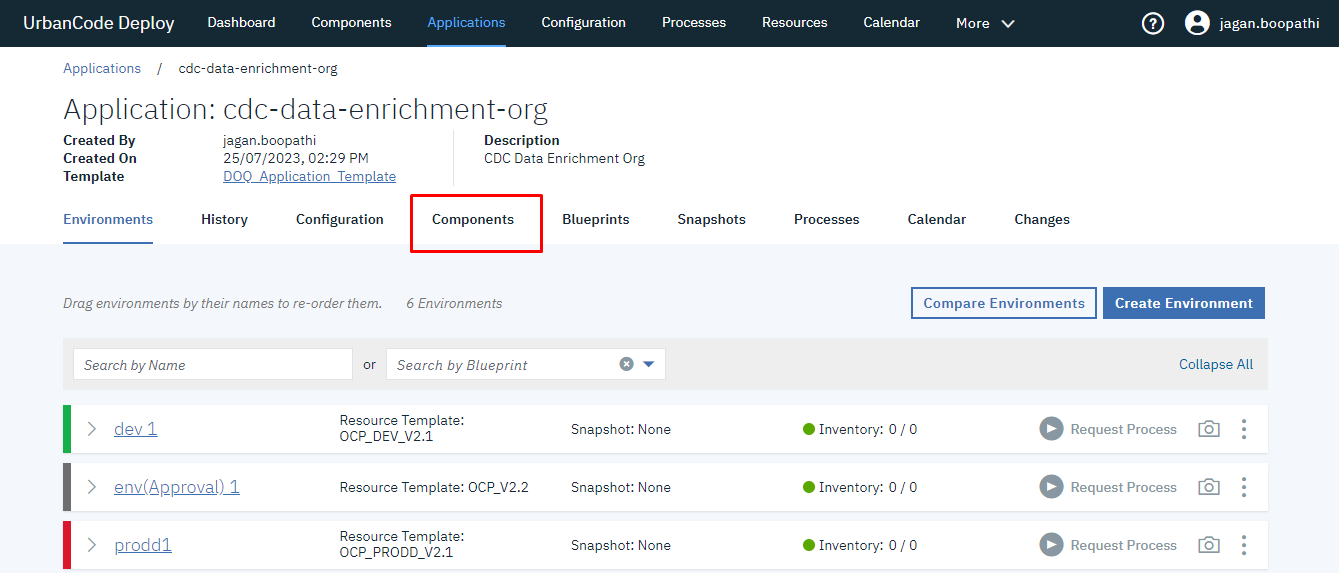
Now your application is created.

* Go to **Application** Tab and Search for your application name in **Name** field.

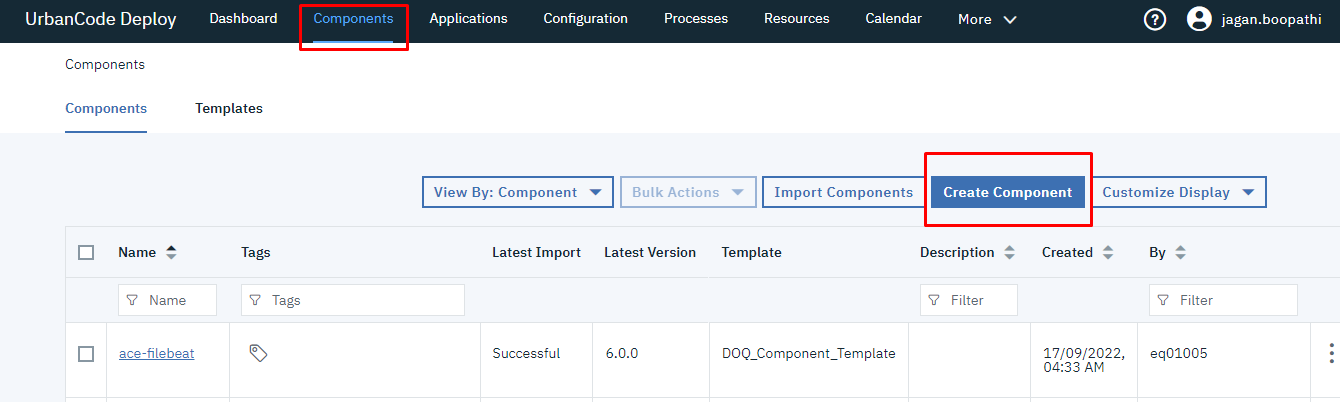


**Create Component**

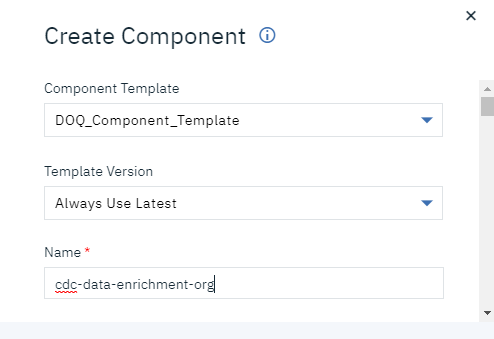
* Click on your application and then Go to **Components**.



* Click on **Create Component**
* Don’t fill **Git URL** and **Git folder**
* Give Artifactory Project folder name as **“cpm”** and Artifactory project Repo name **“cpm-dockerolocal”**

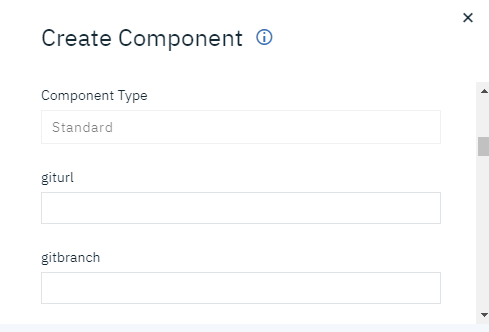


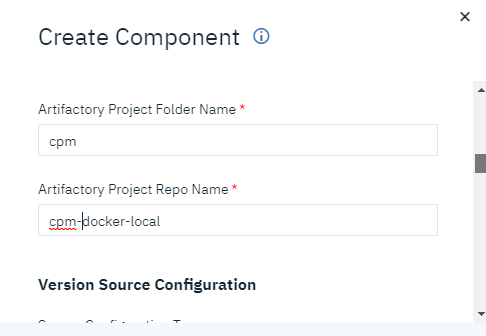
* Fill the details as shown below

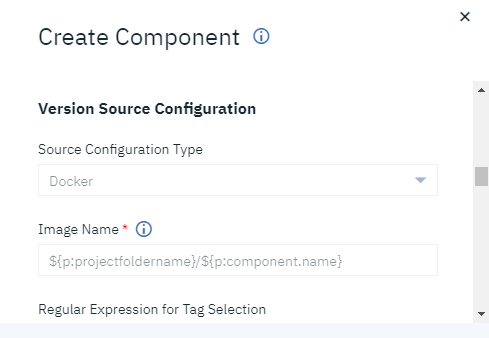




* Don’t fill anything into Git url and Git branch

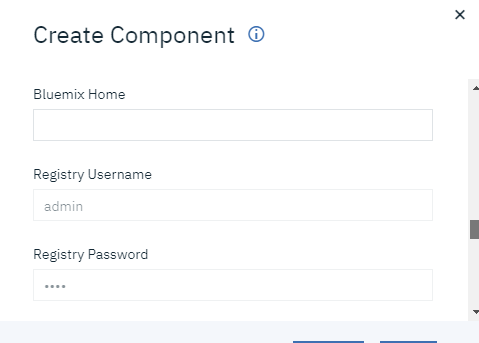


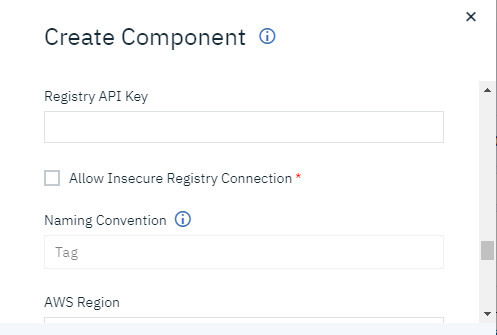


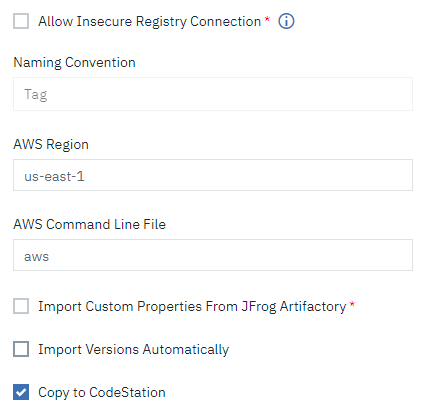


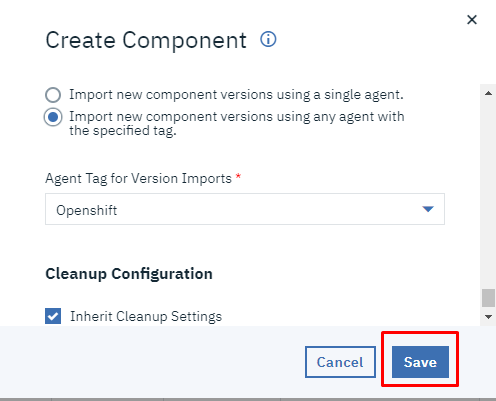








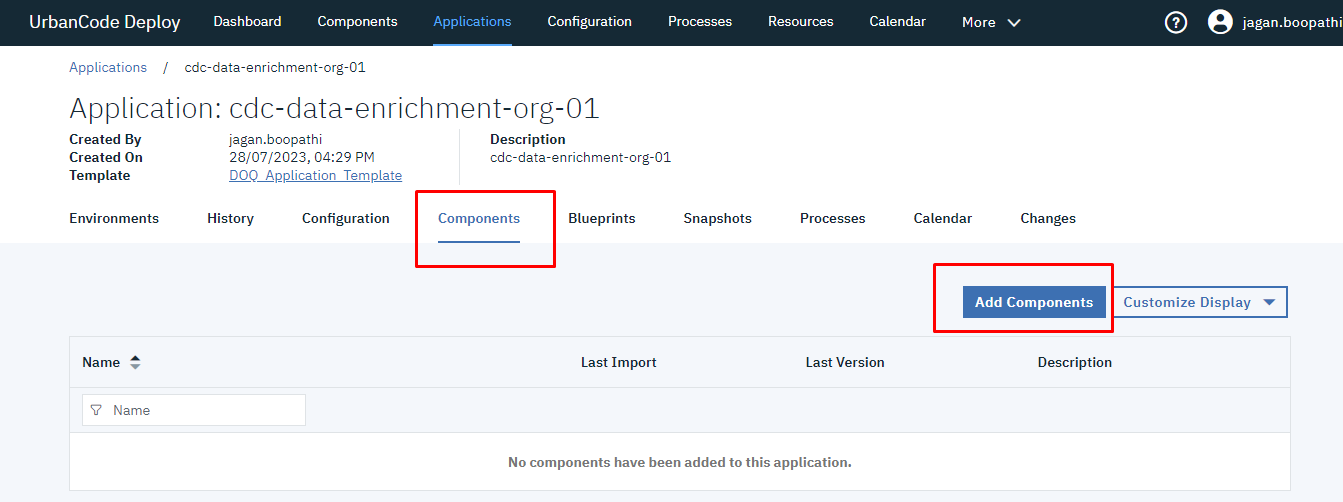




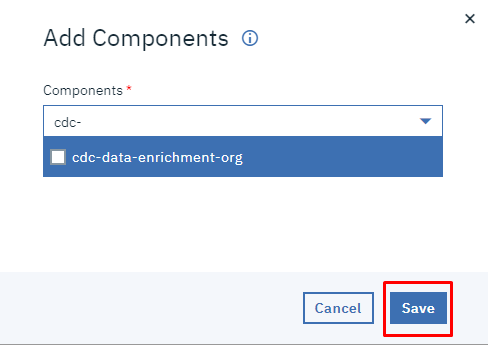
* Component is created successfully.

**To Link the component and application:**

* Go to **Applications** Tab in main menu
* Open the newly created application
* Click on the **component** tab and **Add Component**



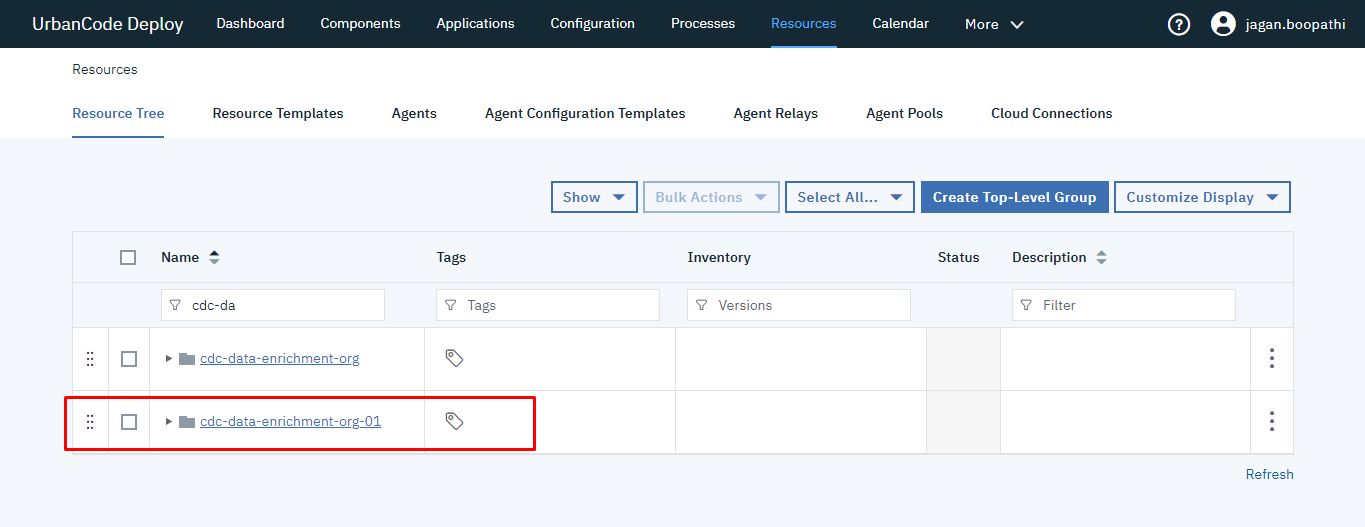
* Select the newly created component from the dropdown and click on save.



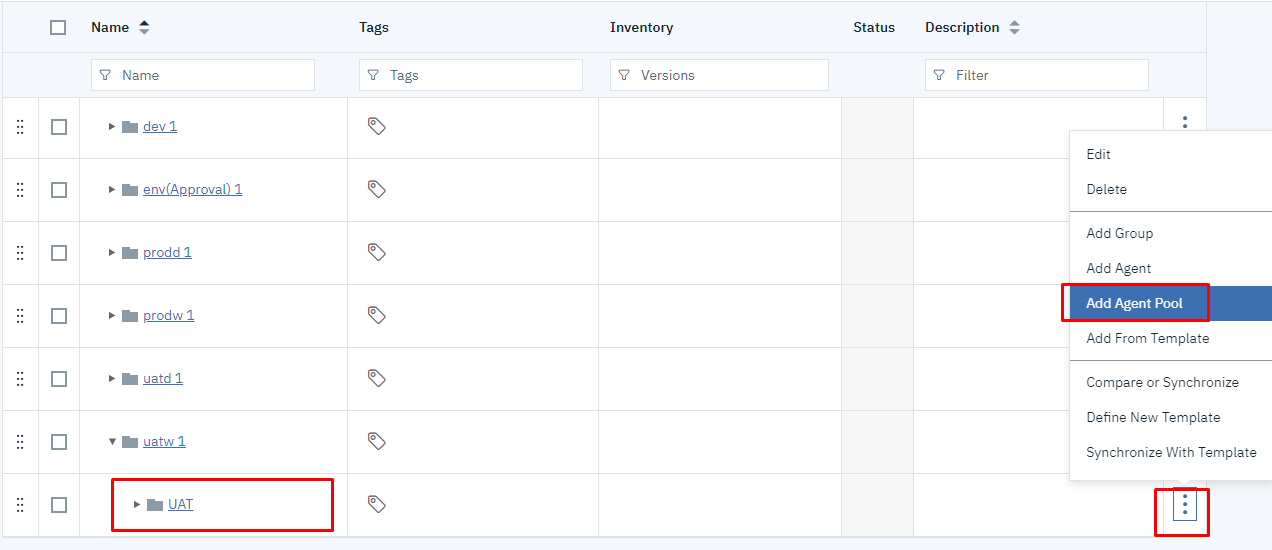
Component and application are linked successfully.

**To add resource:**

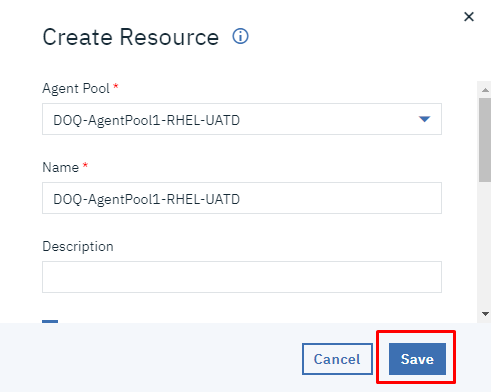
* Click on the **Resource tab**
* Go to newly created application and expand it.



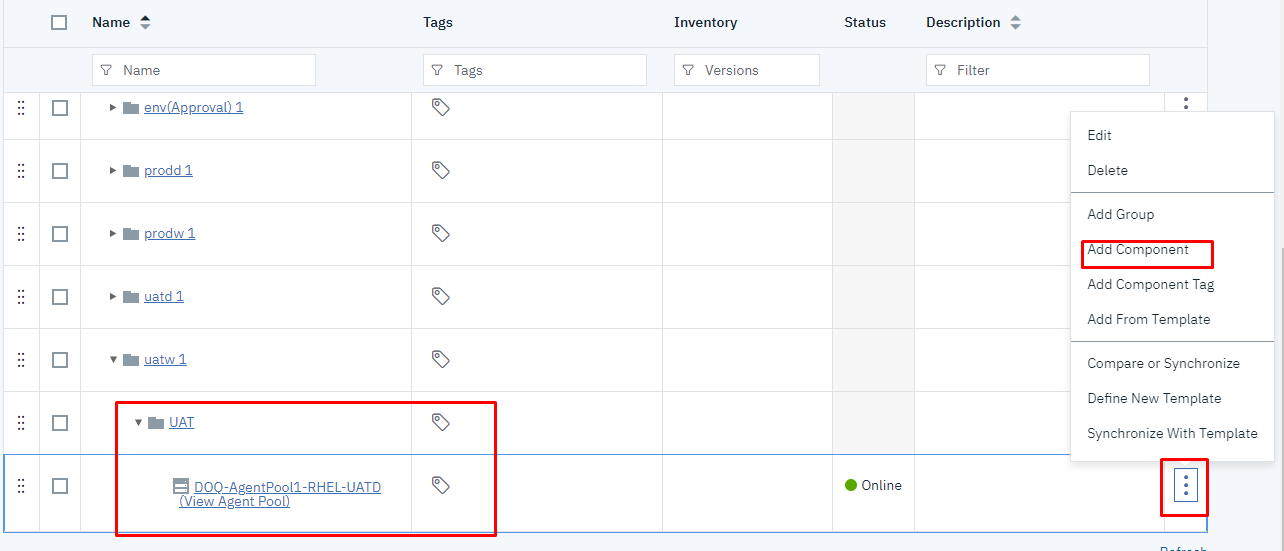
* Expand each subfolder and click on Three dots and do the following setup mentioned in the screenshots
* Select the **Agent pool** with respect to **Env**



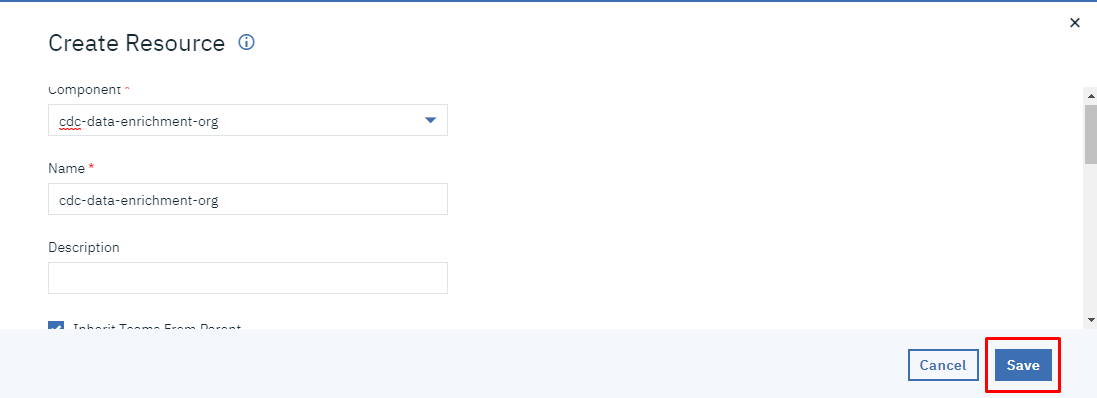
* Click **Save**

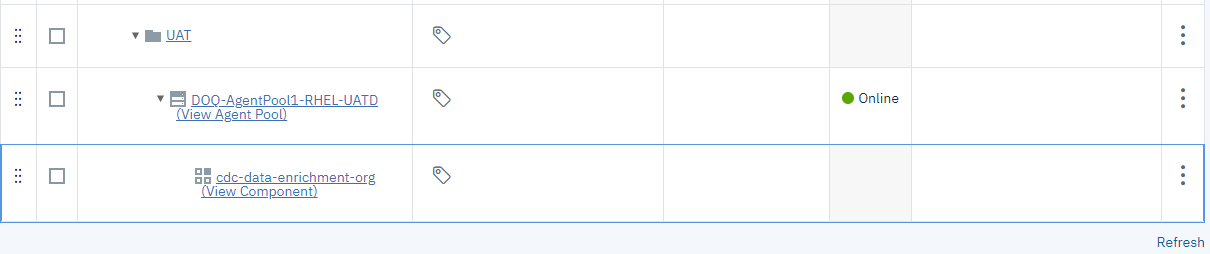


* Expand Agent Pool and add **Agent Component**



* Select **Add Component** andEnter the name of **Resource**
* Clickon **Save**





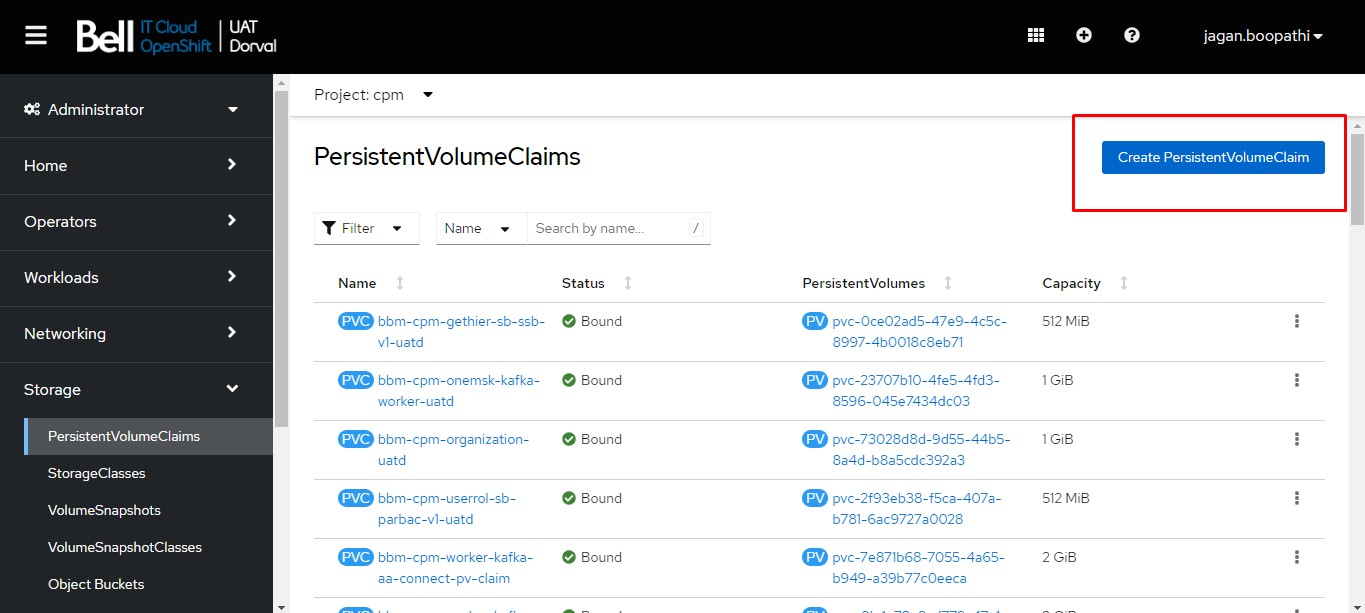
**Agent pool and Add Component** adding for One server is Successfully done.

* **Likewise, Agent pool and Components must add to the other servers under the application.**
* **Once you have added Agent pool and Components for all the servers, the UDeploy setup is done.**

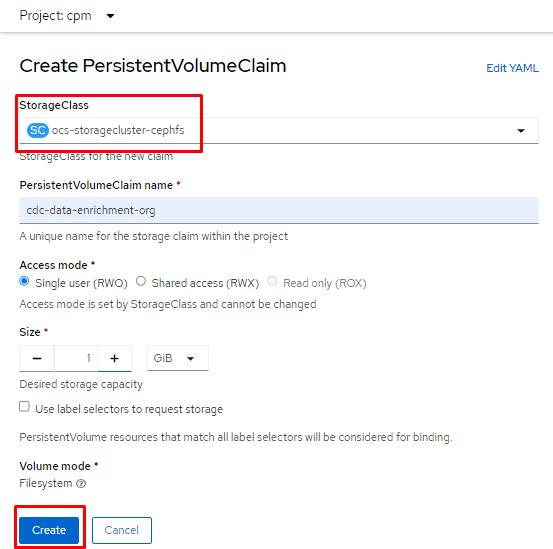
**Openshift Storage:**

**Url:** <https://console-openshift-console.apps.ocp-uat-dor.bell.corp.bce.ca/k8s/ns/cpm/core~v1~PersistentVolumeClaim>

* Click on Create PersistentVolumeClaim and fill the required details



* Fill the details as shown below and click **Create**



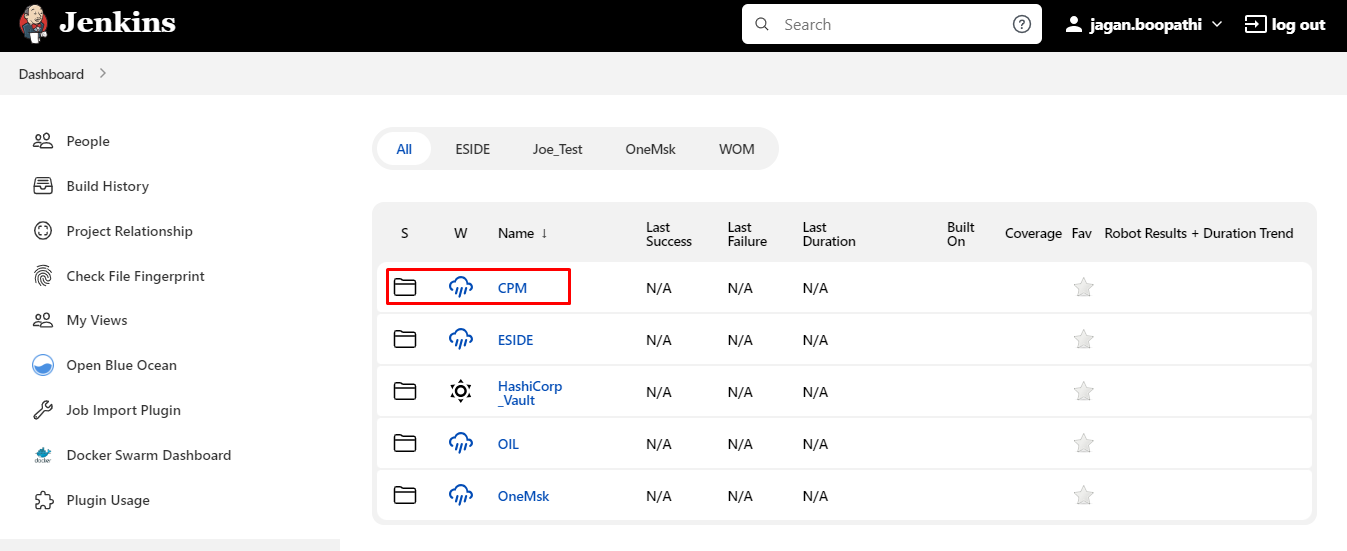
Storage Volume is created successfully.

**Jenkins - Job**

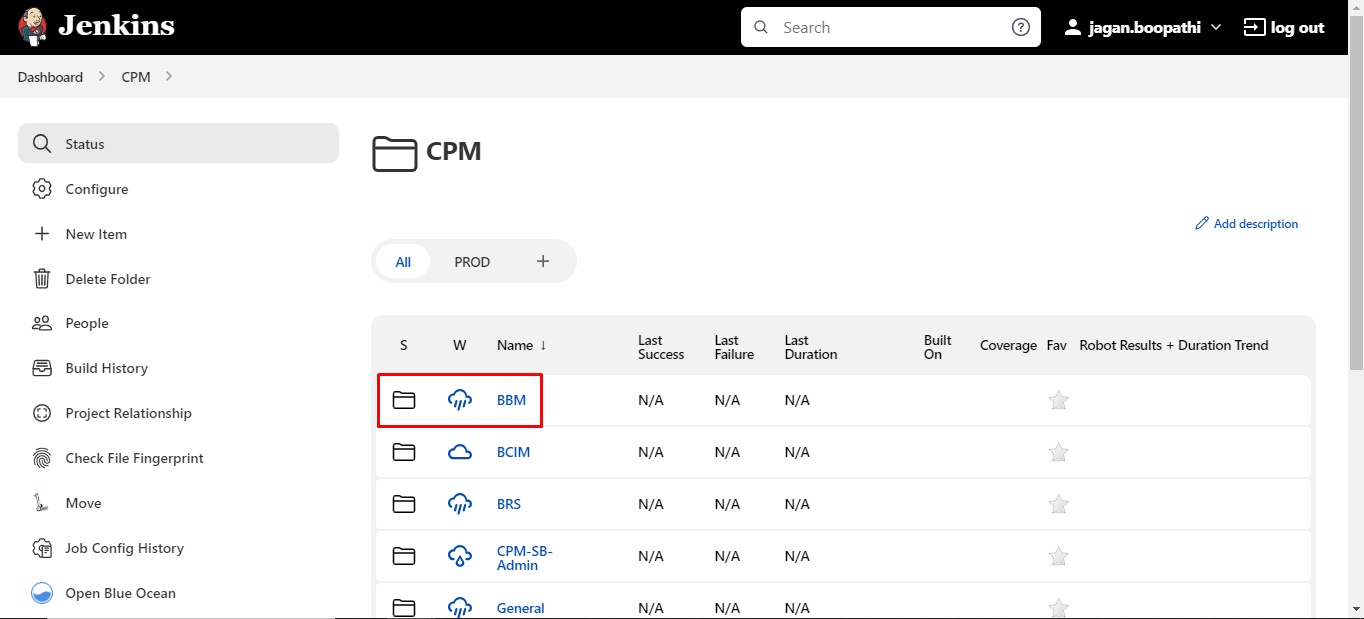
Login to Jenkins.

URL: <http://builds.bell.corp.bce.ca:8080>

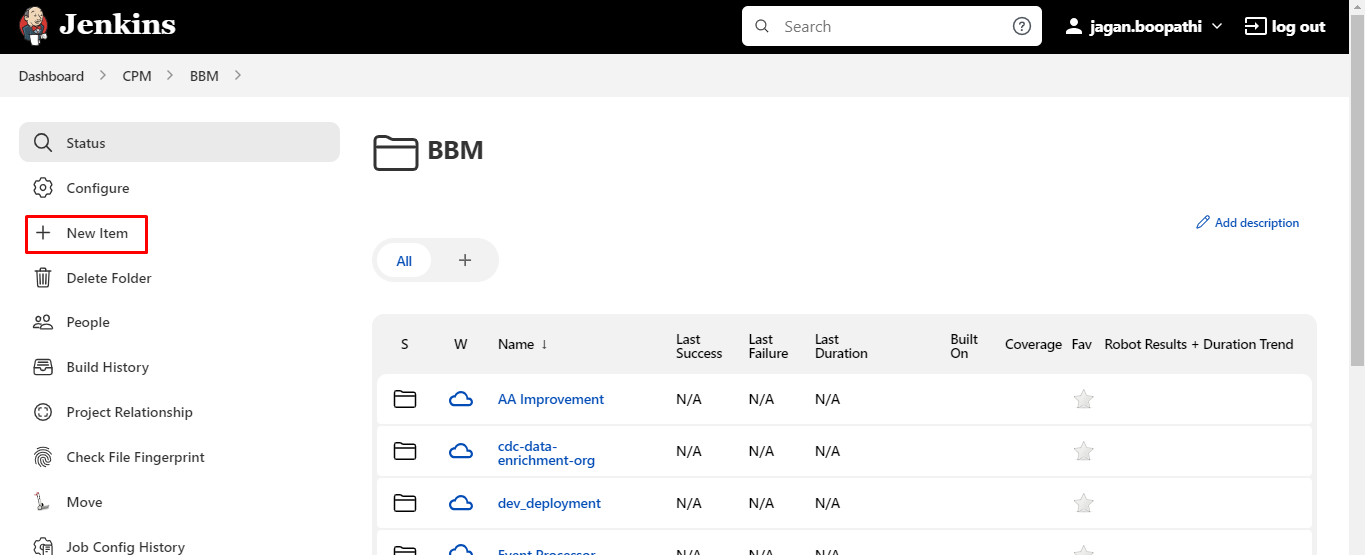
* Move to the required folder.
* Click on **New Item.**



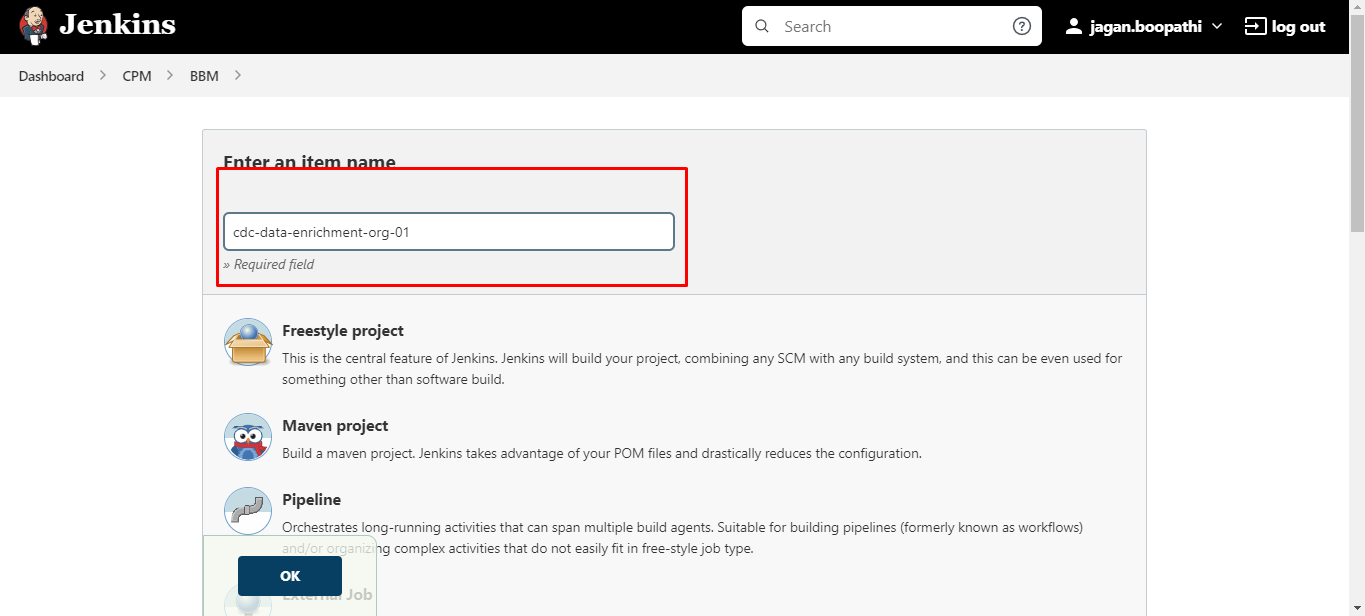
* Since this project is on **BBM**. We are moving to **BBM** folder.

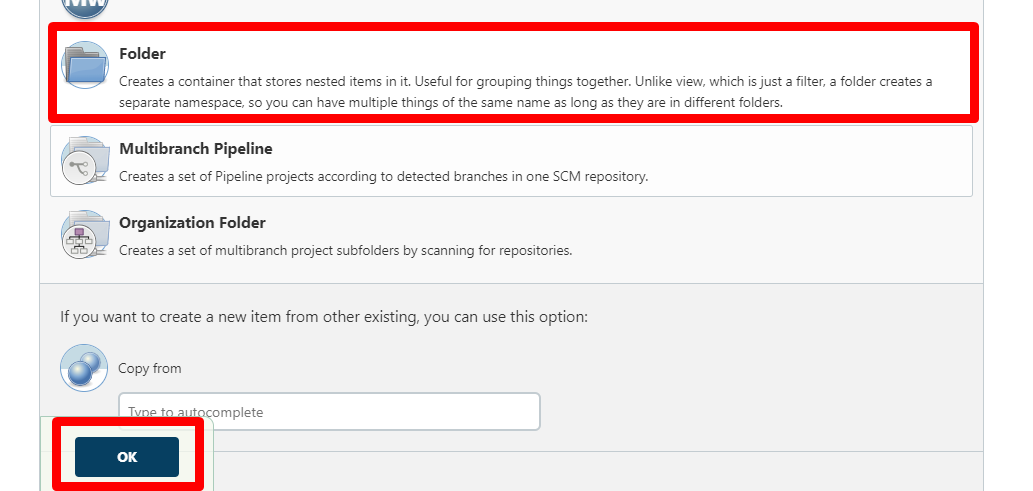


* Under **BBM** Click on **New Item** to create new Project

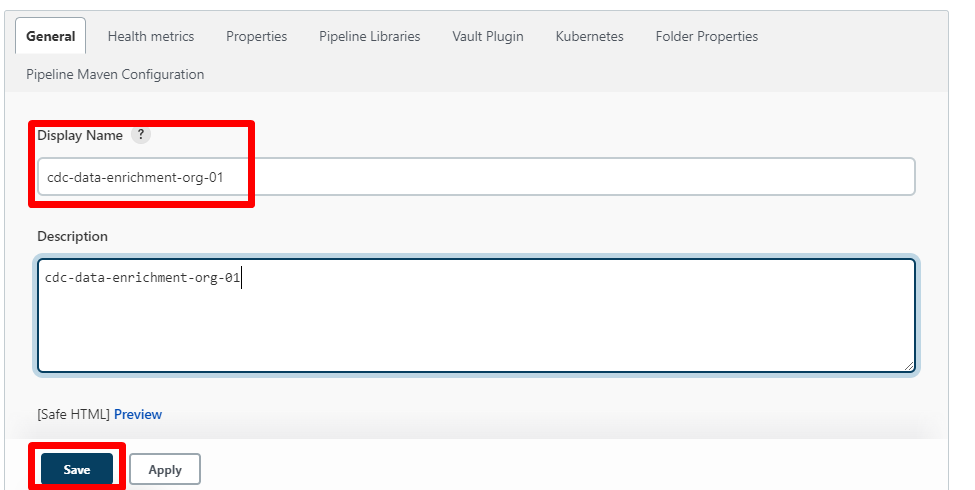


* Enter **Item name**
* Scroll down and select type as **Folder**
* Click on **OK**

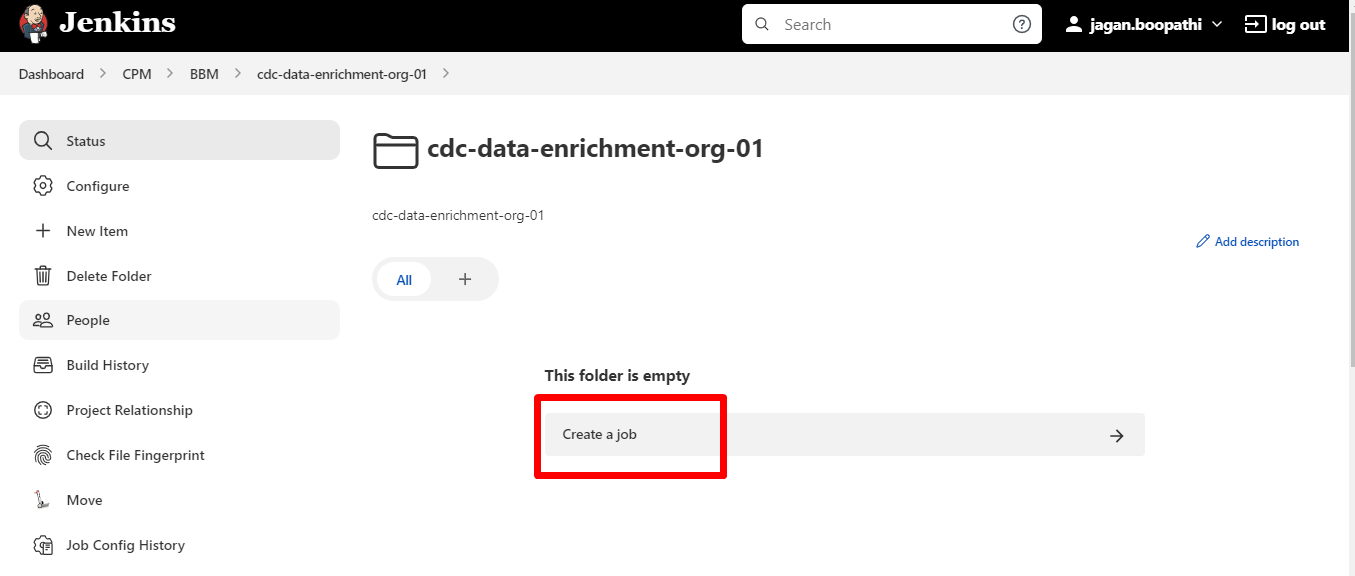




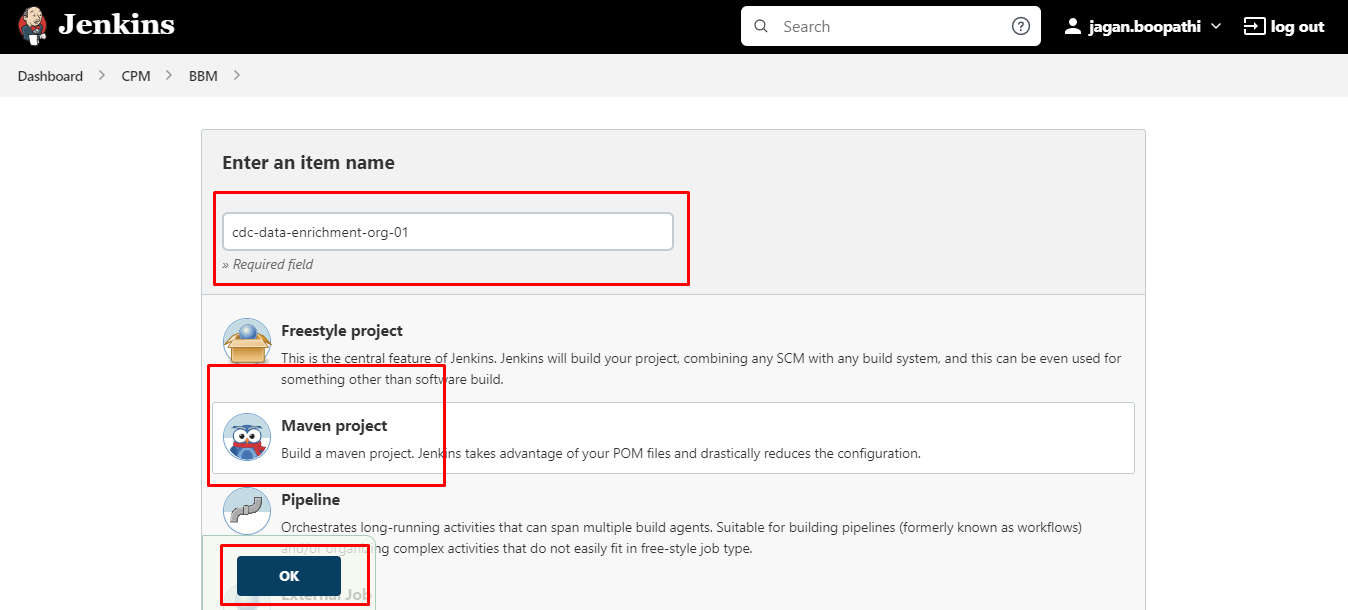
* You will find the below screen, enter Display name
* Click on **SAVE**



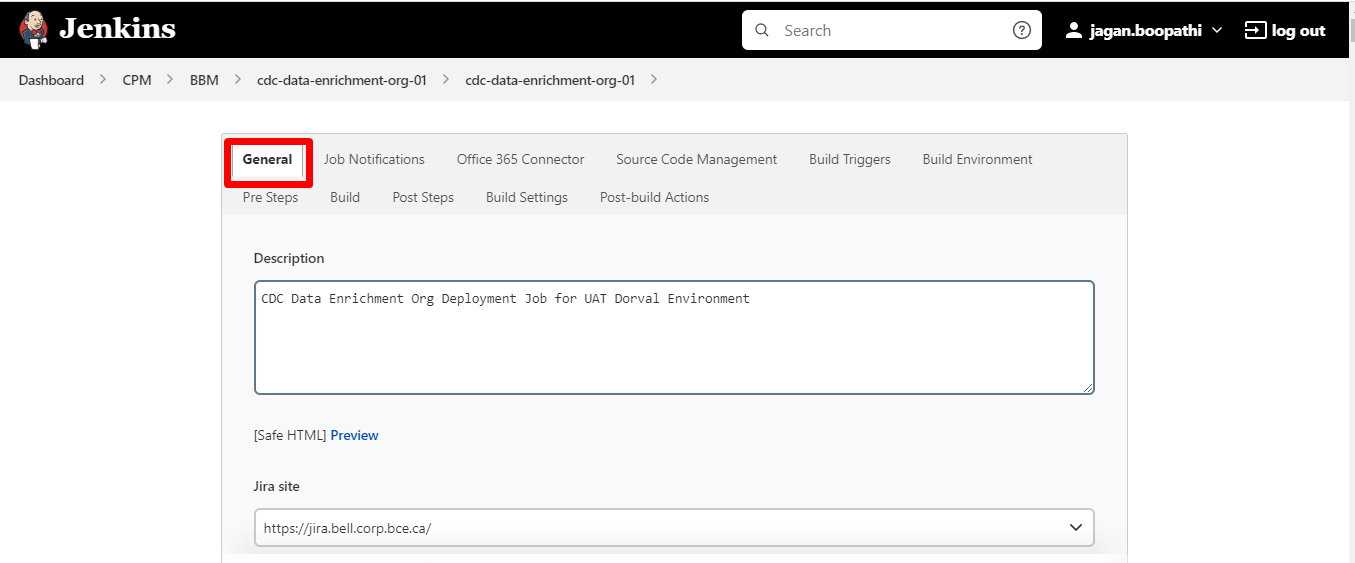
* Click on **Create job**



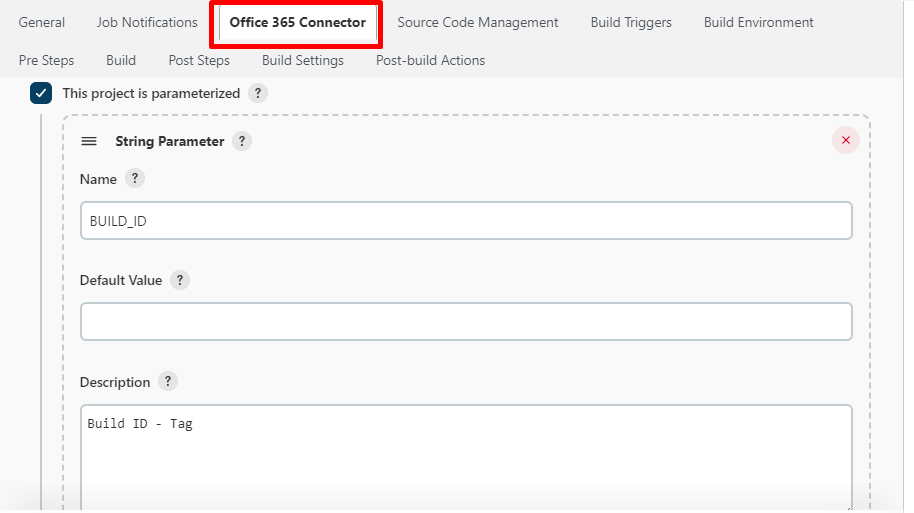
* Enter your **project name**
* Select **Maven Project**
* Click on **OK**.



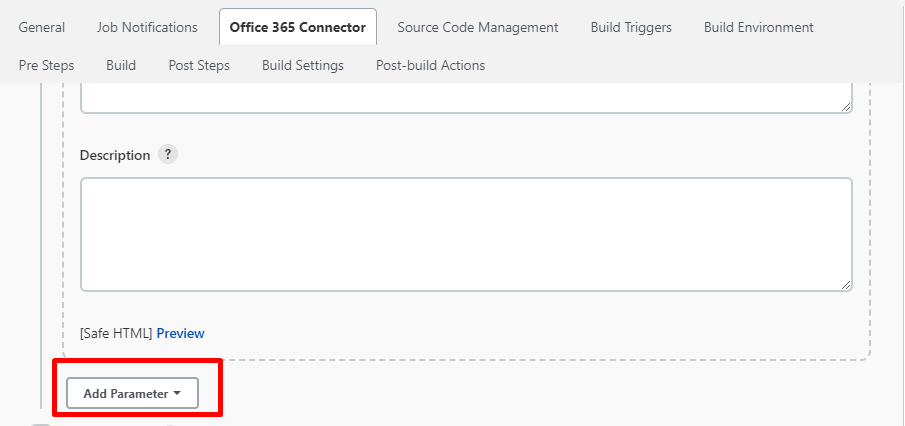
* Then you will find the below page and fill the required details as shown bellow
* Follow the below steps to complete the configuration part of job
* **Step 1**

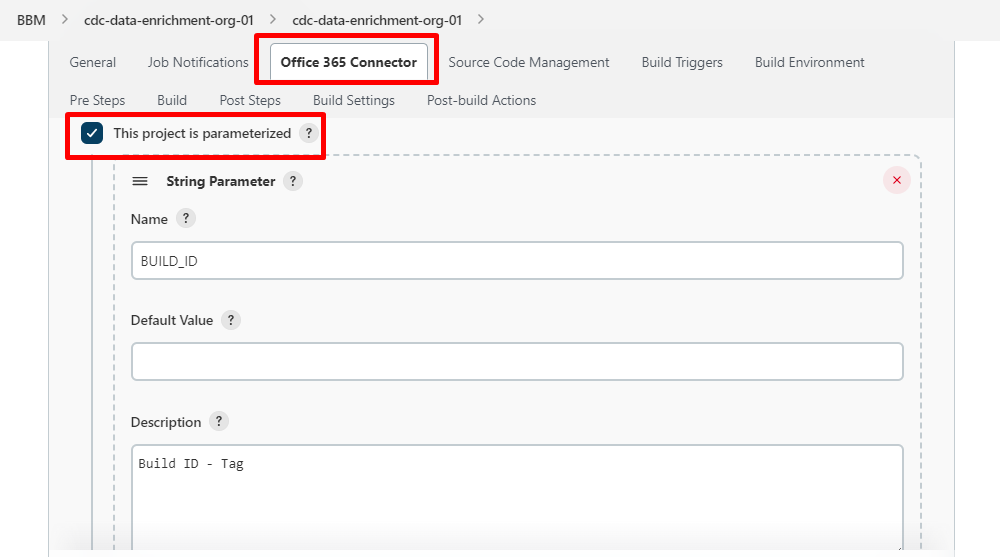


* **Step 2**



* **Step 3**
* Click on **Add Parameter** and choose **String Parameter**

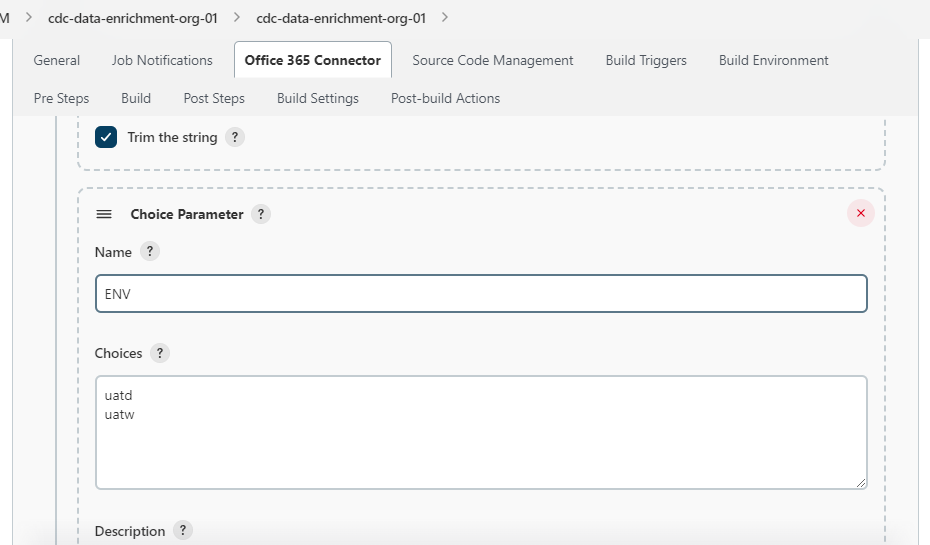




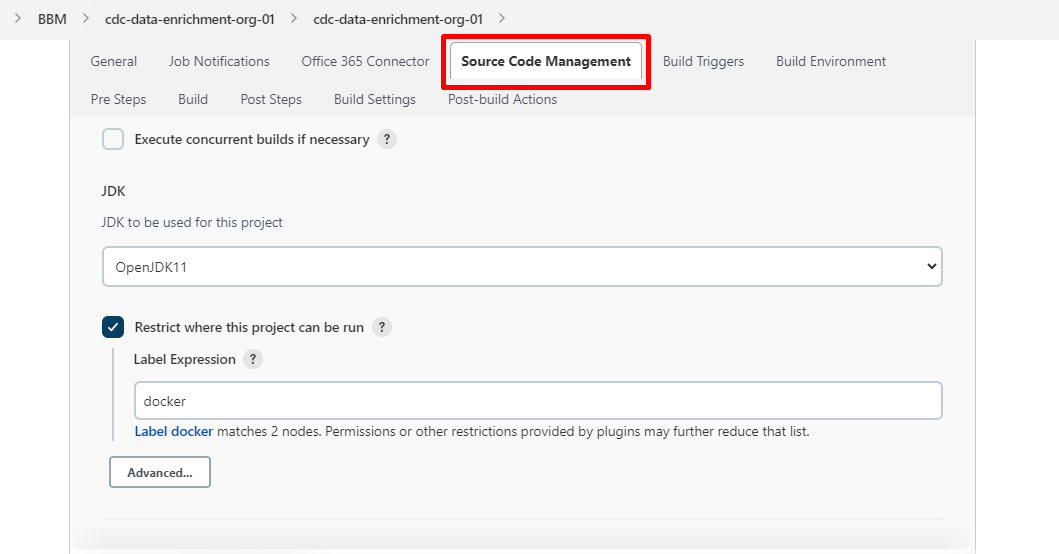
* Check the **Trim the String**
* Again, Click on **Add Parameter** and choose **String Parameter**



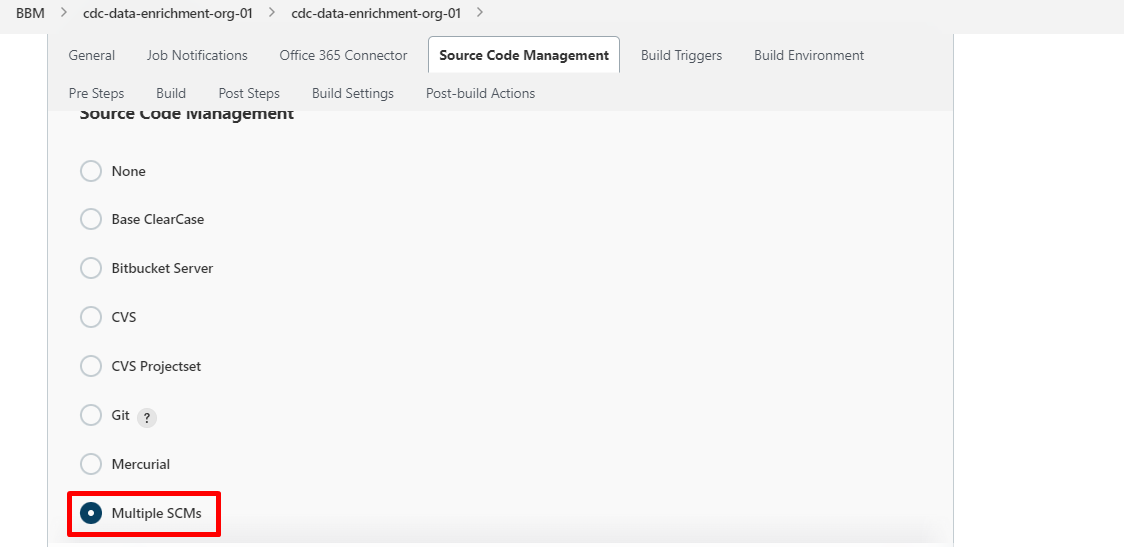
* Check the **Trim the String**
* Again, Click on **Add Parameter** and choose **Choice Parameter**



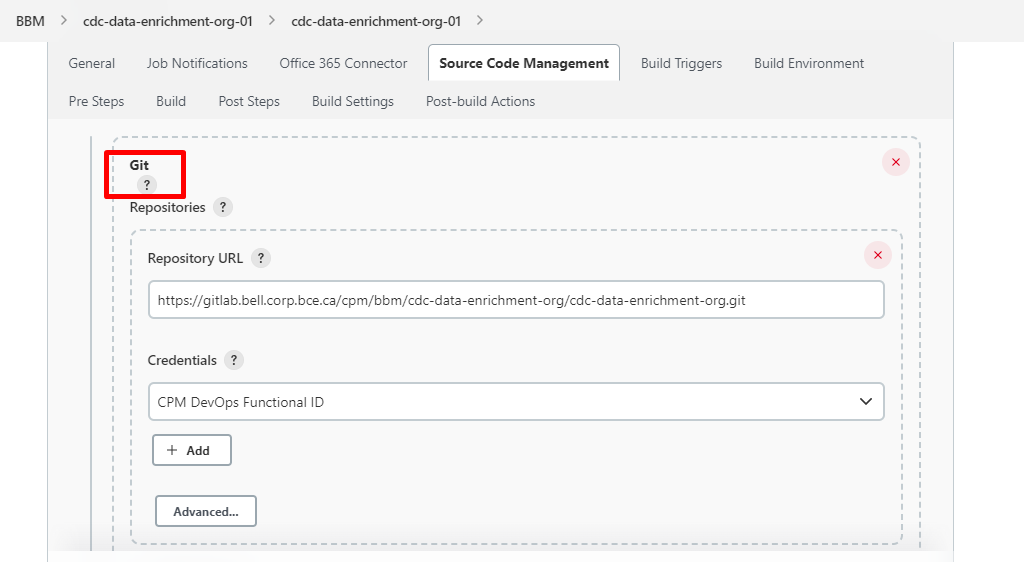
* **Step 4**

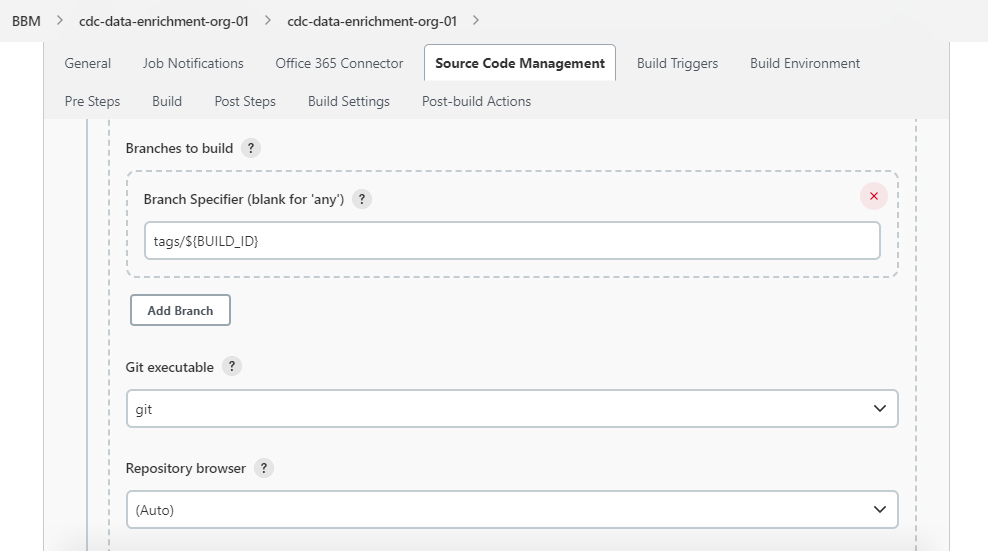


* **Step 5**

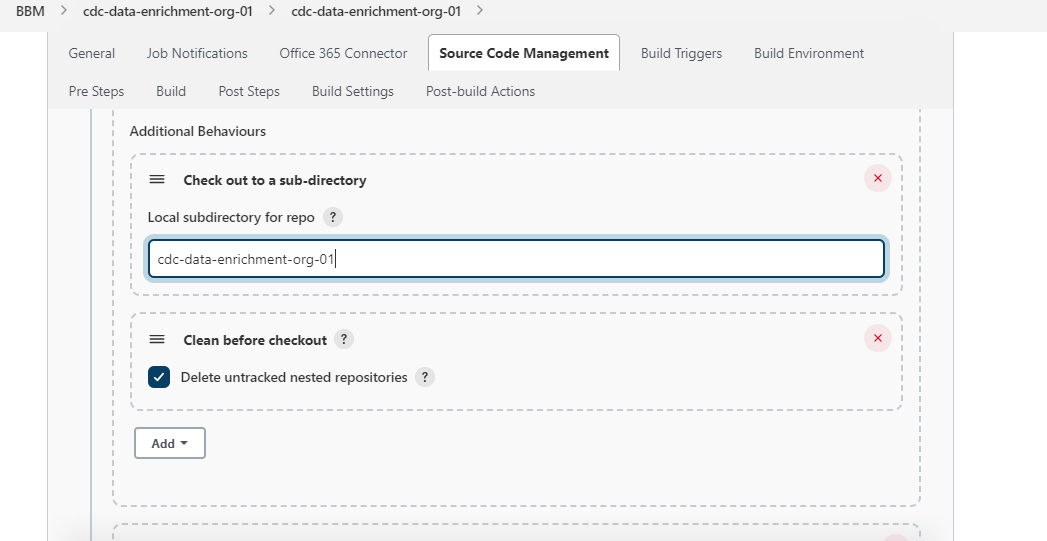


* Click on **Add Repository** and select Git
* Fill theGit Repo **HTTPS URL**
* Select Credentials as **CPM** **DevOps Functional ID**

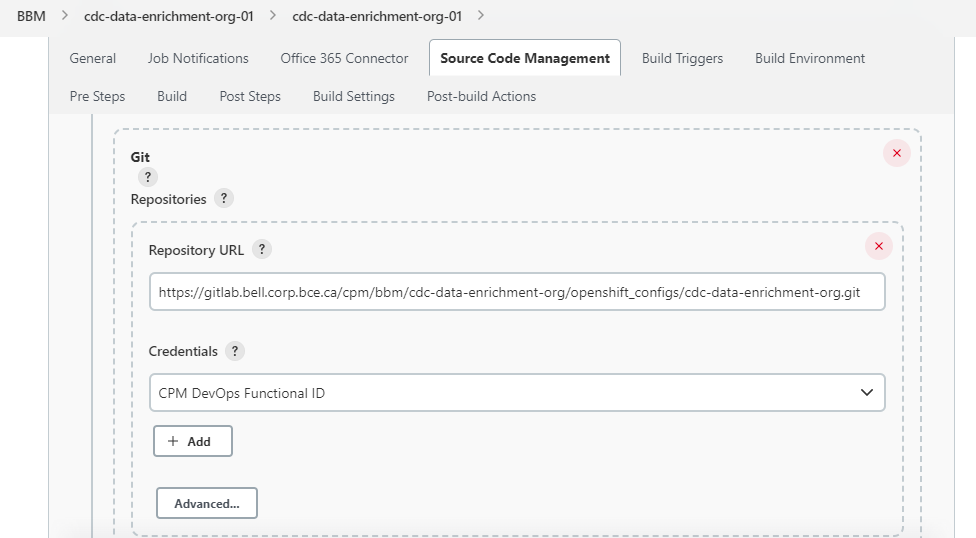


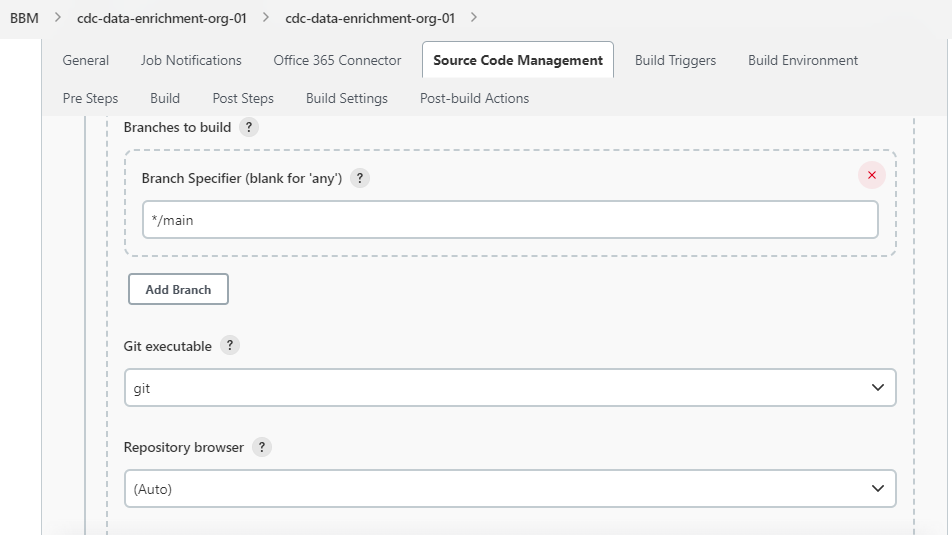


* Click on **Add** and Select **Check out to sub-directory** and enter your directory name
* Again, click on **Add** and Select Clean before checkout
* **Check** on Delete untracked nested repositories

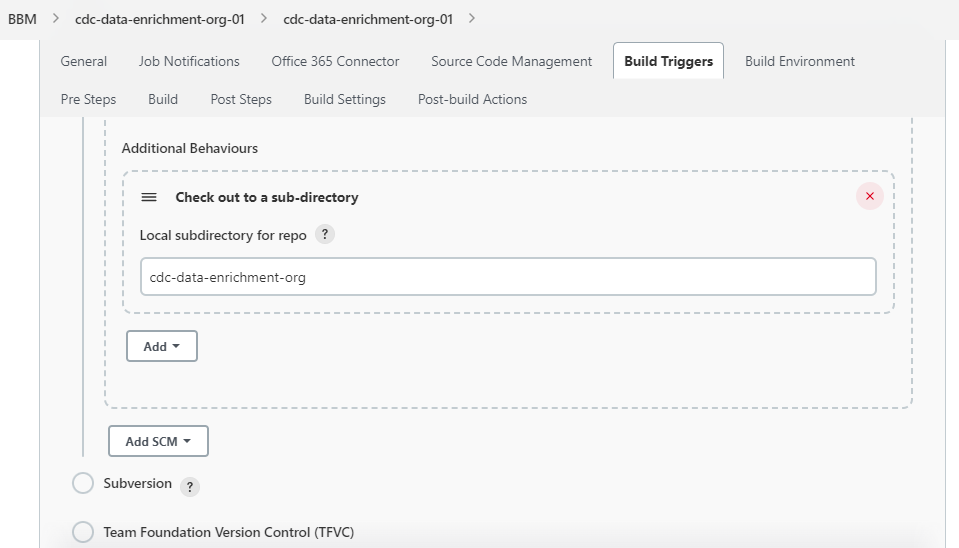


* Click on **Add Repository** and select Git
* Fill theGit **OpenShift** Repo **HTTPS URL**
* Select Credentials as **CPM** **DevOps Functional ID**

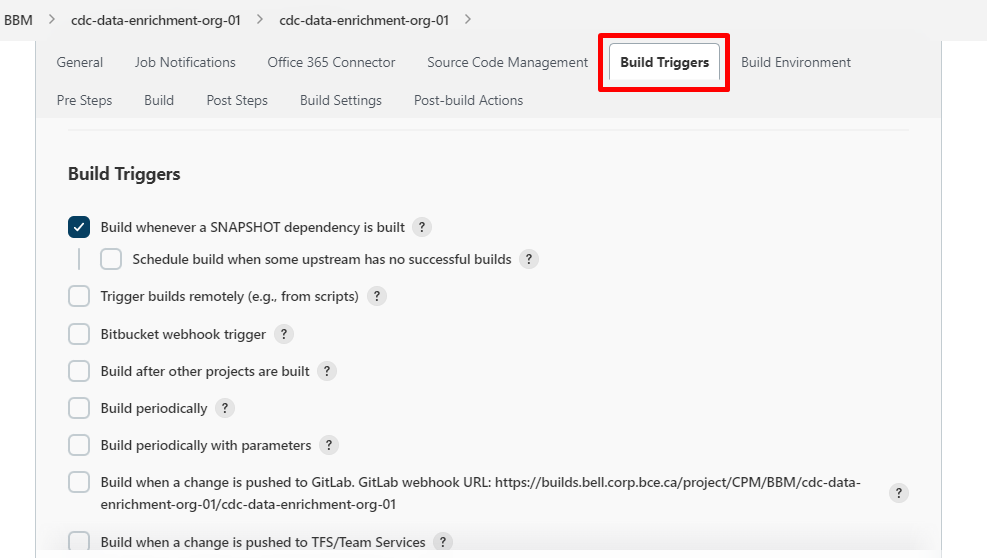




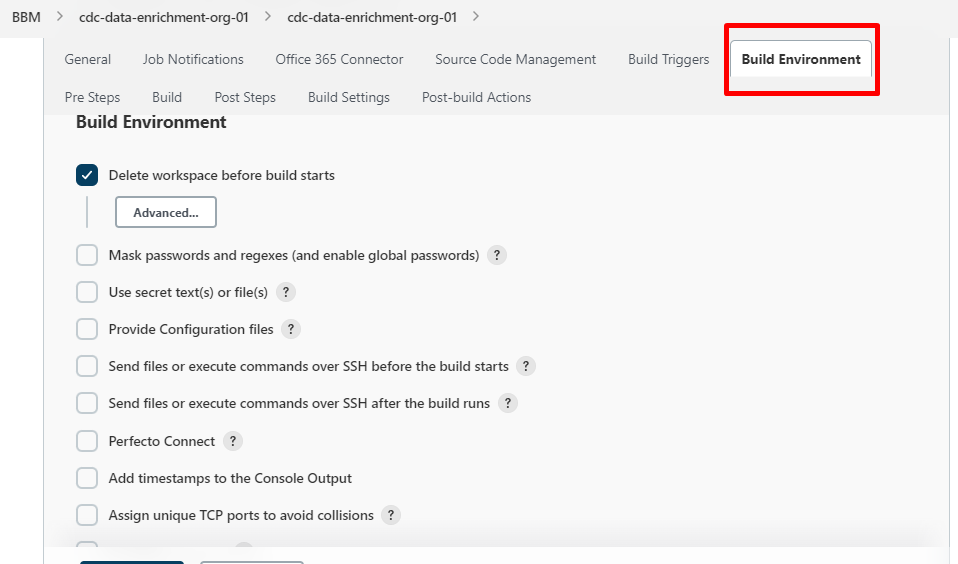
* Click on **Add** and Select **Check out to sub-directory** and enter your directory name



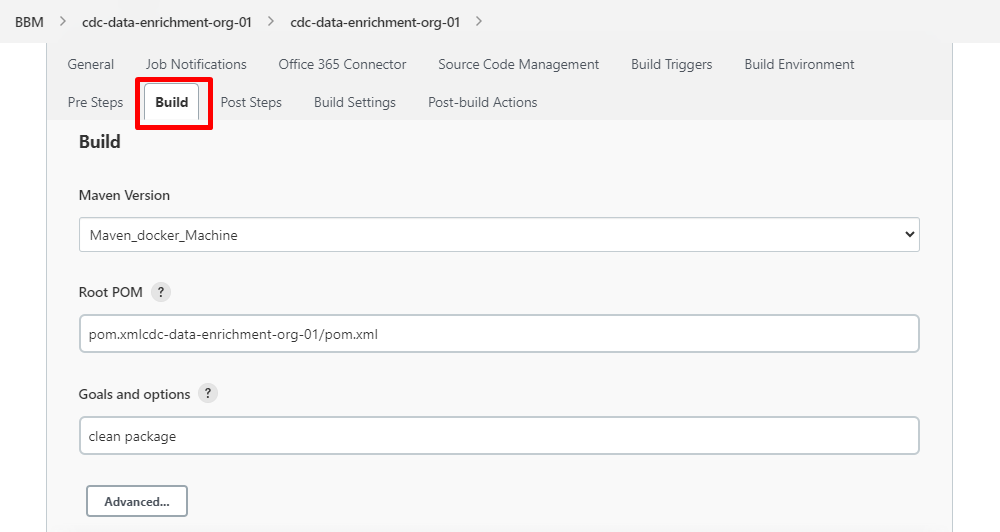
* **Step 6**



* **Step 7**



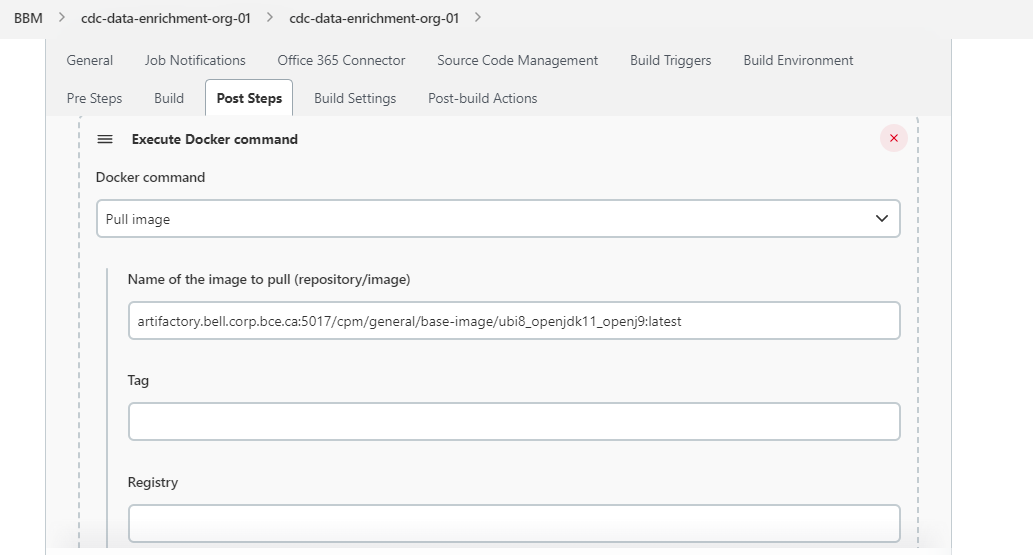
* **Step 8**

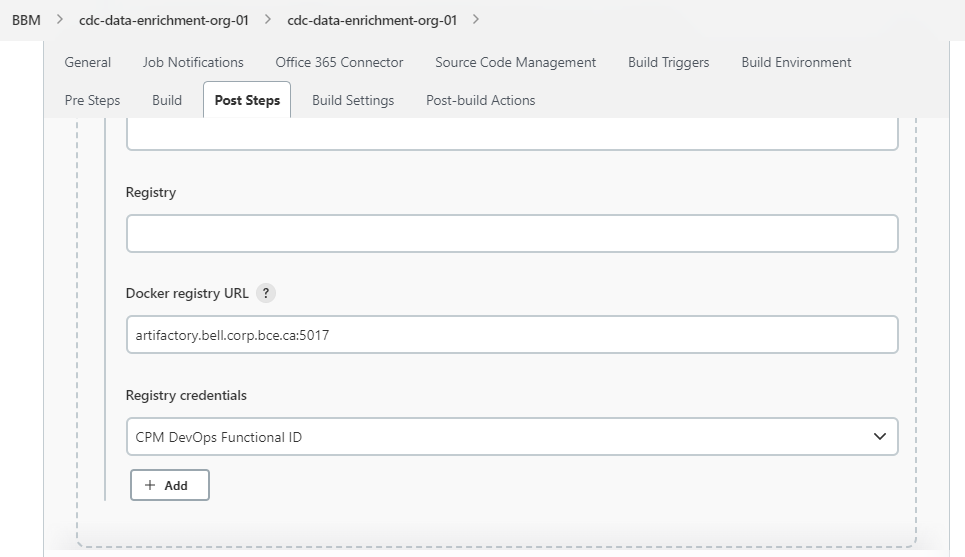


**Step 9**

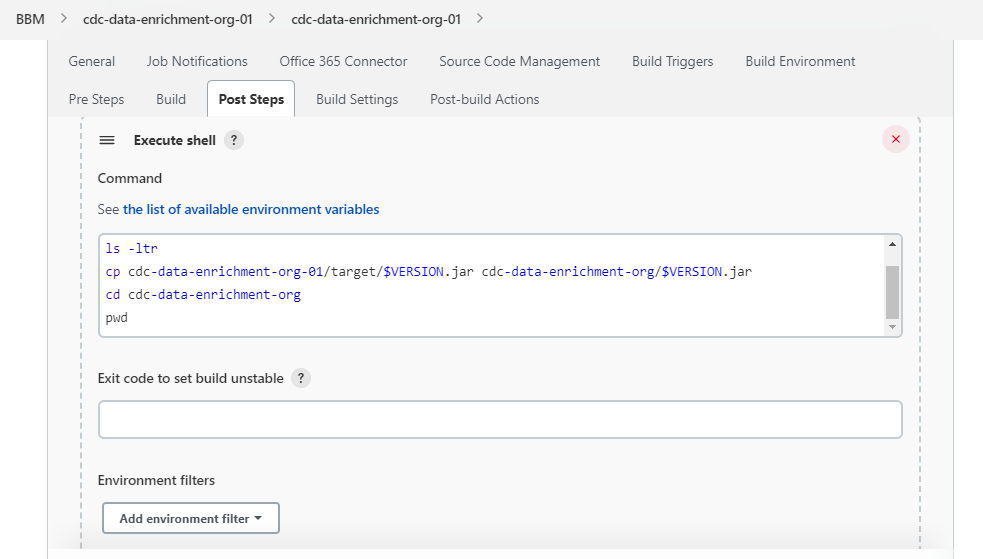


* **Step 10**
* Click on **Add post-build step** and Select **Execute Docker Command**

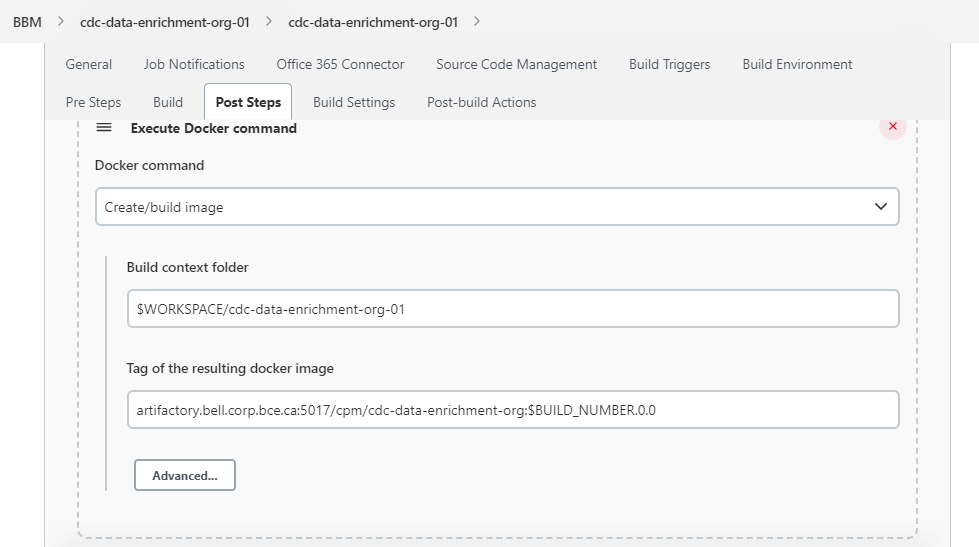




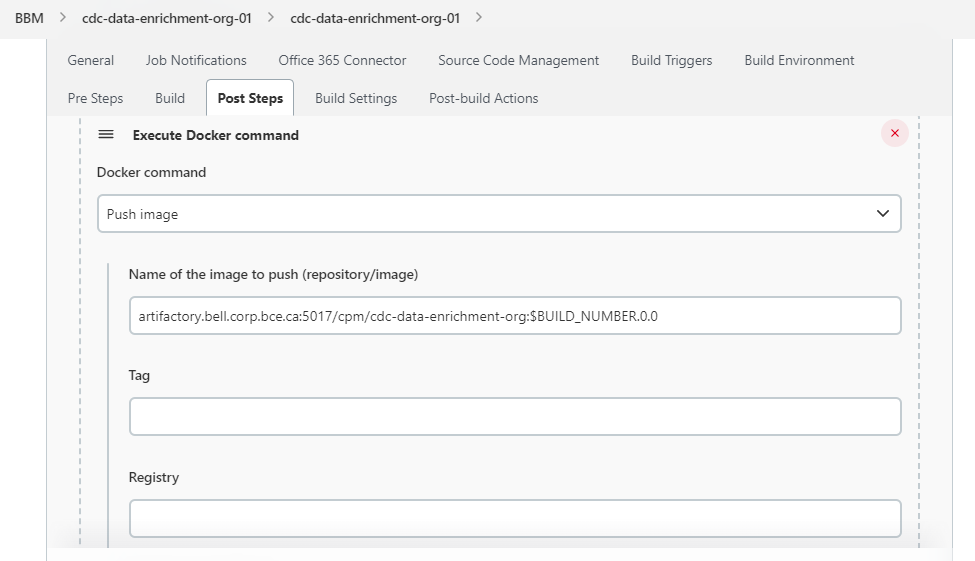
* Click on **Add post-build step** and Select **Execute Shell**

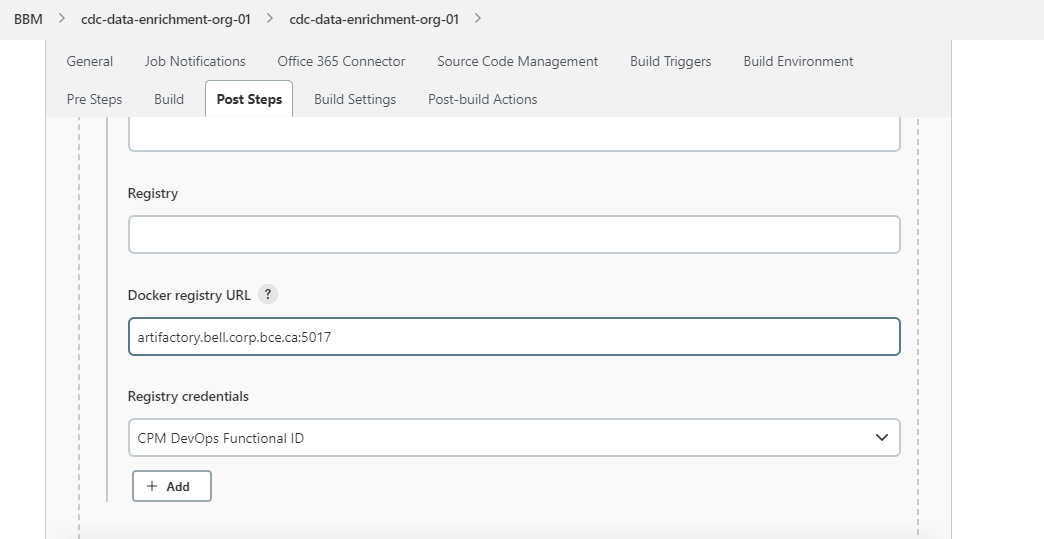


* Click on **Add post-build step** and Select **Execute Docker Command**

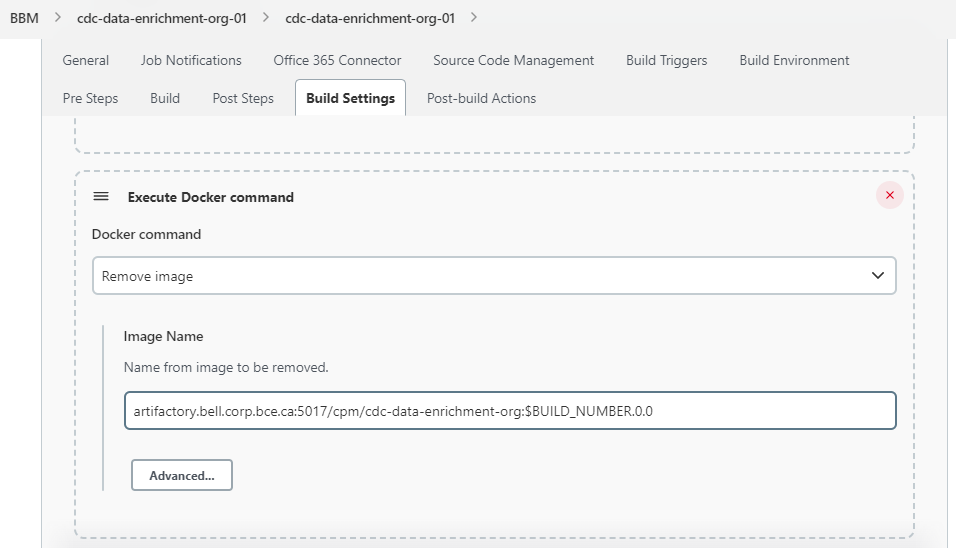


* Click on **Add post-build step** and Select **Execute Docker Command**

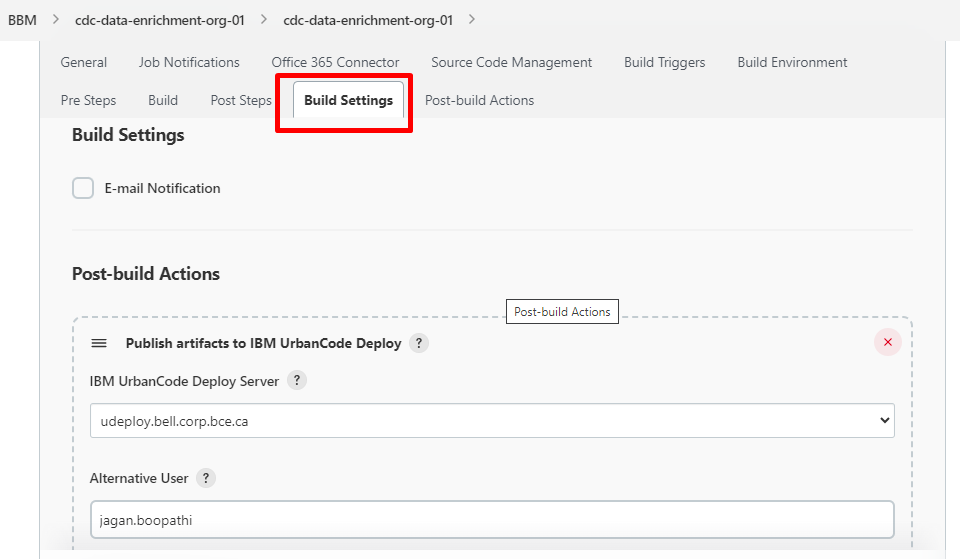


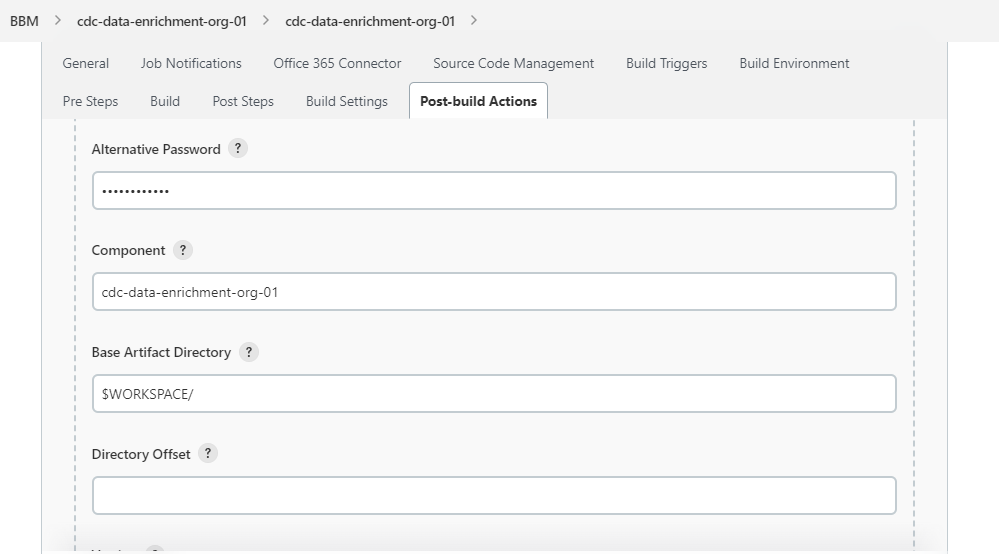


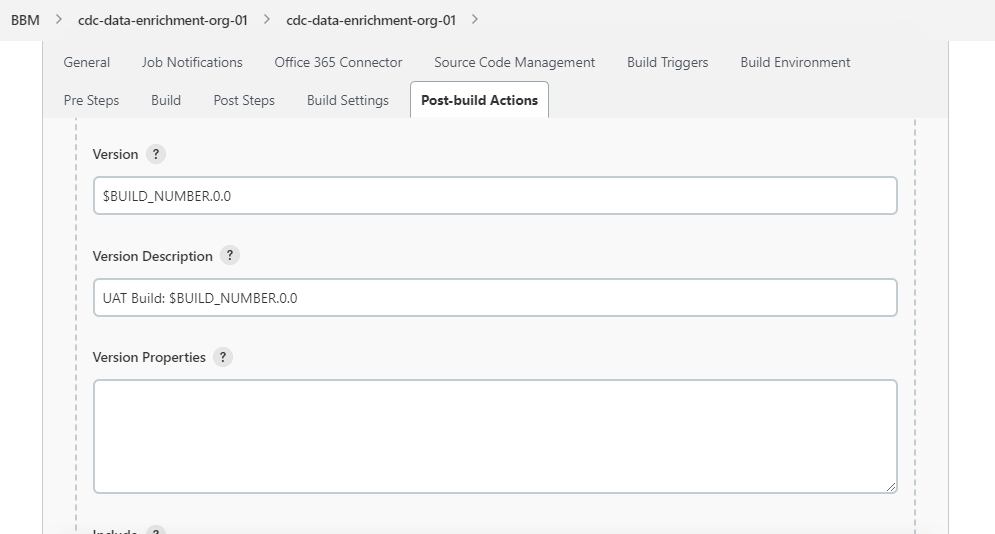
* Click on **Add post-build step** and Select **Execute Docker Command**

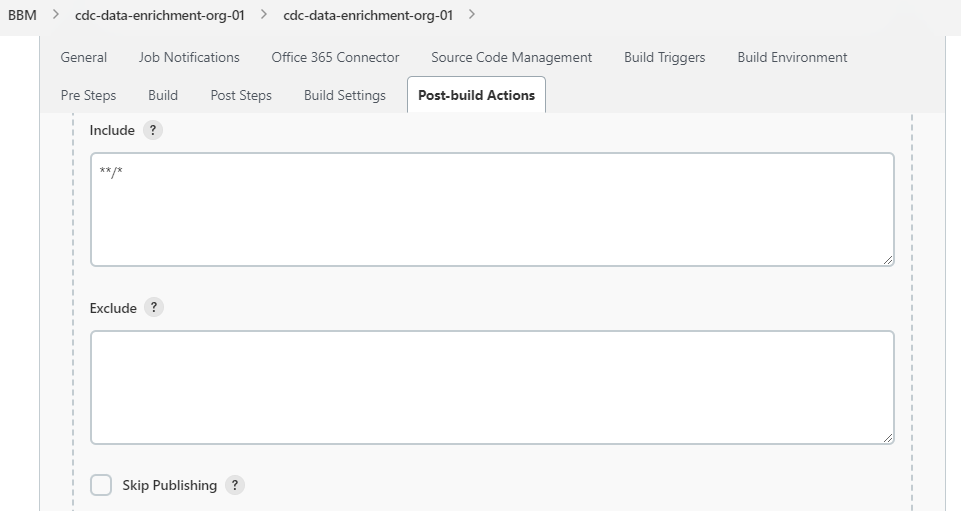


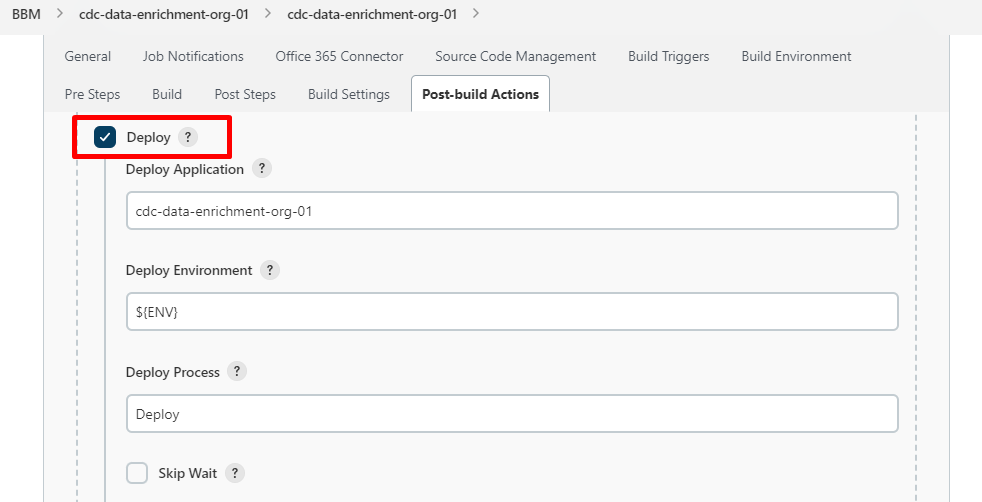
* **Step 11**



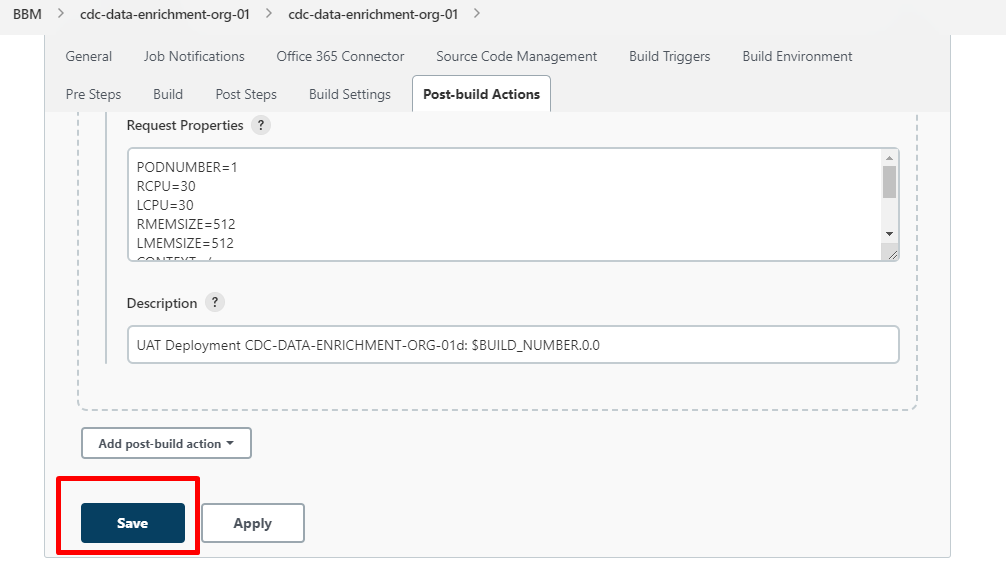




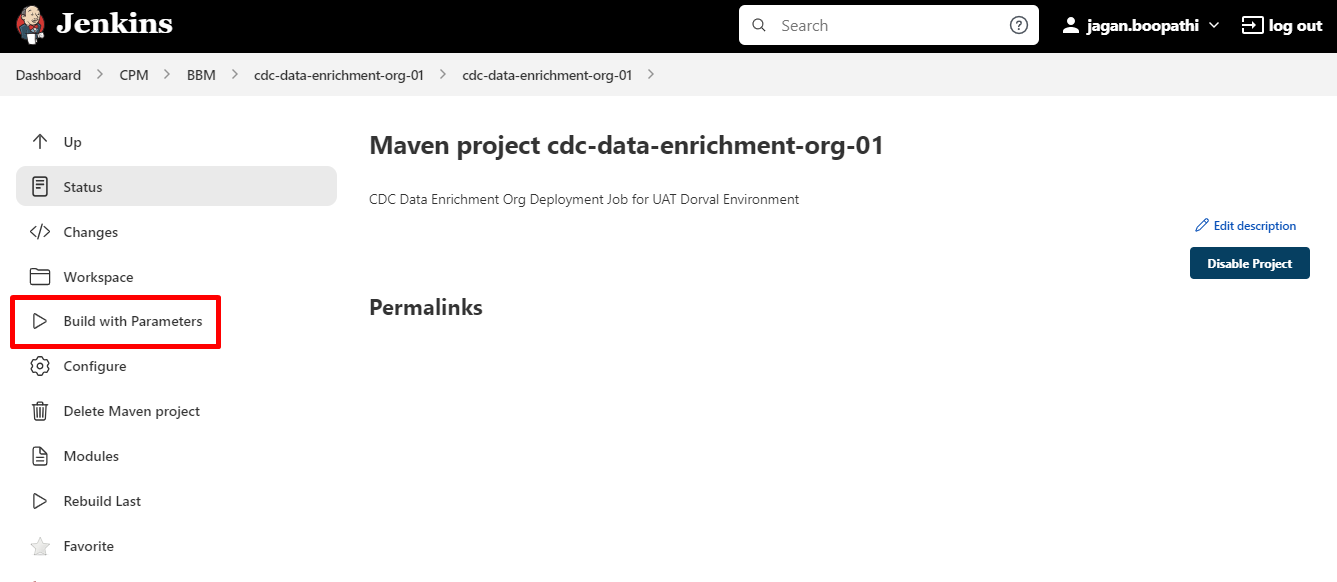




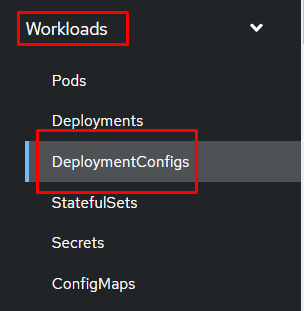
* Request Properties you can copy from existing job/projects
* In last Click **SAVE**



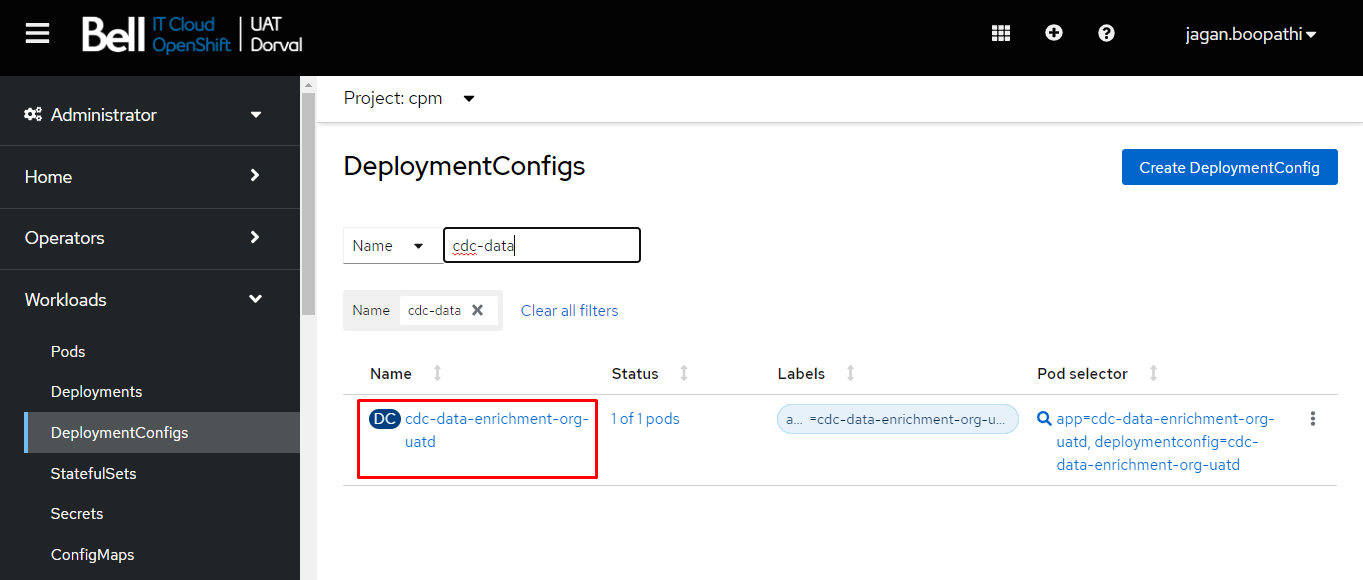
* It will redirect to below page
* Click on **build with parameters**



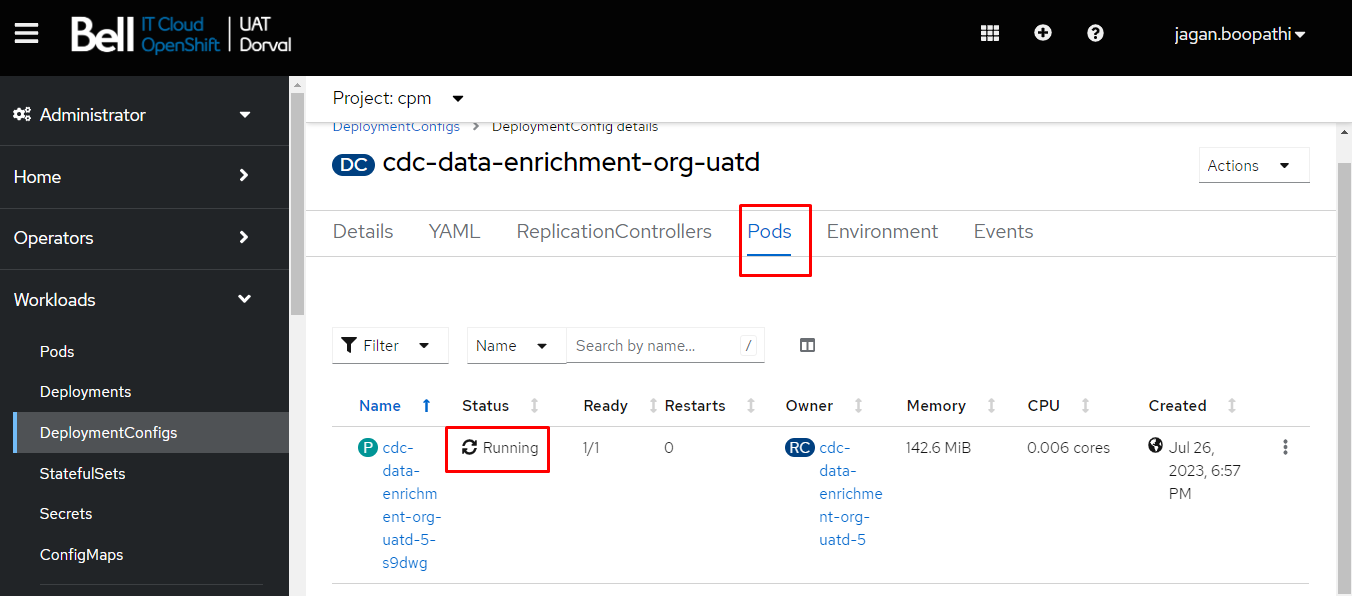
* Your job will get started to run after getting Success Result you can check into Openshift in Deployment configs your application has been created automatically
* Go to Deployment config and check pods status and Logs of your Project/Application
* Under Workloads select **DeploymentConfigs**



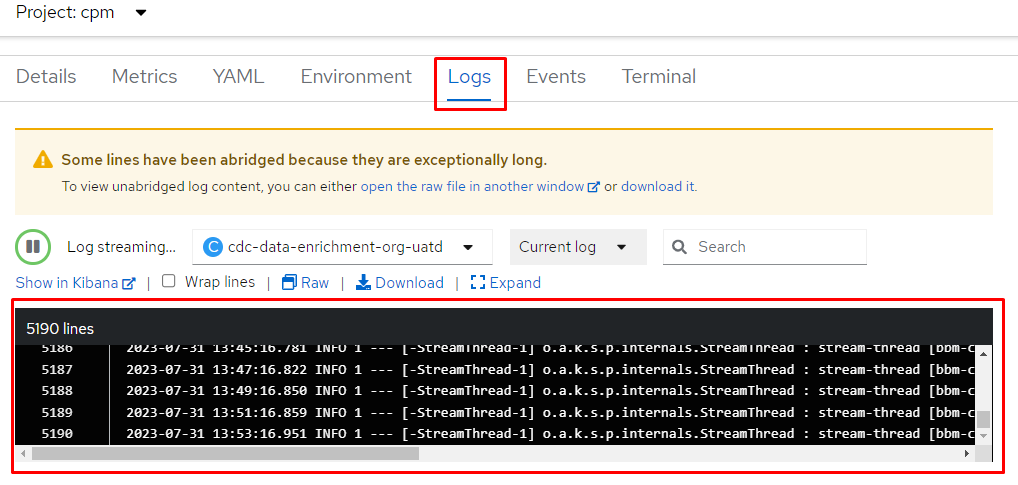
* Search your application name and select



* Go to pods and check the status



* Go to logs and monitor your application is properly running



* If there is no error it means that your application is deployed successfully.