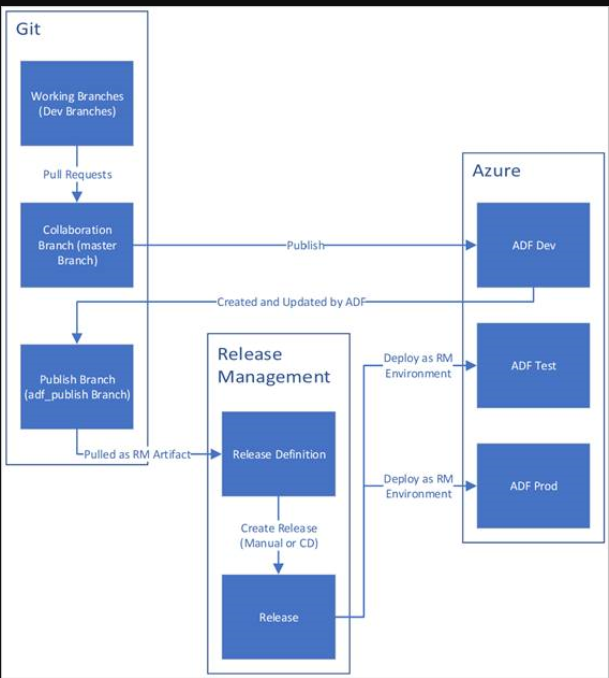
**Azure Data factory – CICD using Azure Devops**

**Overview Diagram:**

****

The above diagram illustrates the source control or code control, release management and Environments (Dev, Test & PROD). Here release management can be done manually or automated by code release from publish branch to Target environments. In Git part Dev branches are considered as feature branches, Collaboration branch is nothing but Master or Main branch.

**Flow of the CICD will be:**

**Dev or Feature branches -> Pull request -> Main or Master or Collaboration branch -> Publish action in ADF ->**

**Continuous Integration**

**-> Change in adf\_publish branch ->Release pipeline in Devops ->Creating a release and pipeline run -> Code or ADF resources in Target environment (UAT or PROD)**

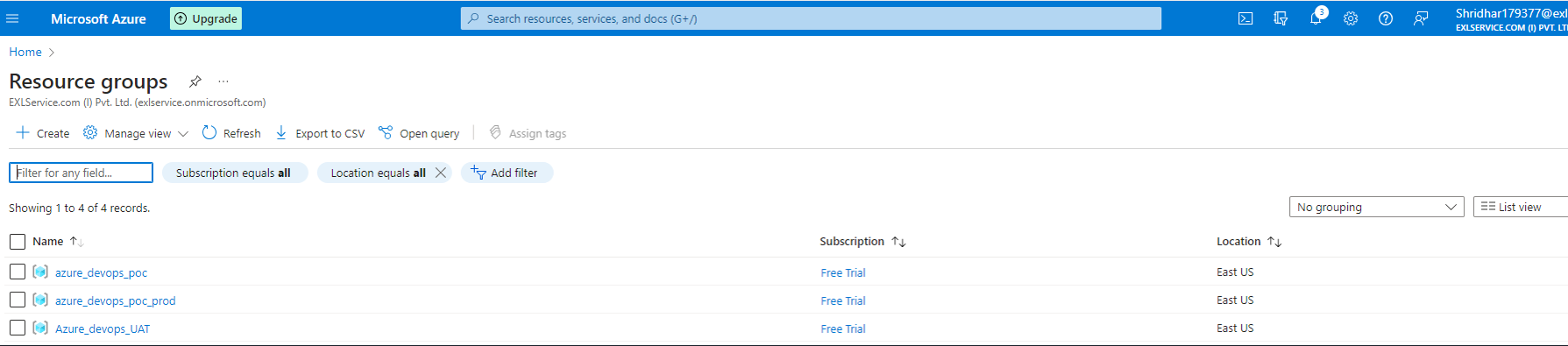
**Continuous Deployment**

The upcoming steps explains what are all the necessary steps and configurations need to be done to deploy ADF pipelines & components from one environment to another.

Dev -> UAT, Dev -> PROD using Azure Devops Git.

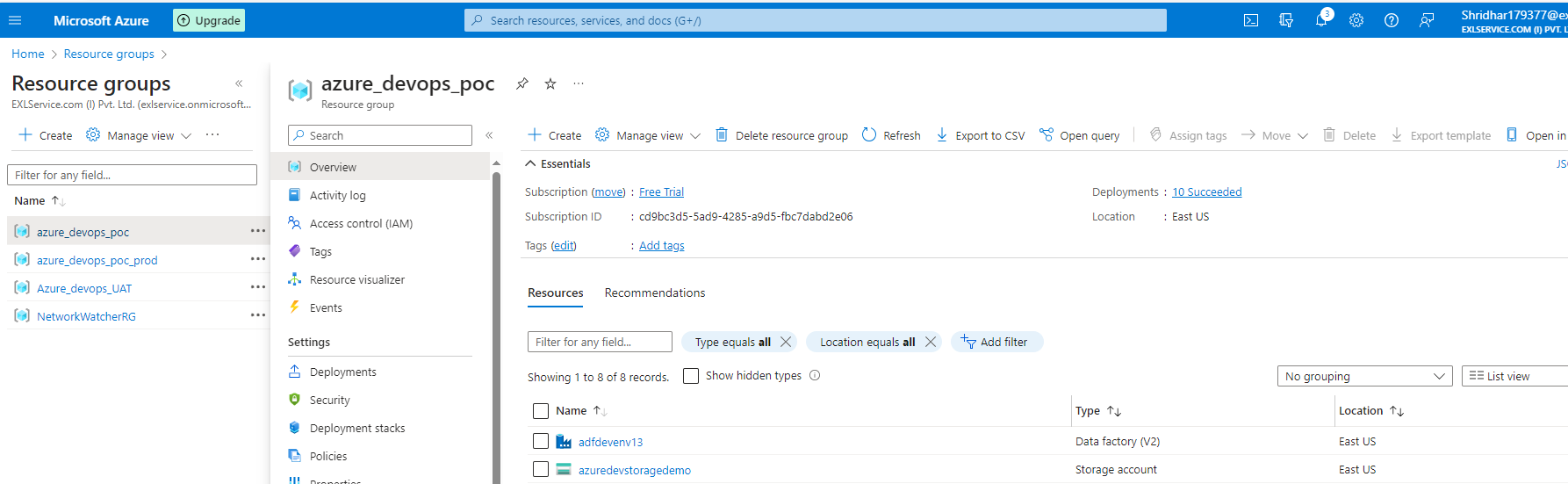
**Step 1:** Create the required resource groups i.e., resource groups for Dev, UAT and PROD in Azure portal. Where this each resource group should have ADLS Gen2 and Azure Data factory.

Resource Groups:

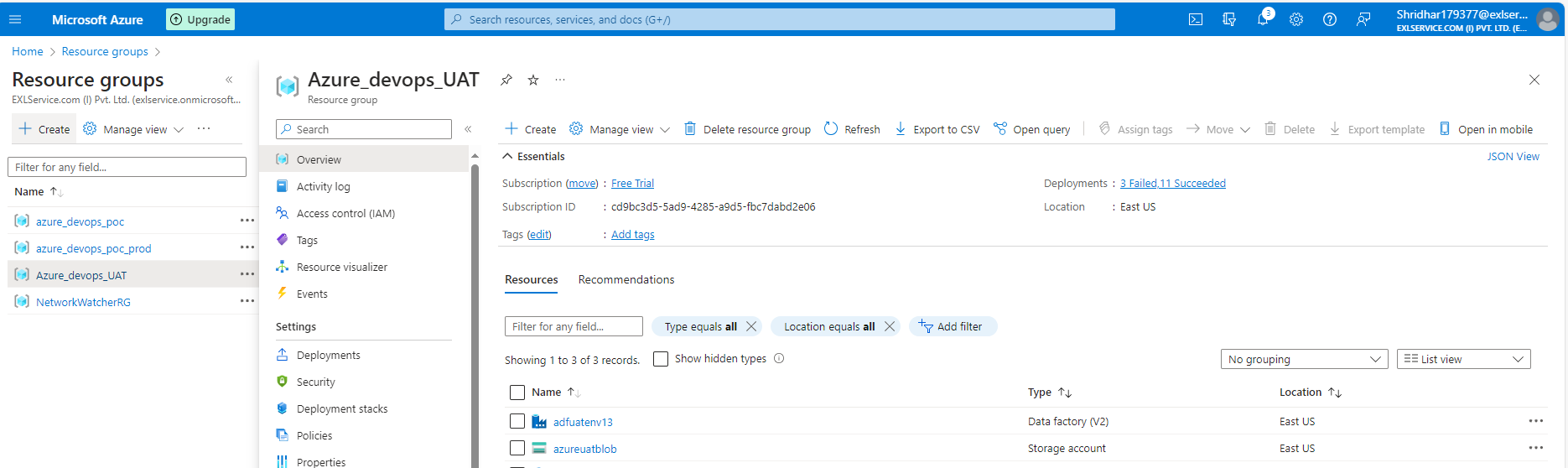


Each resource group with ADLS Gen2 & Azure Data factory service:

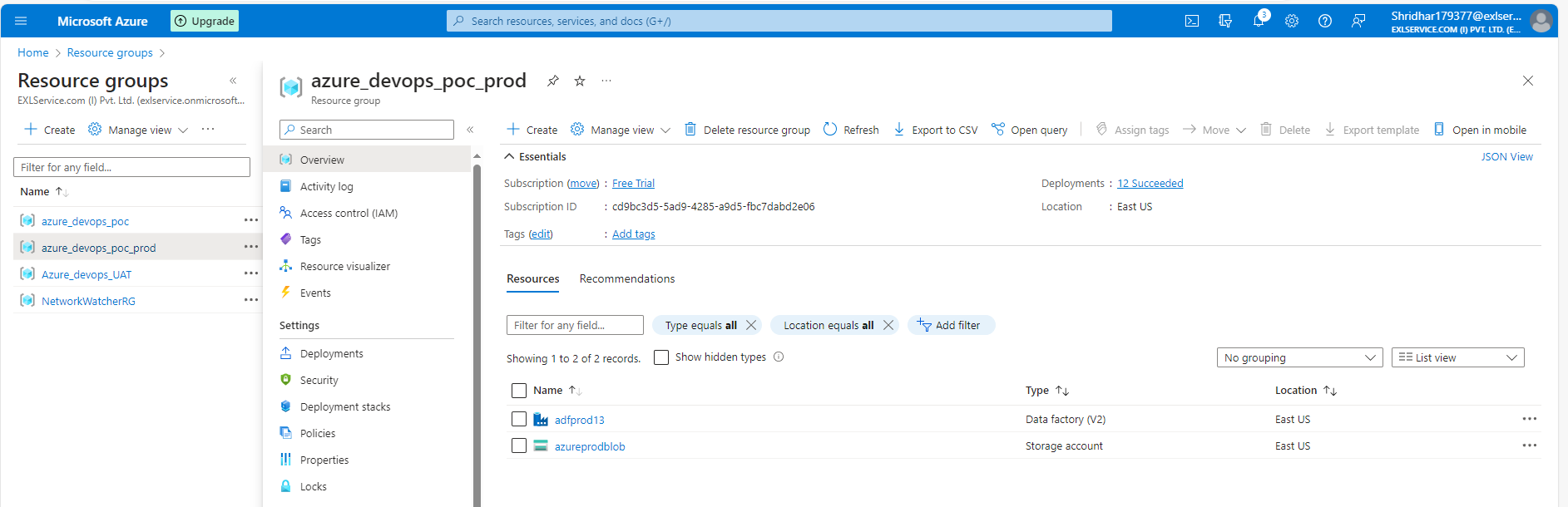
**Dev**: Resource Group azure\_devops\_poc holds one ADLS Gen2 and Azure Data factory.



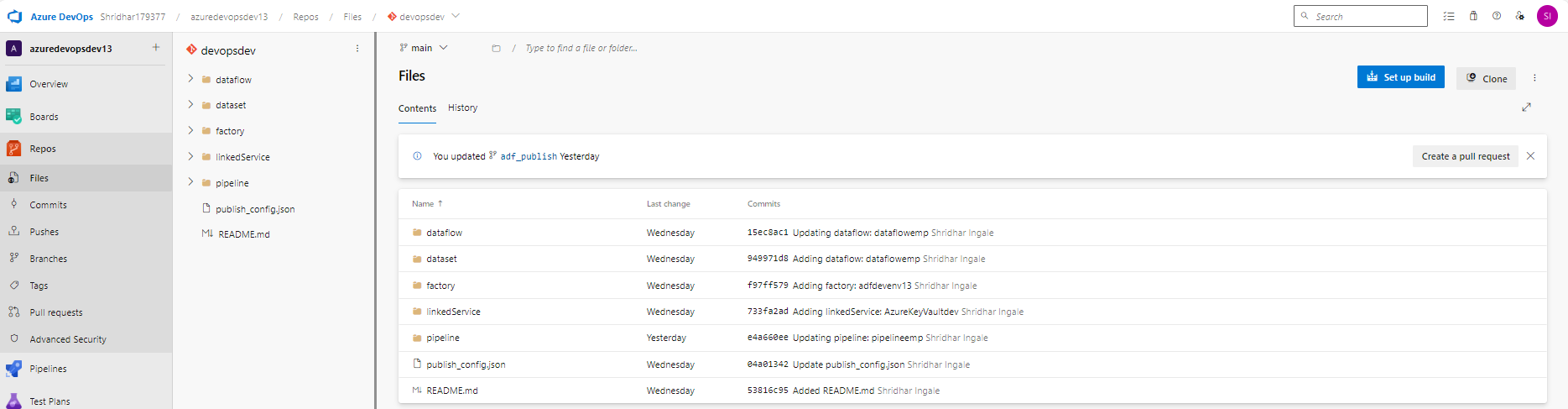
UAT: Resource Group Azure\_devops\_UAT holds one ADLS Gen2 and Azure Data factory.



PROD: Resource Group azure\_devops\_poc\_prod holds one ADLS Gen2 and Azure Data factory.

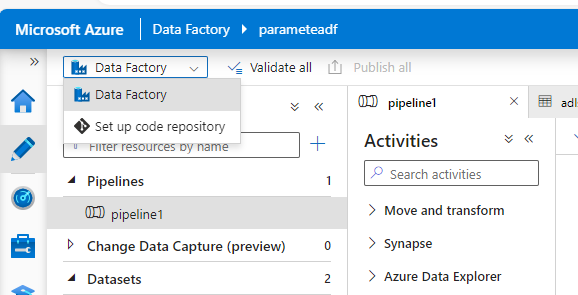


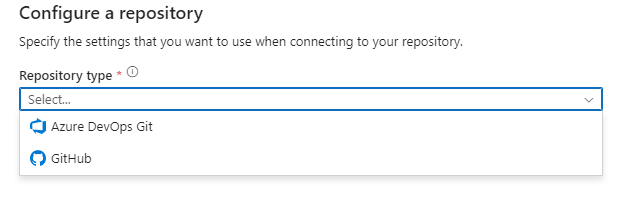
**Step 2**: Create Project in Azure devops. After creating a project create repository for the Dev environment i.e., for Dev Azure data factory.

.

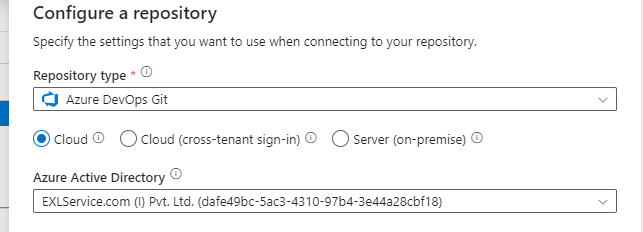
Here project is azuredevopsdev13 and DEV repository name is devopsdev. The DEV repository automatically creates and holds all the factory resources as folders. As shown in above screenshot.

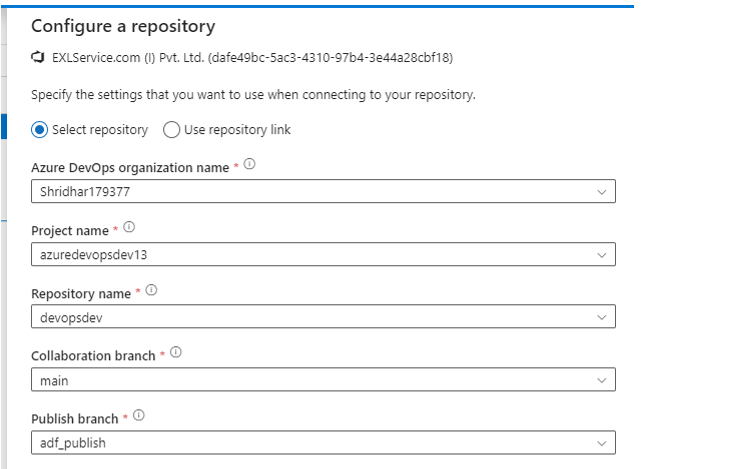
In Azure data factory UI, the azure provides us option to connect the ADF (set up code repository option) with created Azure Devops Git as shown below.



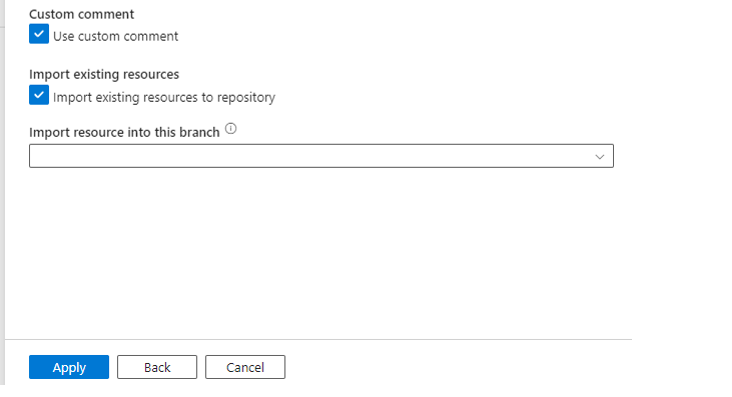


Select azure devops git option and configure the appropriate details to set up the devops code repo in ADF.

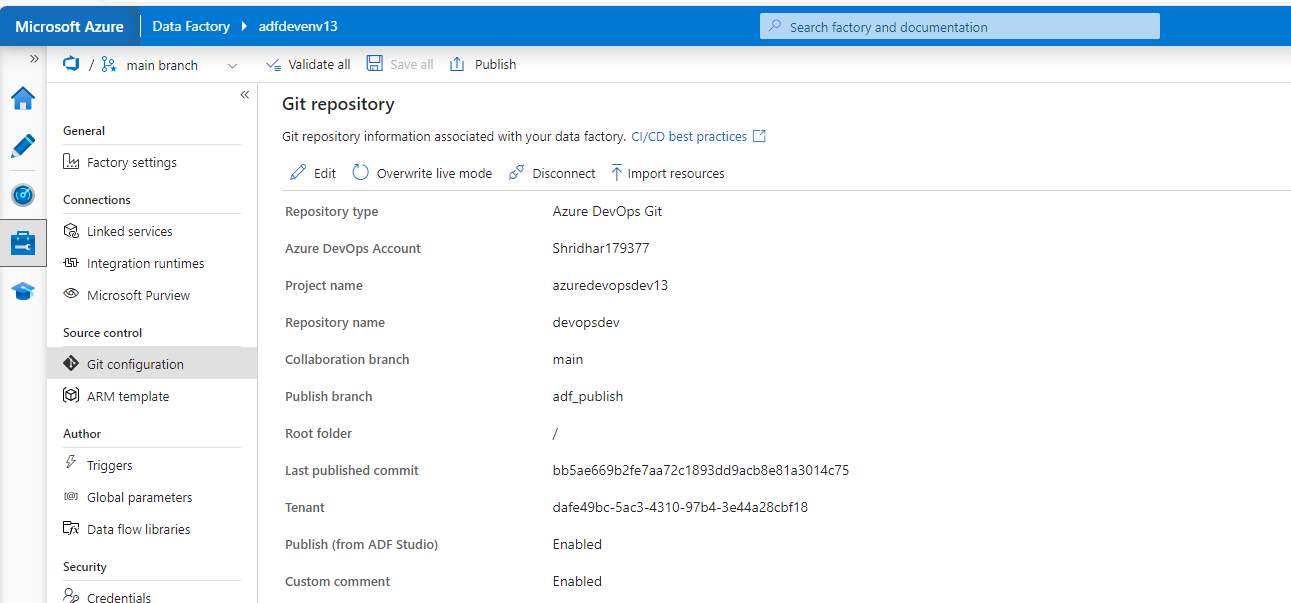




After configuring all the options, click the apply button shown in below screenshot.

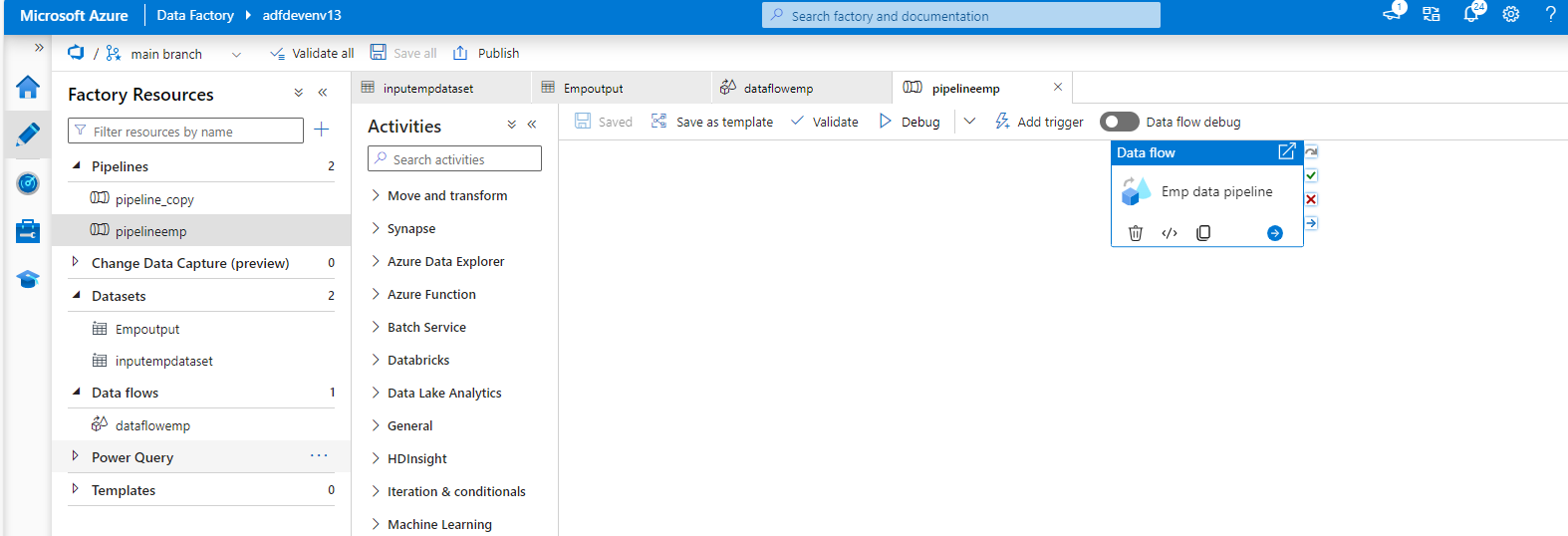


Once the ADF is connected to Azure devops repo we can see the folders in Azure devops repository as show above and also in ADF devops Git configuration can be viewed.

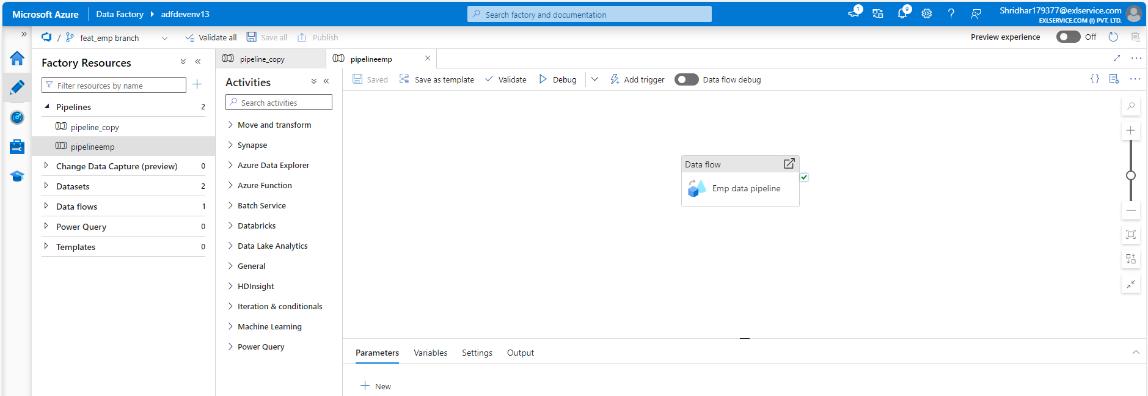


**Step 3**: In Dev ADF, main branch is automatically created when the repository is created. Even we can create a feature branch, so that Dev person can do the change in that feature branch and merge the change to main branch.

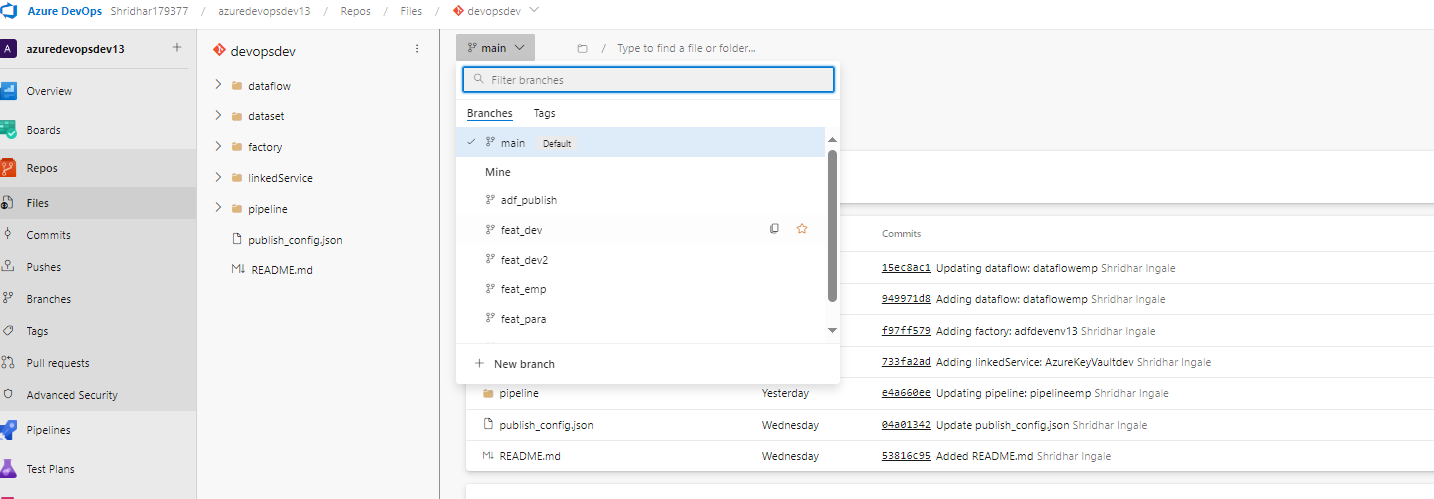
Main Branch:



Feature Branch:

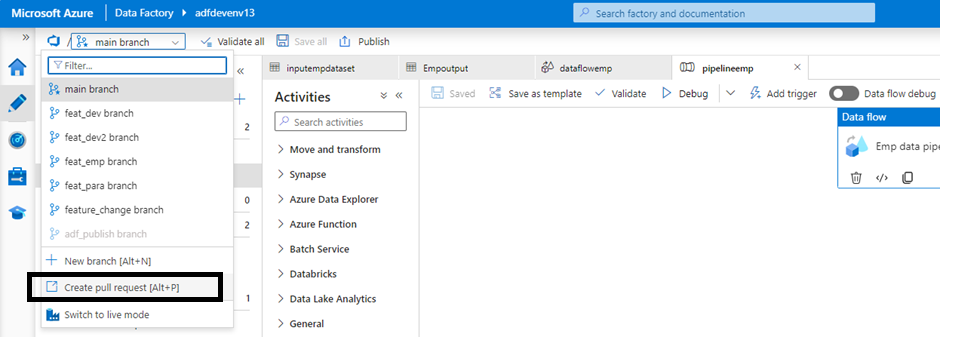


Feature Branches structure in the Azure Devops:



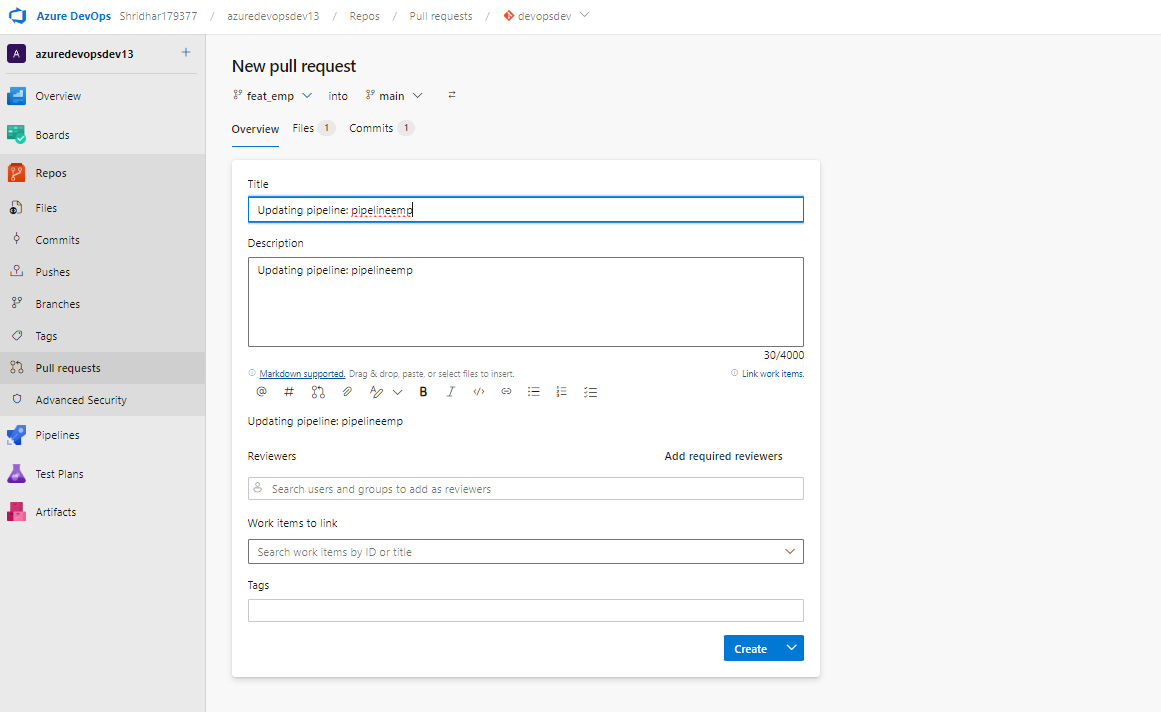
adf\_publish is branch created default in ADF based devops repos. Where the latest changes as published here as json template which is used later for CICD.

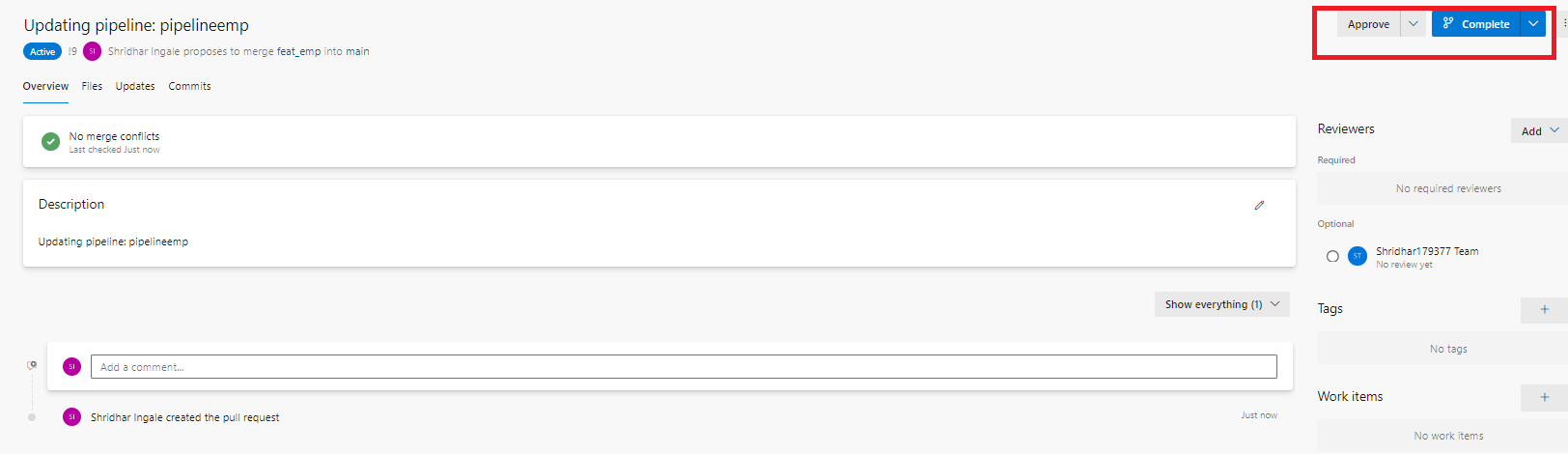
**Step 4:** Create a pipeline or edit an existing pipeline in any feature branch and save it. Once the ADF is ready now create a pull request from ADF interface as shown below.



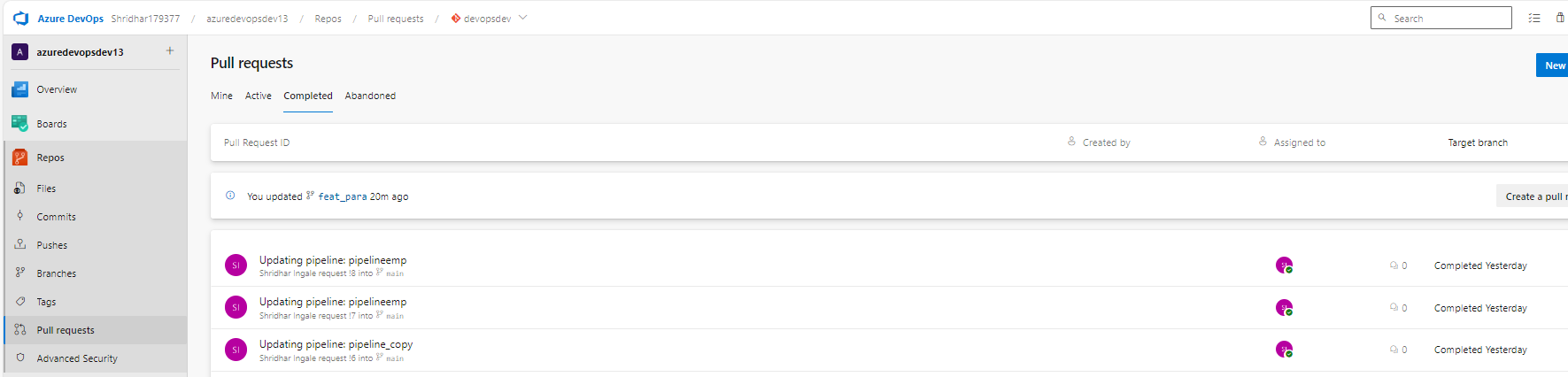
This pull request will be shown in Azure Devops and appropriate person should approve it. Once the approval is done, the feature branch change will be applied or collaborated in main branch.

Pull request window in Devops:





After approval:

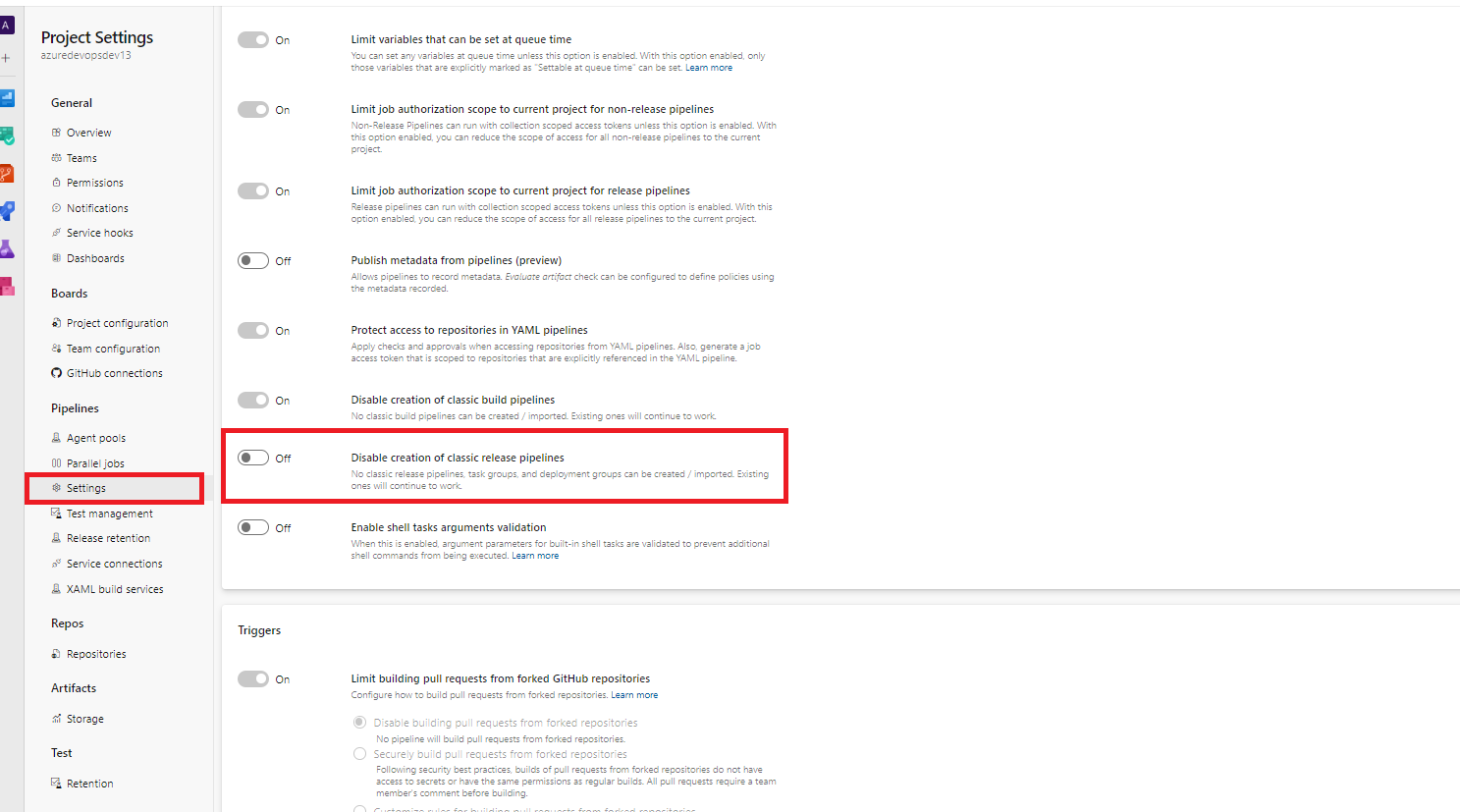


Here all the pull request is shown and it is also approved accordingly w.r.t to the change.

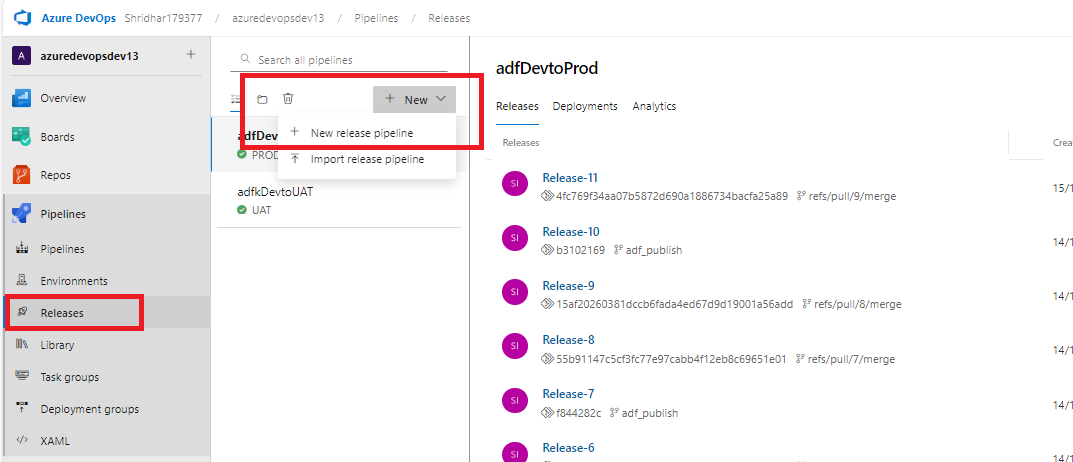
Once pull request is approved mail branch will have the updated ADF version.

**Step 5:** Till now the part which the above steps are involved is Continuous Integration. For continuous deployment we need to configure the release in the Azure Devops Interface.

Before creating a release pipeline, a setting has to be disabled in Project settings so that Release option will be available for the project deployment release.

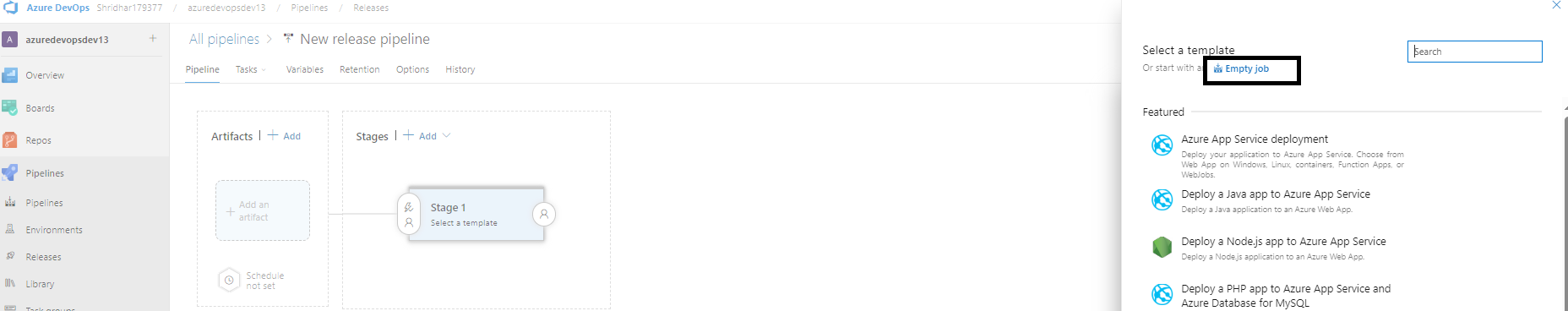


After disabling above option, release option will be enabled for the project.

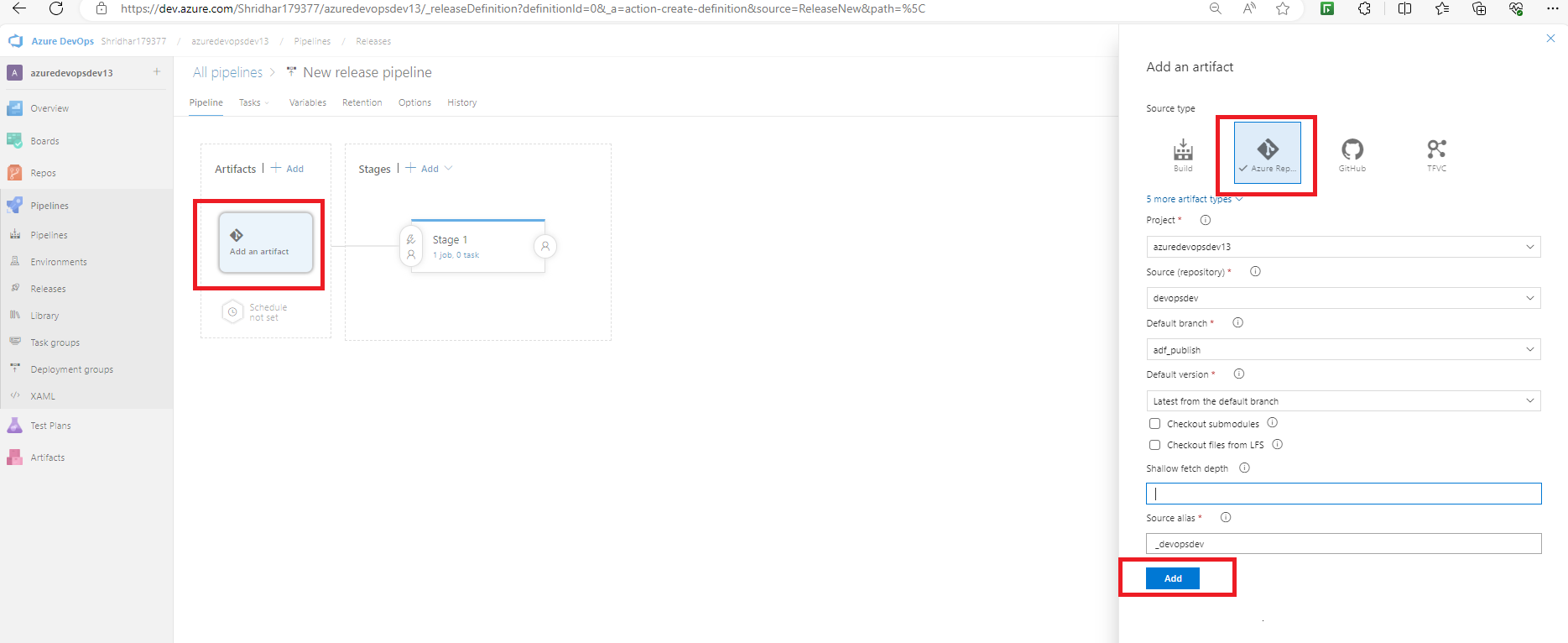


To create a new release pipeline, click on the Release option and click on the new release pipeline option as show in the above screenshot.

After clicking the new release pipeline option, the empty job will be show as below screenshot. Click it.

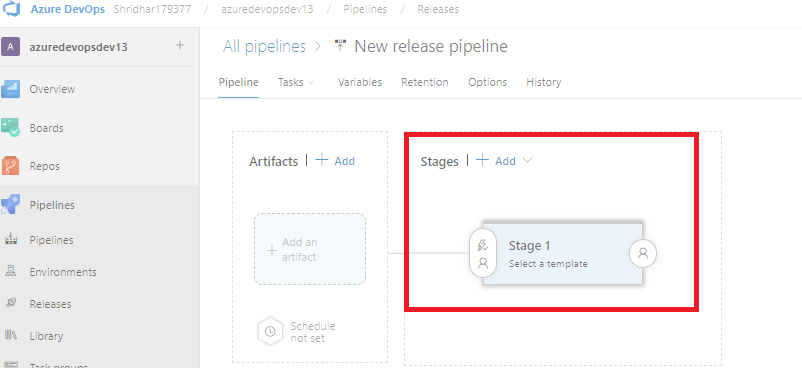


Once the empty job option is selected, we need to configure the Artifacts for the empty job.



As shown in the above screenshot click on the artifact-> select source type Azure Repository -> configure the required repo details accordingly (Here adf\_publish branch should be selected as default branch) -> click Add button to finish.

**Step 6:** Configure the Stage section in the release pipeline.

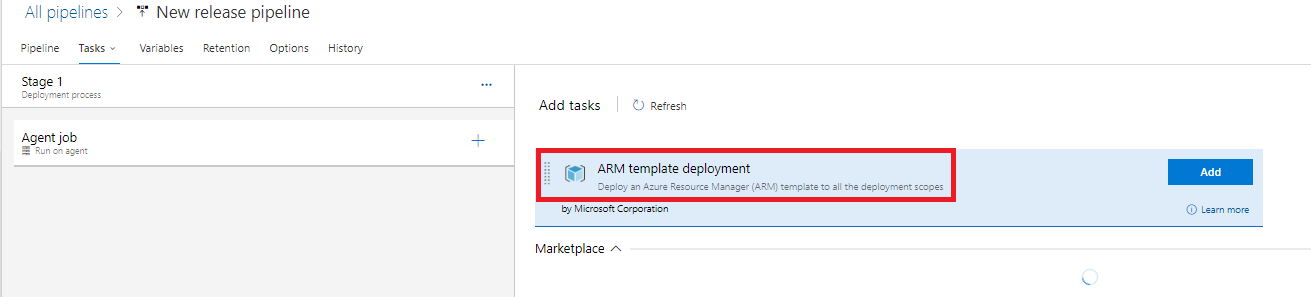


Click the Stage section as shown in the above screenshot, the agent job option will be popped in the screen to configure the destination environment.



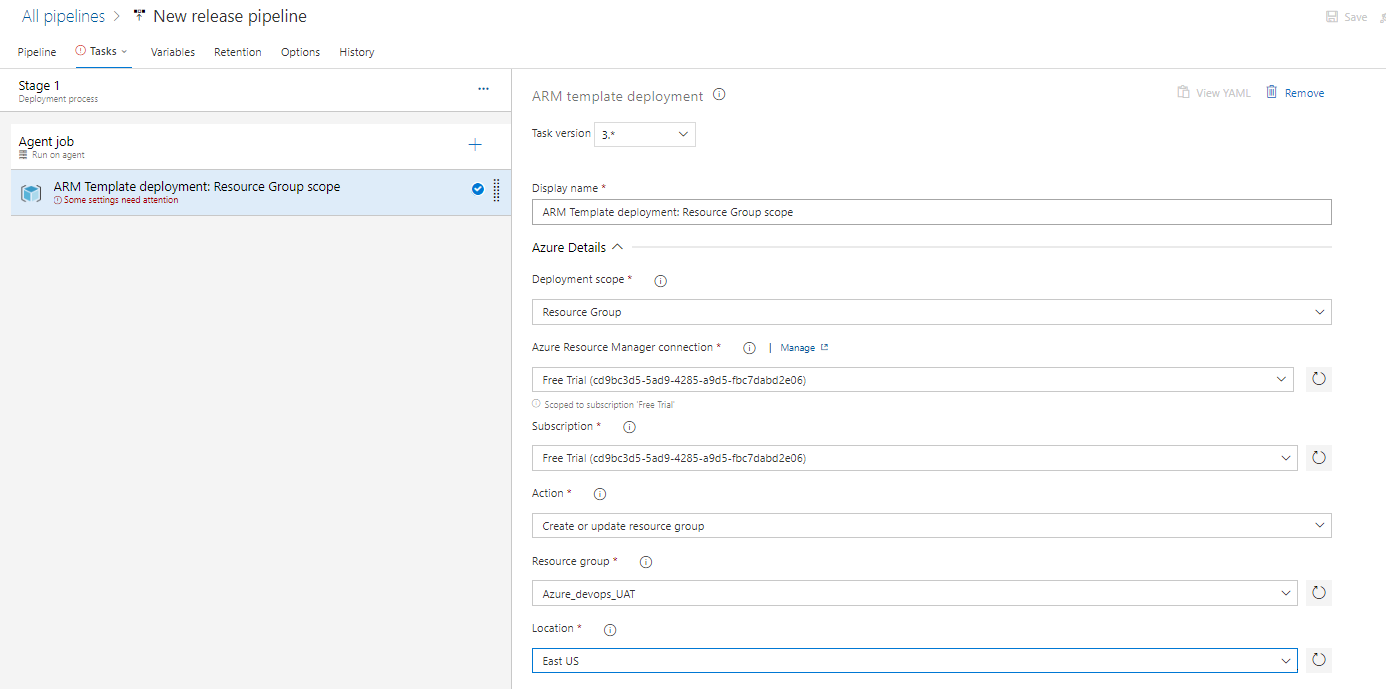
By clicking + option, we can add the required task (i.e., Deployment template or options listed by the Devops).

To add the task template, in the search option type ARM and select the ARM Template deployment task for the stage section as shown below.

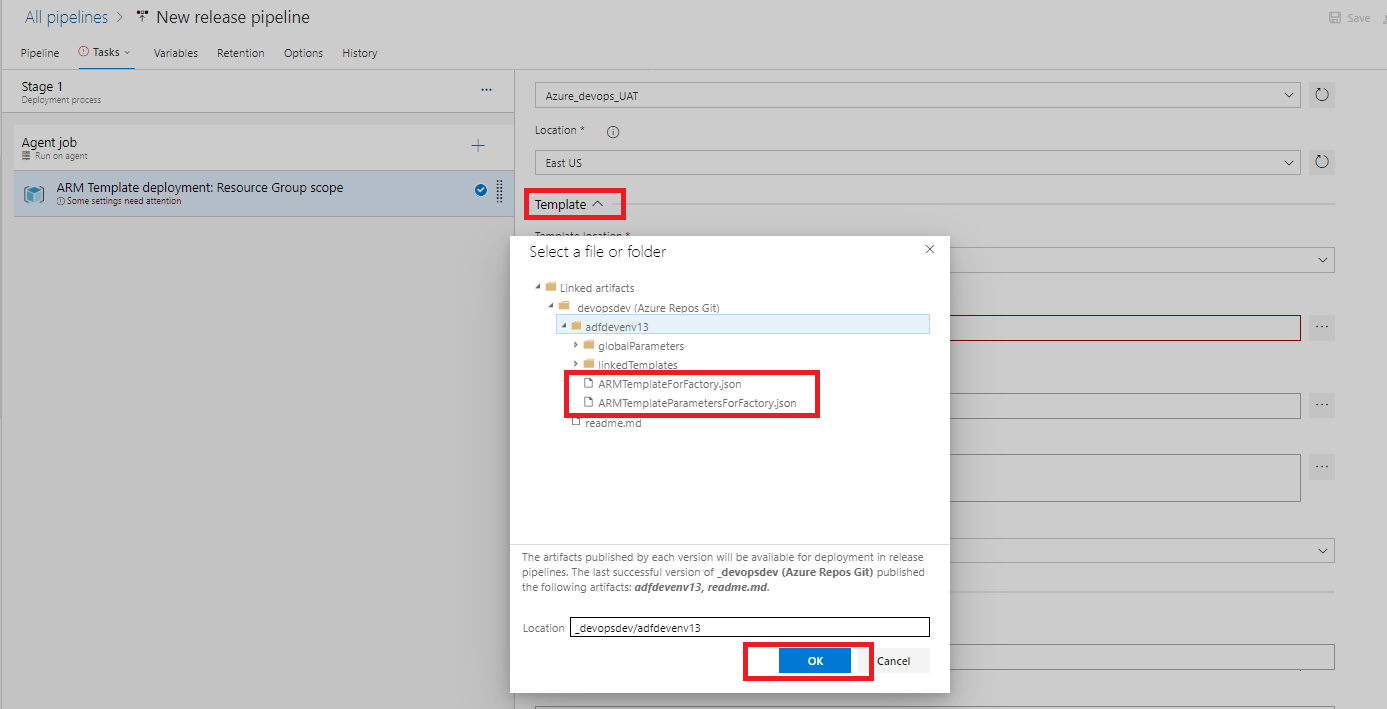


Click Add option.

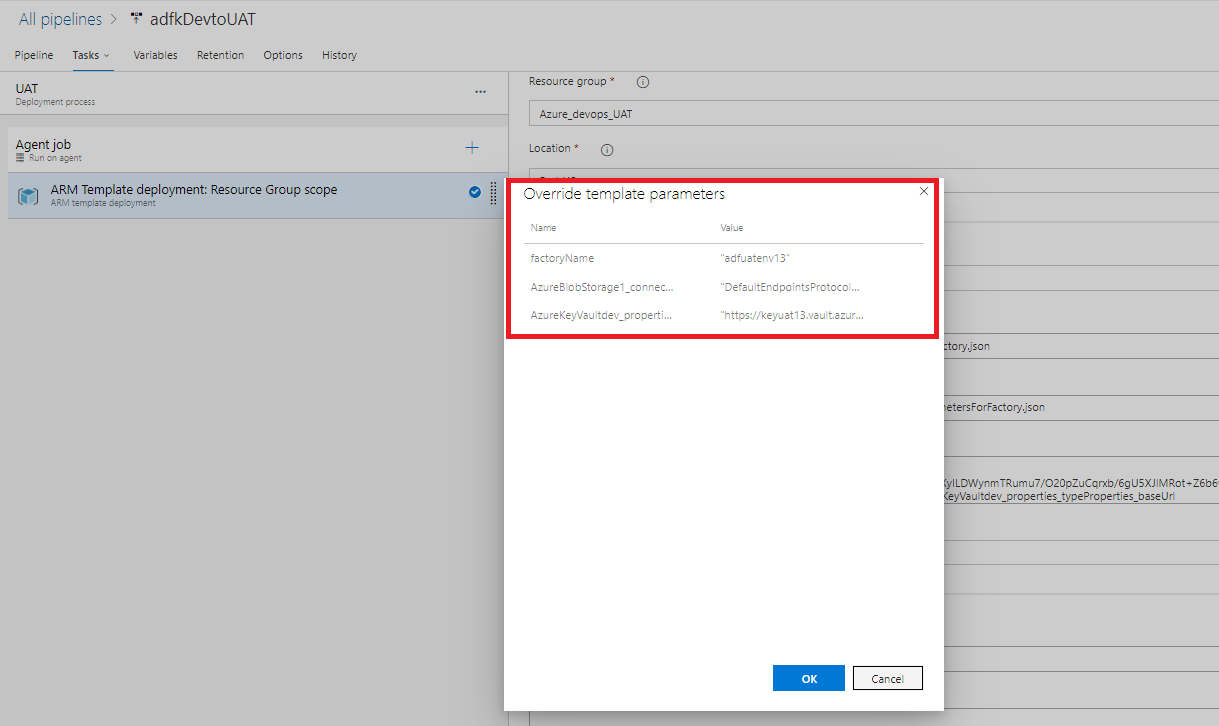
**Step 7:** After adding the ARM Template deployment task to the agent job, configure the job or the task w.r.t the target environment (UAT or PROD Environments).



While configuring the ARM template agent job there are few important aspects that need to be done accordingly.



As we are using the ARM template deployment, we need to provide the ARM template which generated by the ADF during publish action and this template is saved as json in adf\_publish branch. So, the Data factory Json and factory parameter josn need to be configured accordingly as shown in the above picture.

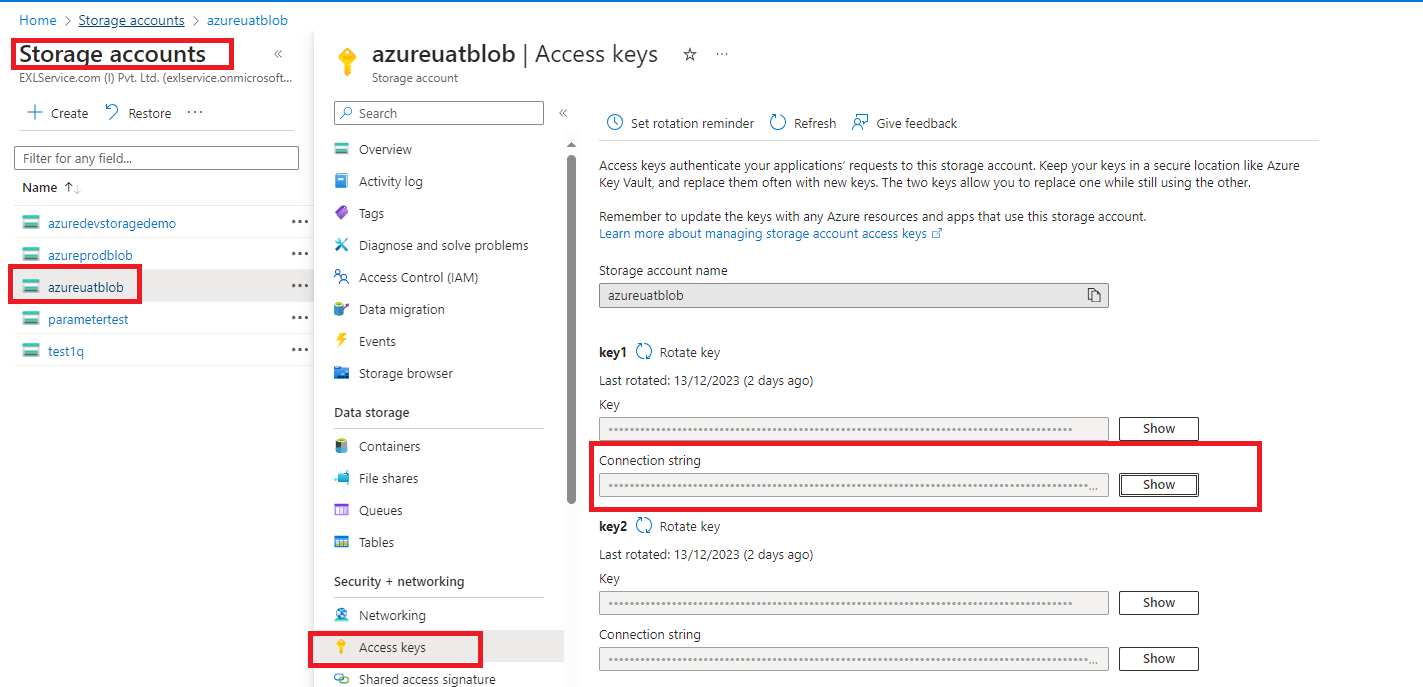


Once the ARM Json is configured then in override template parameter option we need to configure the Target (UAT or PROD Environments) factoryName, Target (UAT or PROD Environments) azure blob connection string.

Parameter:

factoryName -> Target (UAT or PROD Environments) data factory Name

blob connection: for blob connection string, go to the Target (UAT or PROD Environments) storage account -> click on key access -> copy the connection strings as shown below.



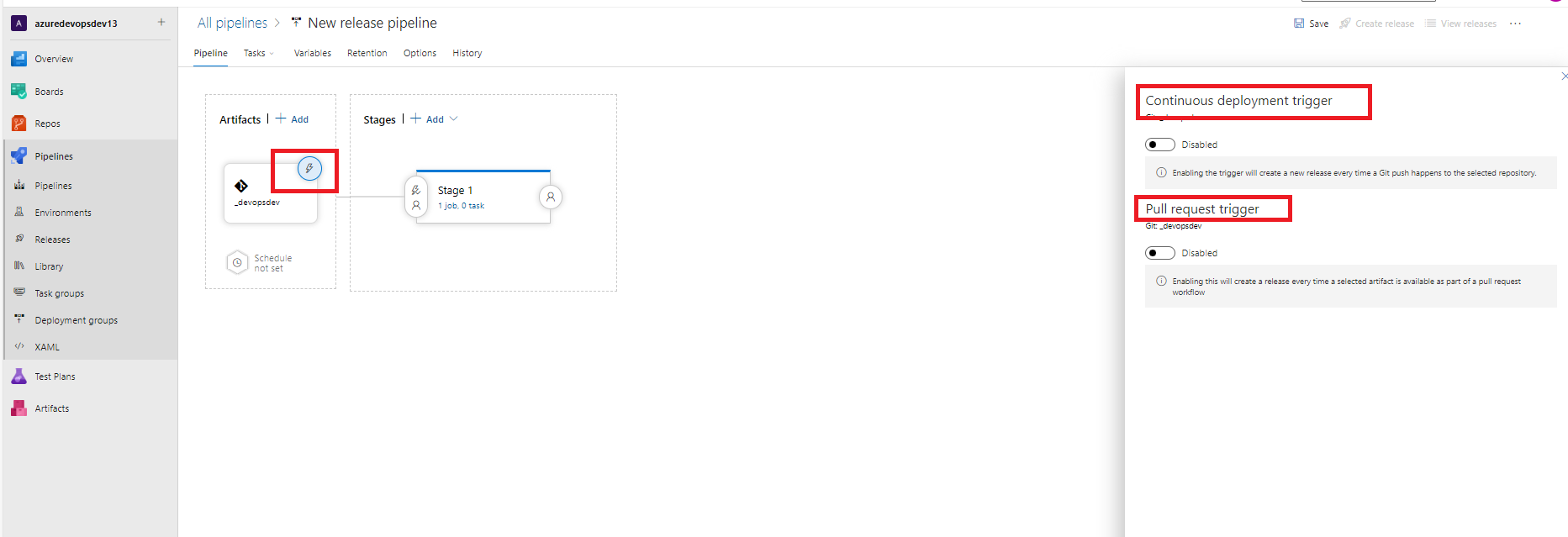
If the value is not properly configured the release pipeline job will fail.

Save the whole release pipeline and also, we can create a release manually for the same to test.



By clicking this create release option, we are manually deploying the DEV ADF resource to the Target (UAT or PROD Environments) environment.

**Step 8:** For continuous deployment we need to enable few options in the Artifacts and Stages section of Release pipeline.



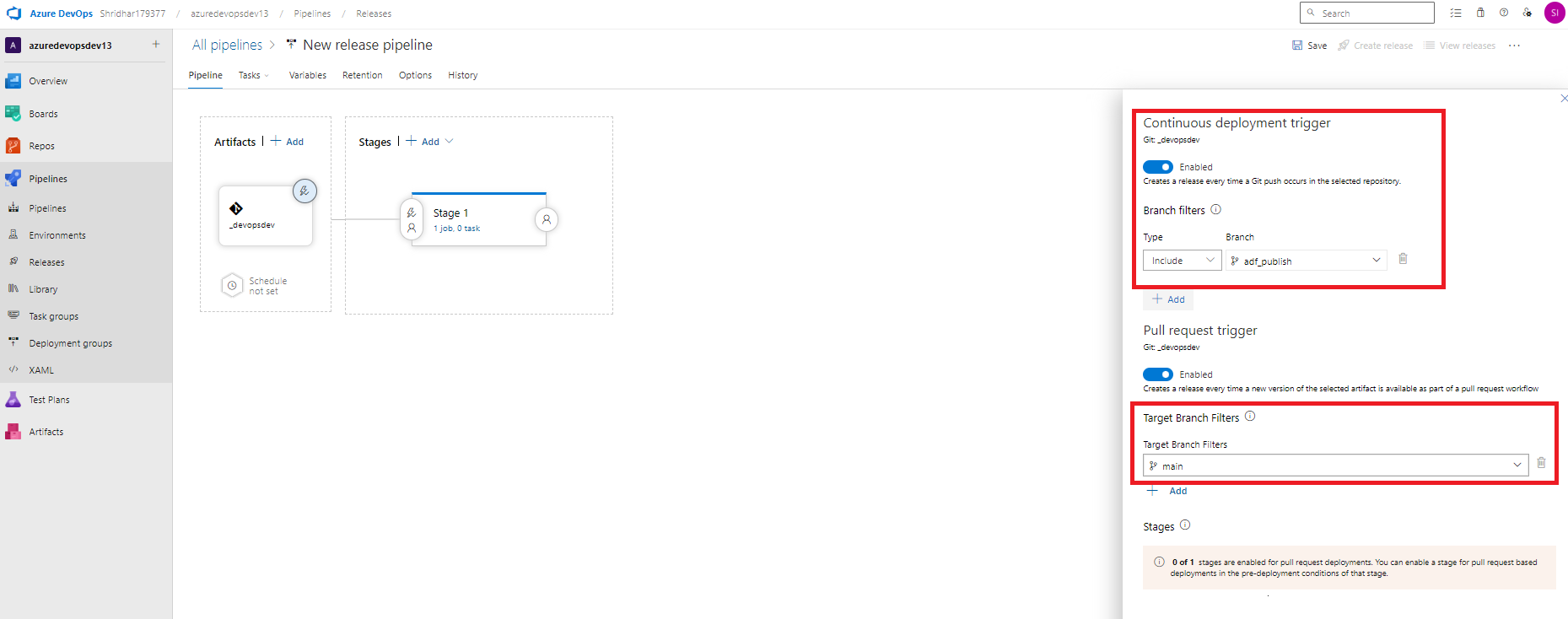
As shown in the above screenshot, click on the artifact’s continuous deployment trigger where it will provide the continuous deployment trigger option as well as Pull request trigger option.

Enable both the buttons and provide the appropriate branch options (*In continuous deployment trigger configure the adf\_publish branch and in Target Branch filters configure collaboration branch i.e., main or Master branch*) in that. Click on Save.

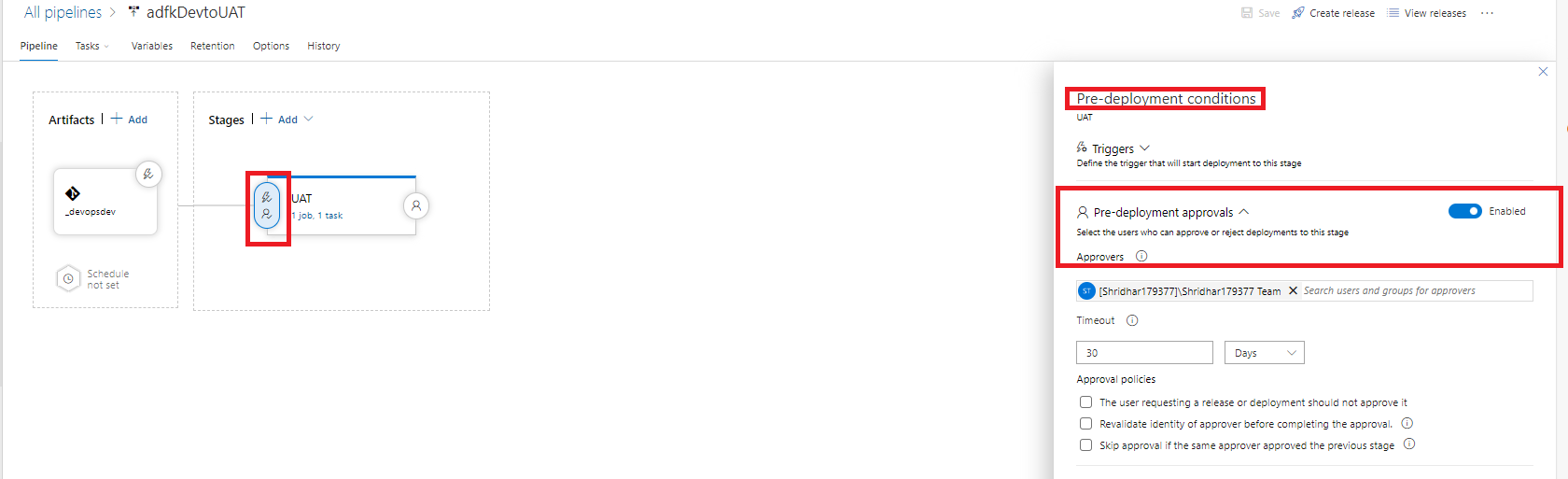
Note: In Azure DevOps, Continuous Deployment is a process that automatically deploys code changes to a target environment. You can use Continuous Deployment triggers to create a release every time a new build artifact is available.

Target branch filters can be used to trigger deployment for a specific target branch.

For example, you can set up a release pipeline to deploy from multiple branches using Azure Pipelines. Artifact filters can be applied to a specific branch to enable deployment to a specific stage when all the conditions are met.

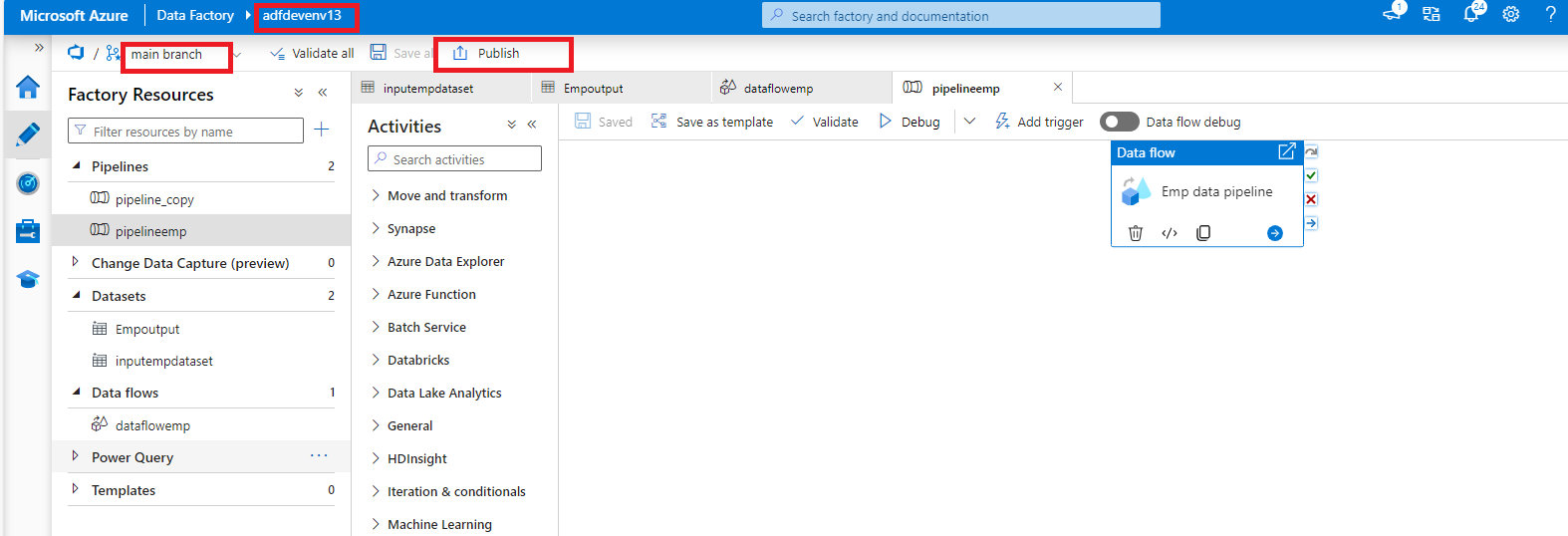


Once the Artifacts section is configured, we need to configure the Stage section’s Pre-deployment condition as shown below.

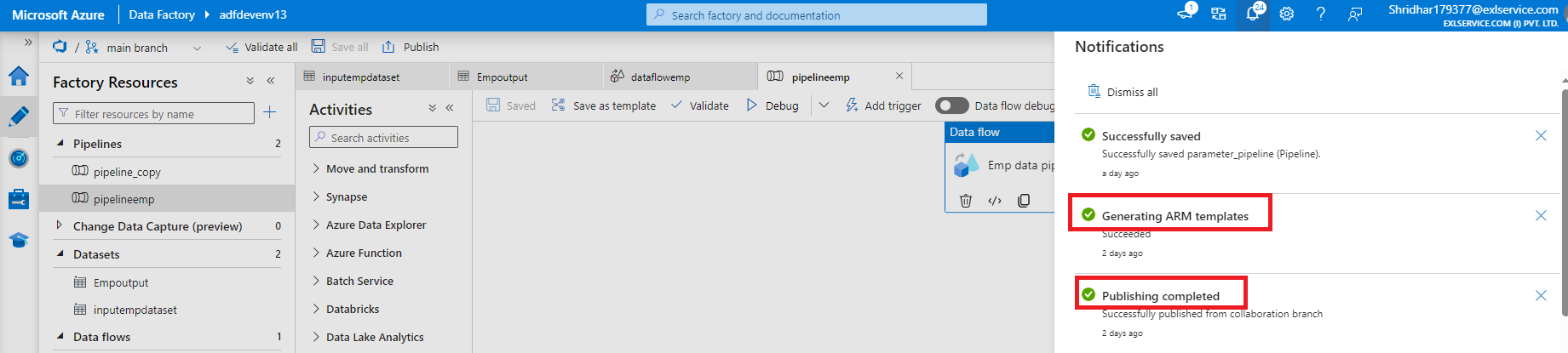


Click on save. And don’t click on create release option.

**Step 9:** Once all the steps are done in the Azure Devops interface for CICD, go to the Azure Dev ADF -> main branch -> click on the Publish button.



After Publishing the Dev ADF resources, the ADF starts publishing and also generates the ARM Templates to the adf\_publish branch.



In Devops Interface we need to approve the publish approval, so that deployment will be done in Target (UAT or PROD Environments) environment.

Release deployment window:

