# Continuous Integration and Continuous Delivery with Azure Data Factory and Azure DevOps

**Azure Labs by Roque Daudt (rdaudt@yahoo.com)**

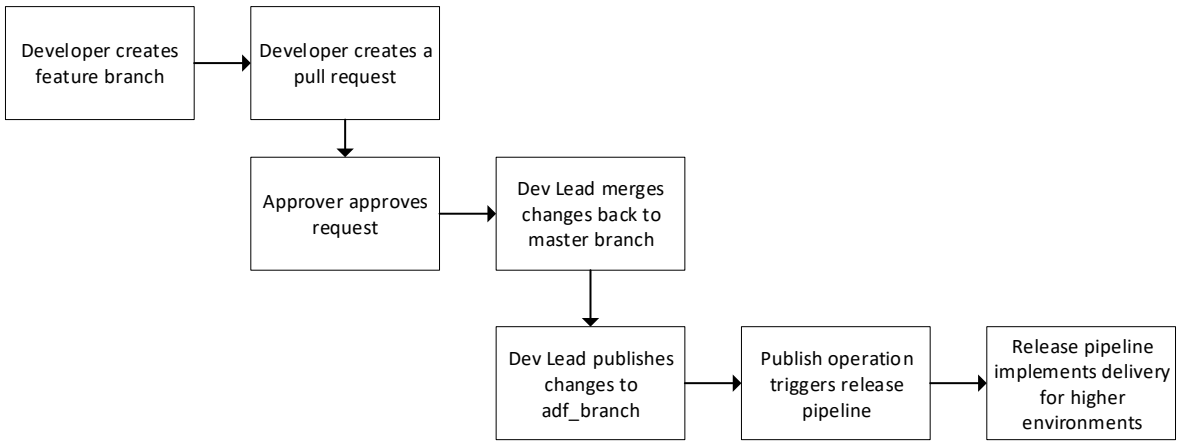
## 04 – Set Up Code Repository for ADF Dev

In this lab we will connect ADF Dev to Azure DevOps. Follow the steps below in sequence.

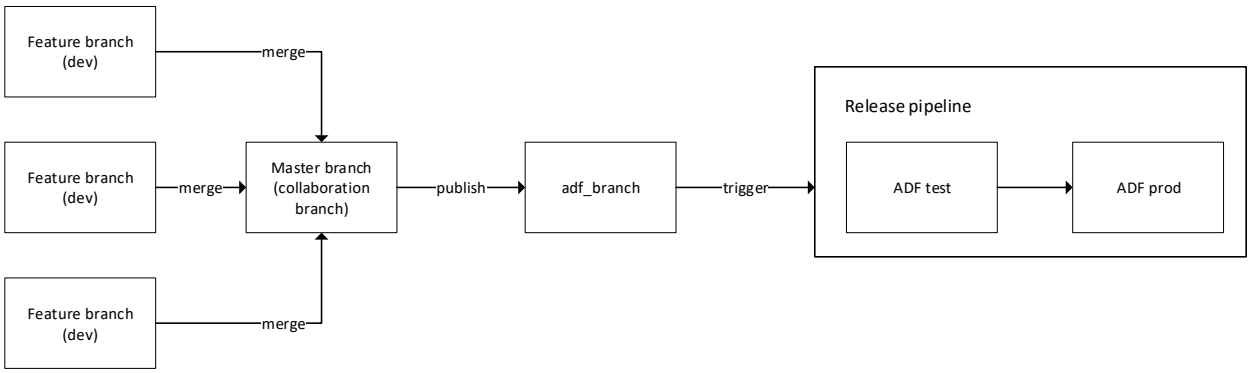
### Understand CI/CD with ADF and Azure DevOps

There is not only one way to setup CI/CD with ADF and Azure DevOps. Microsoft's "official" process is the one that we will explore first but later we will review other approaches.

The simplified version of Microsoft's approach that we will see in these labs implement a workflow like this.




Branch and operations-wise, this is how artifacts move from one place to another in our demo.

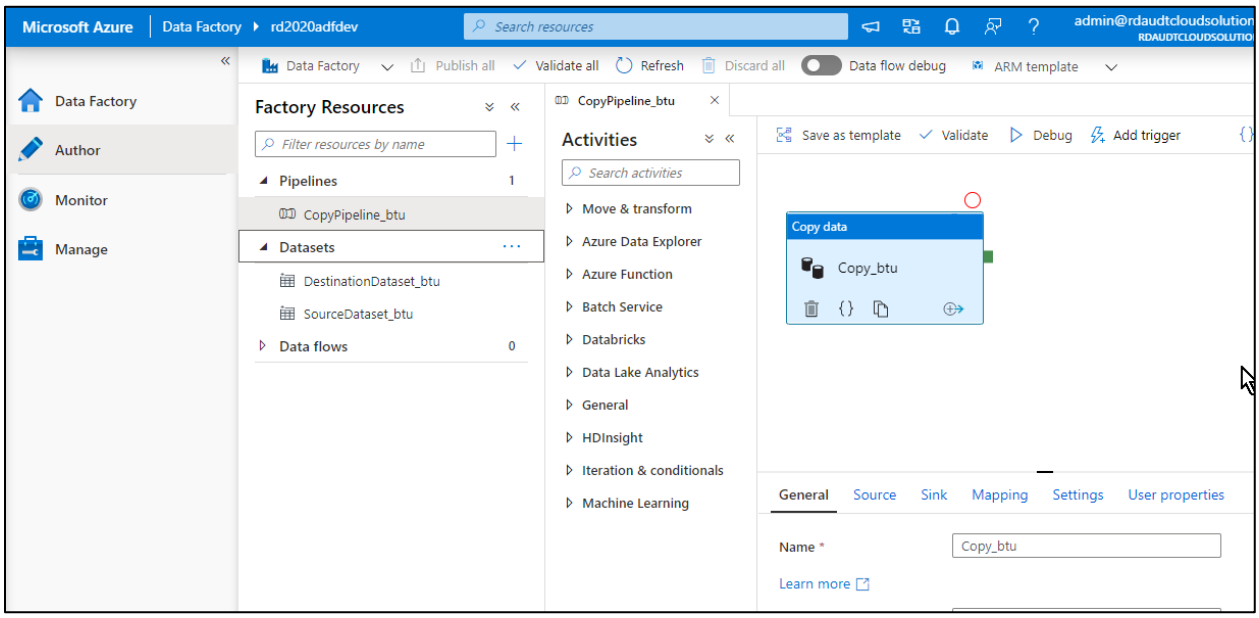


It will all make much more sense as we run the labs. We will reference these images several times.

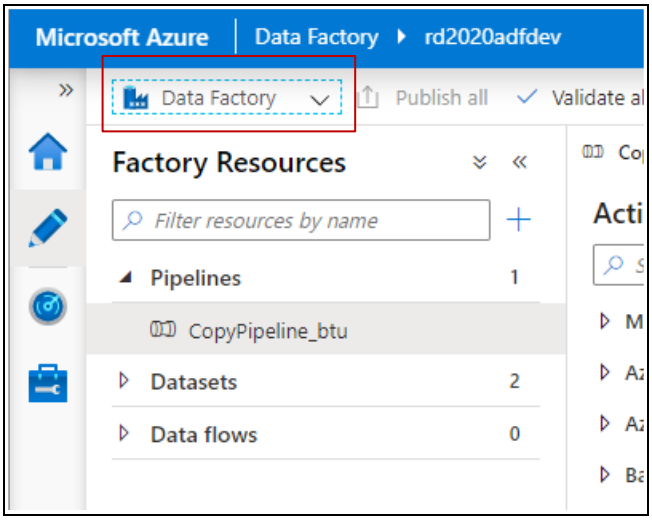
## Set up code repository for ADF Dev

At the end of the previous lab we were exploring ADF dev. If you closed it, please get there again.

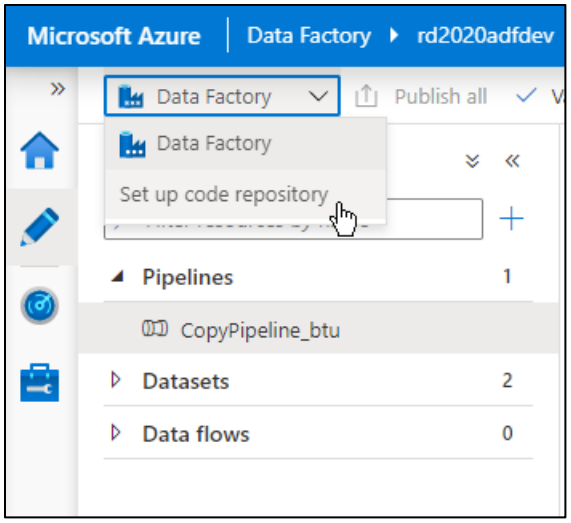
The image below shows the only pipeline in ADF dev.



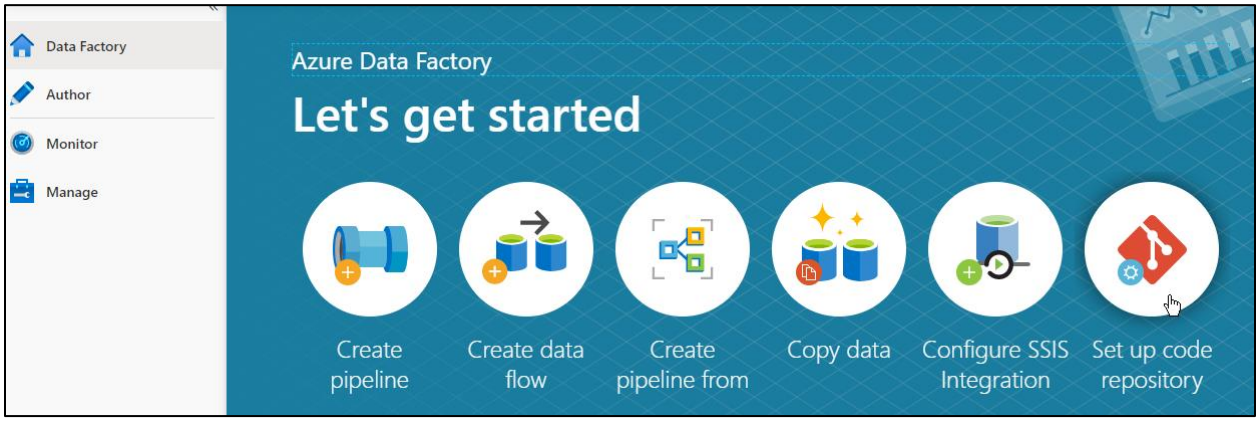

There are a few other things to see here. One of them is that ADF Dev is in “Data Factory” mode, as indicated by the highlighted area below.




The other mode is **Azure DevOps GIT**. This becomes ADF’s mode once we set up code repository for ADF. This process can be started by selecting **Set up code repository** in the drop-down list.



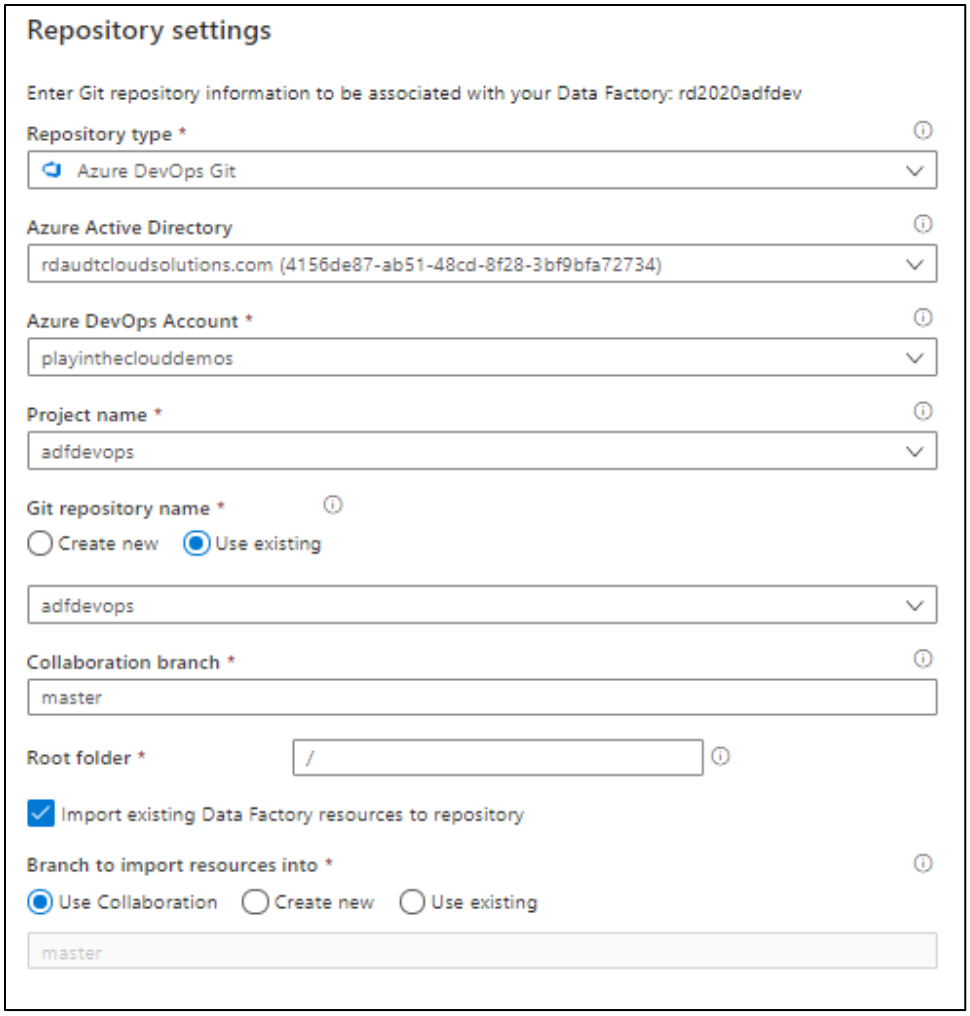

Note: it can also be started from ADF's home page.




Choose any of the two options. With one or the other, we will be taken to the panel below where we can configure the repository. This will open a panel where we will configure the repository.

It starts by a choice between **Azure DevOps Git** and **GitHub**. In these labs we are exploring **Azure DevOps Git**.


Once you enter it, the panel expands.

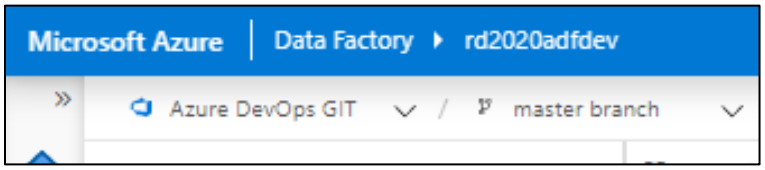



Notes about the form items above:

| Field | Note |
|---|---|
| **Repository type** | Make sure to select **Azure DevOps Git** |
| **Azure Active Directory** | The one associated to your subscription |
| **Azure DevOps account** | Select the organization that you have previously created |
| **Project name** | The Azure DevOps project that was previously created |
| **Git repository name** | For simplicity's sake, select **Use existing** and select the one with the same name as the project |
| **Collaboration branch** | ADF developers will work on feature branches of the repository. Then, once work is ready, they will check it back into the **Collaboration branch**. Traditionally, this is the master branch. |

Make sure also to check **Import existing Data Factory resources to repository** and select **Use Collaboration** (the master branch). Click **Apply.**

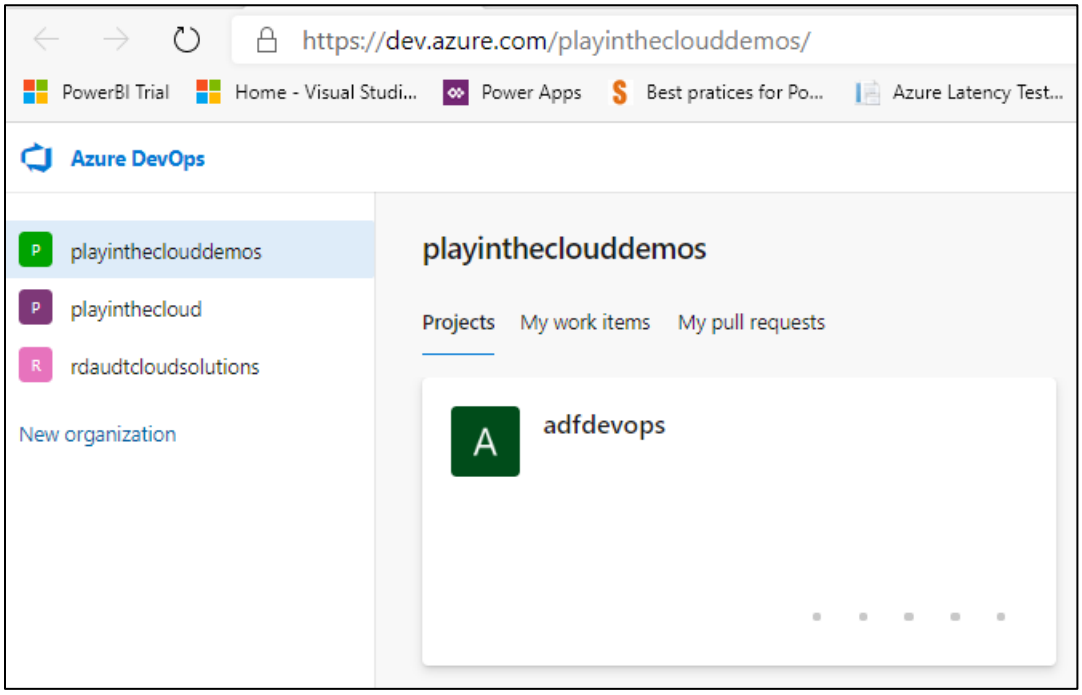
Once setup is done the panel is closed and you are taken back to ADF. What you see now is that the ADF is in **Azure DevOps GIT** mode. You also see that ADF is connected to the master branch.



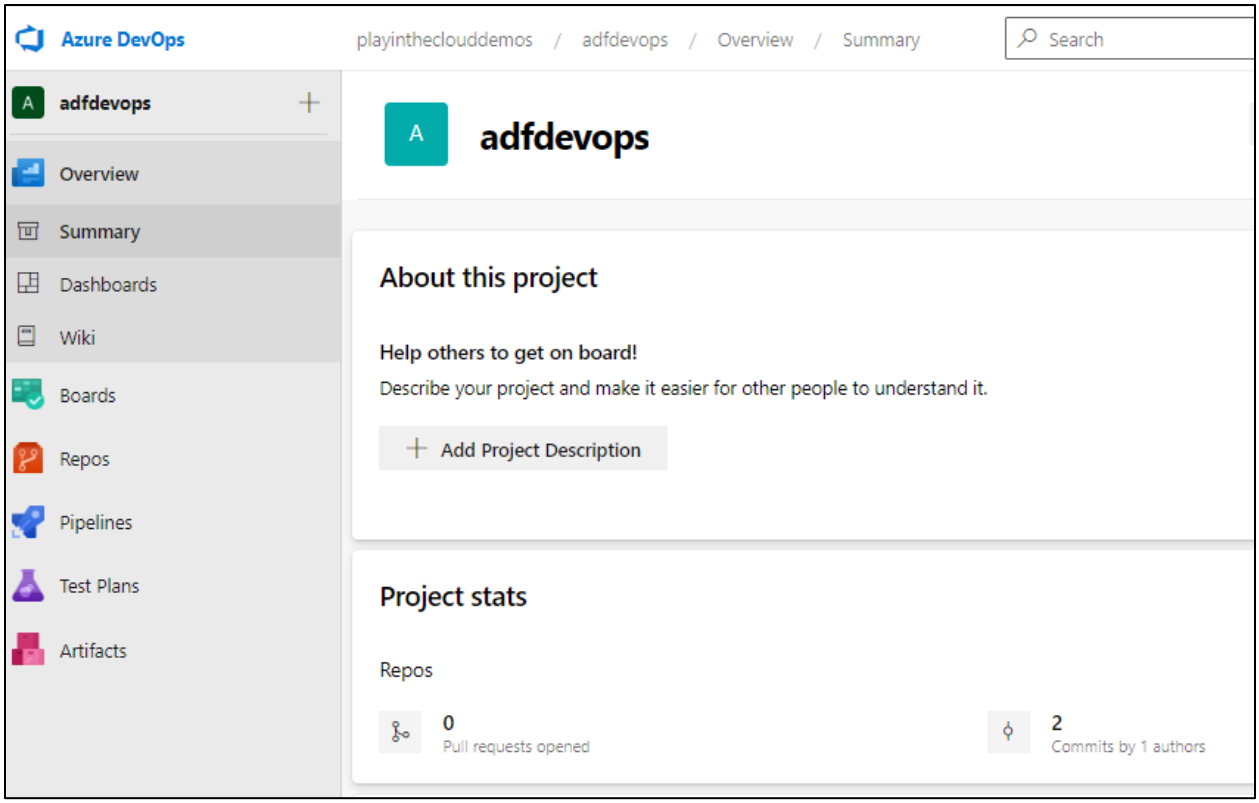

## Explore Azure DevOps

Now that the repo has been set up, we should go back to Azure DevOps and explore it a bit more.

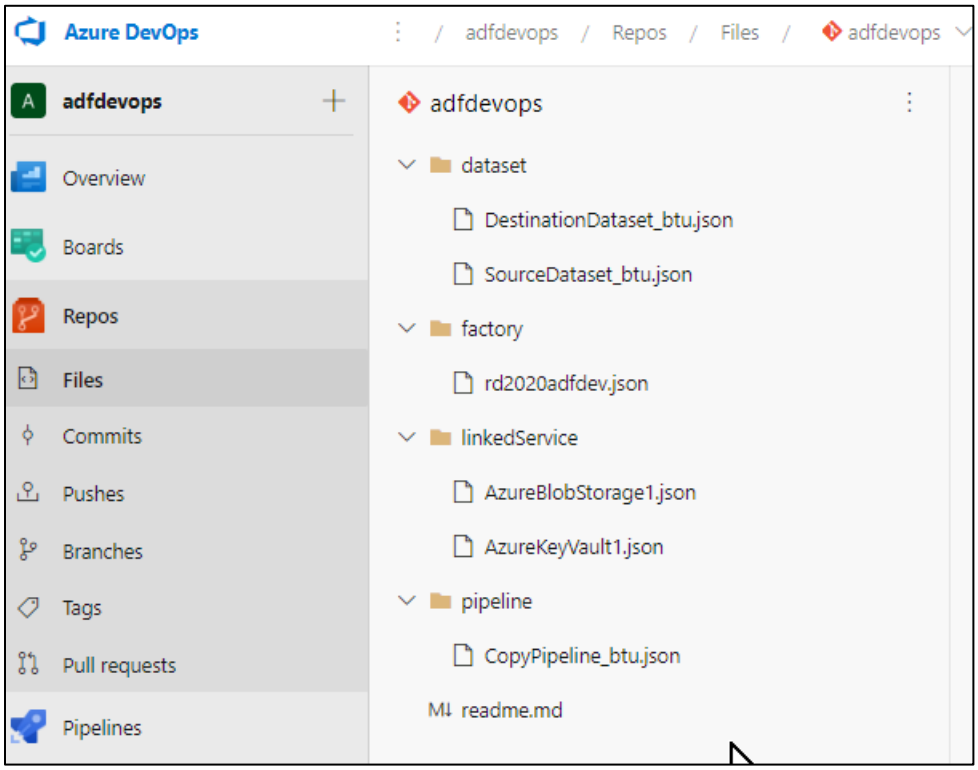
In another tab of your browser, navigate to https://dev.azure.com/. Make sure that the correct organization is selected on the left.



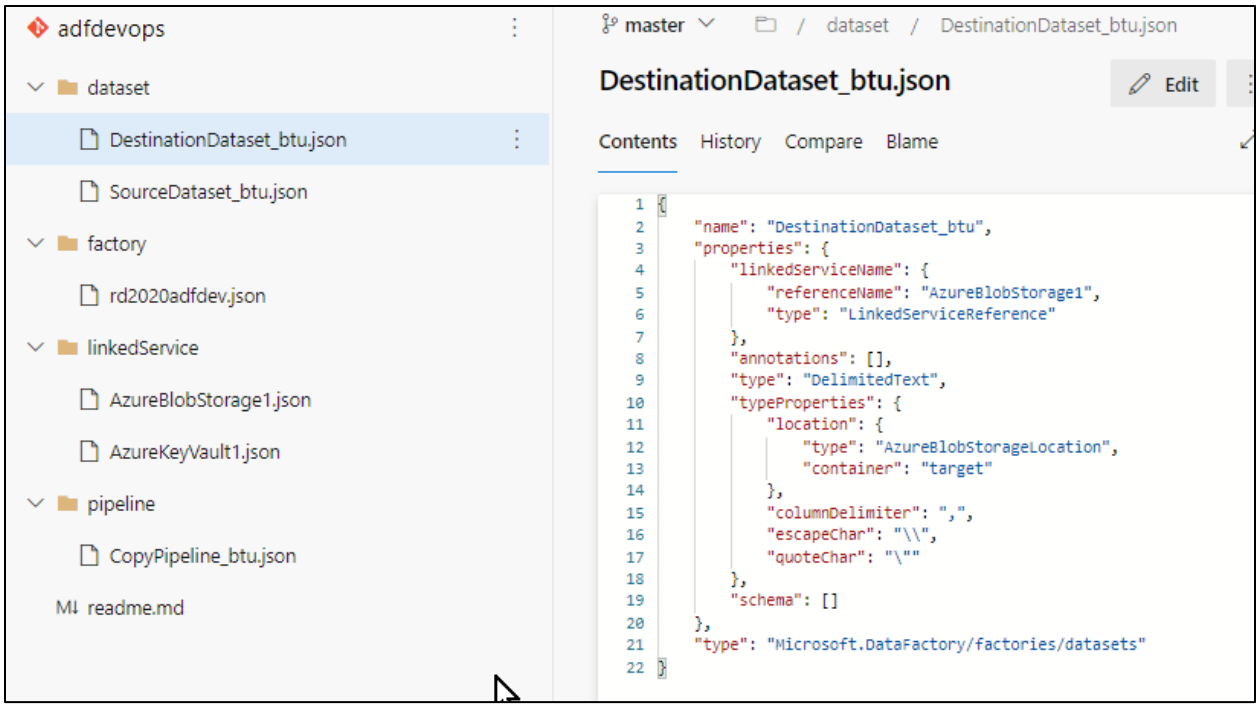

Click on the project (**adfdevops**, in my case).



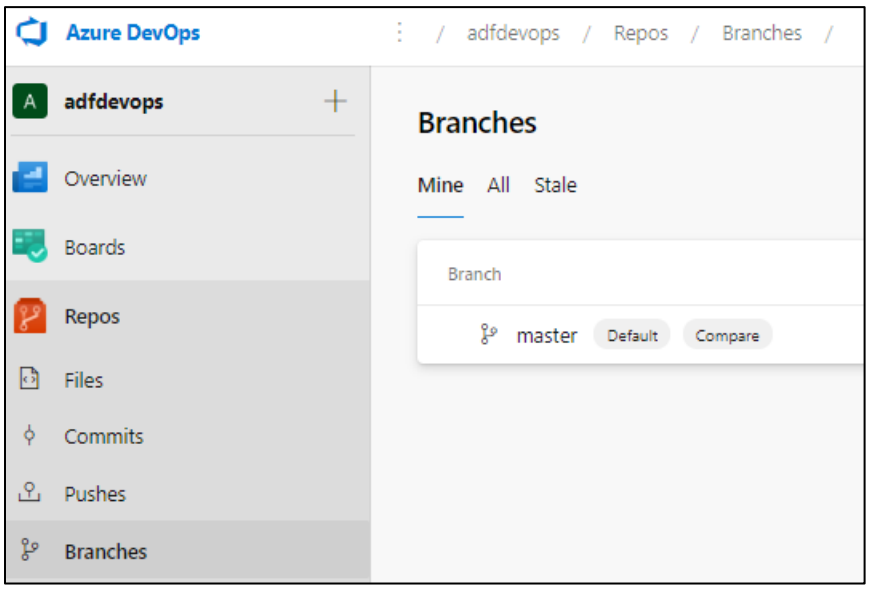

Click on **Repos / Files**. Review and expand the folders. You will see json files for the artifacts that existed in ADF dev when we set up the code repository.



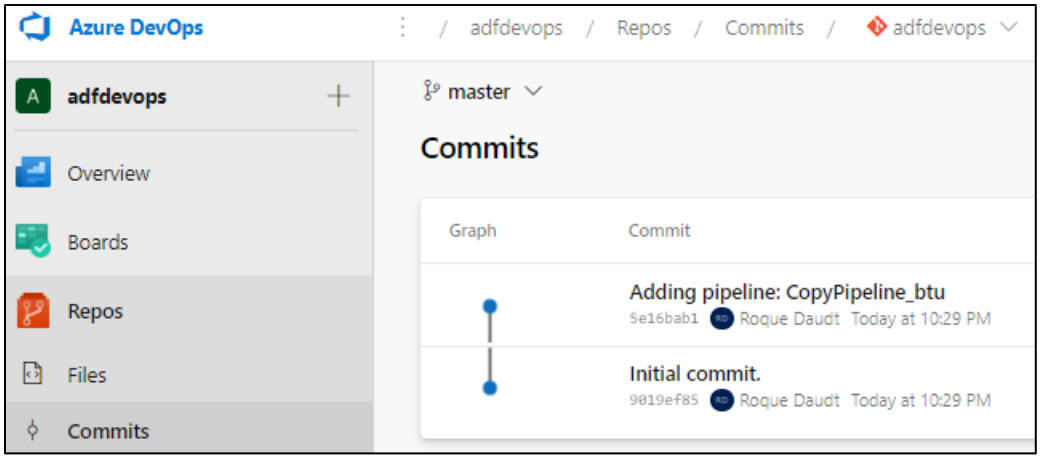
Click on any of the files to see their content.




Click **Branches** to verify that only one branch exists so far, the collaboration branch, master.




Click **Commits** to see the initial push.




## Next

Next, we will learn how to create feature branches, merge them back to the collaboration branch and publish them to the **adf_publish** branch. Go to **05 – Continuous Integration.**