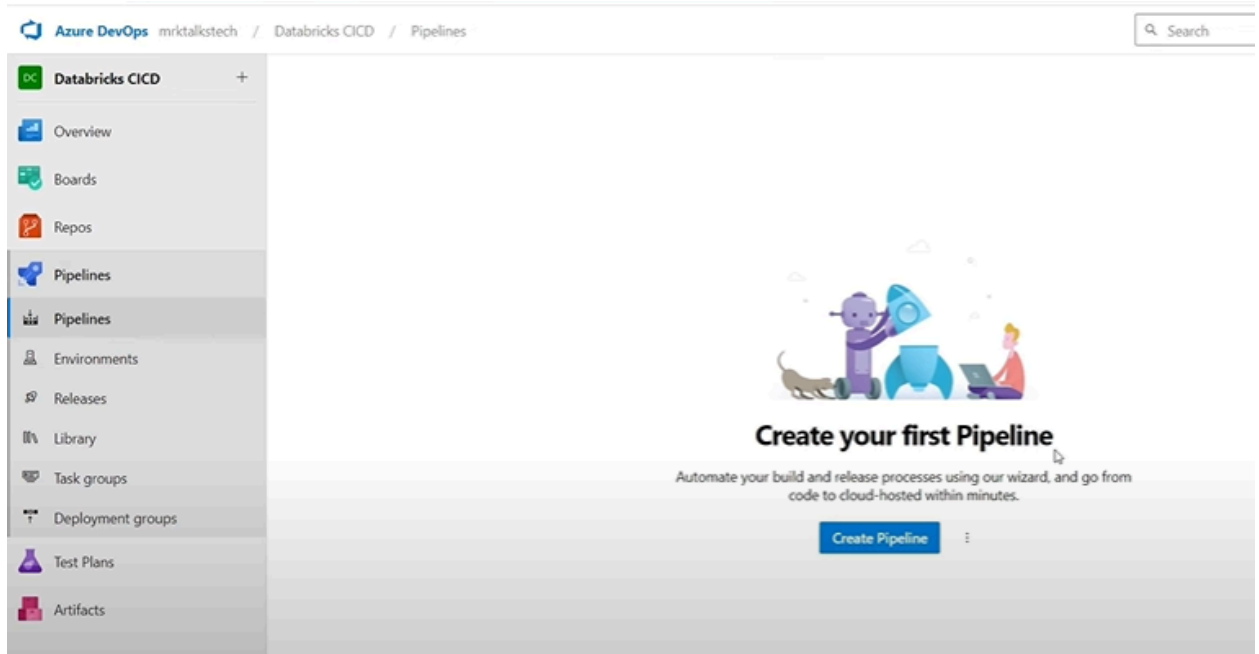
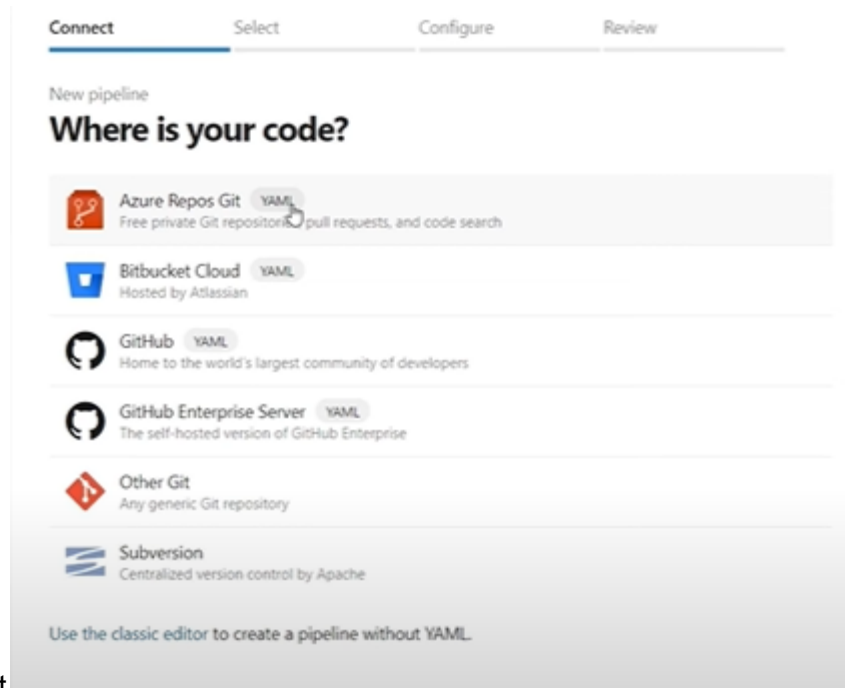


Here we can see all the CICD which we created is in the main branch

Creating Pipeline in Azure DevOps

1. Lets build our first pipeline ...go to pipelines in Azure devops





2. Select azure repos git



next we'll

✓ Connect


✓ Select


Configure


Review


New pipeline

Configure your pipeline

 **Python package**
Create and test a Python package on multiple Python versions.

 **Python to Linux Web App on Azure**
Build your Python project and deploy it to Azure as a Linux Web App.

 **Starter pipeline**
Start with a minimal pipeline that you can customize to build and deploy your code.

 **Existing Azure Pipelines YAML file**
Select an Azure Pipelines YAML file in any branch of the repository.

Show more


then

Select an existing YAML file

×

Select an Azure Pipelines YAML file in any branch of the repository.

Branch

 main


▼

Path

/CI/CD/cicd-pipelines.yml

▼

Select a file from the dropdown or type in the path to your file

[Databricks CI/CD Tutorial](#) 

we'll choose the existing YAML file

3. Lets review our code..before creating the pipeline

New pipeline

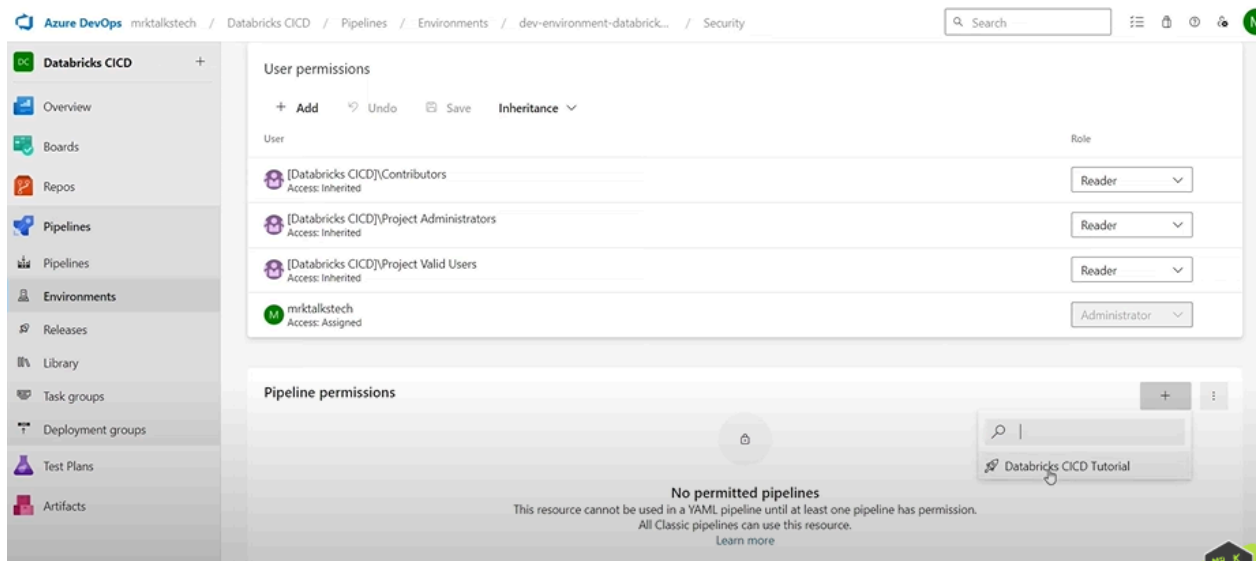
Review your pipeline YAML

Databricks CICD Tutorial / CICD/cicd-pipelines.yml

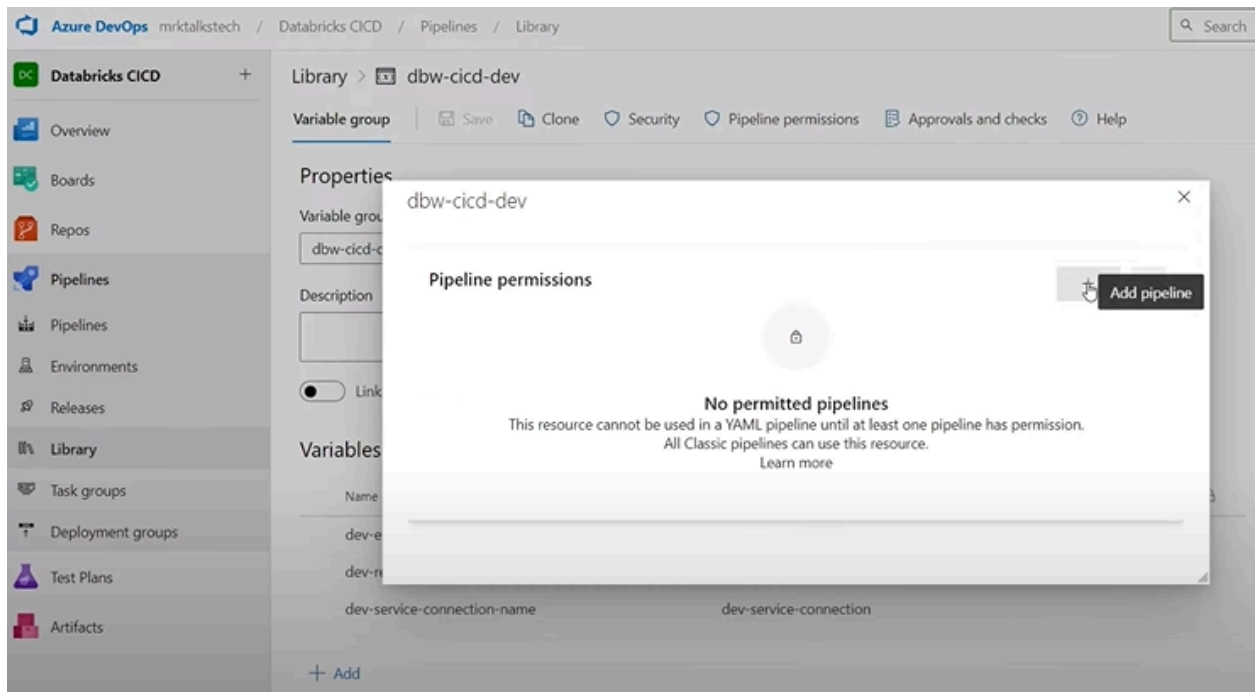
```
1 trigger:
2   -- main
3
4 variables:
5   -- group: dbw-cicd-dev
6   --
7   -- name: vmImageName
8   -- value: "windows-latest"
9   -- name: notebooksPath
10  -- value: "notebook"
11
12 pool:
13   vmImage: $(vmImageName)
14
15
16 stages:
17   -- template: templates/deploy-notebooks.yml
18   -- parameters:
19     -- stageId: "Deploy_to_Dev_Environment"
20     -- env: "dev"
21     -- environmentName: $(dev-environment-name)
22     -- resourceGroupName: $(dev-resource-group-name)
23     -- serviceConnection: $(dev-service-connection-name)
24     -- notebooksPath: $(notebooksPath)
```

and click on save pipeline

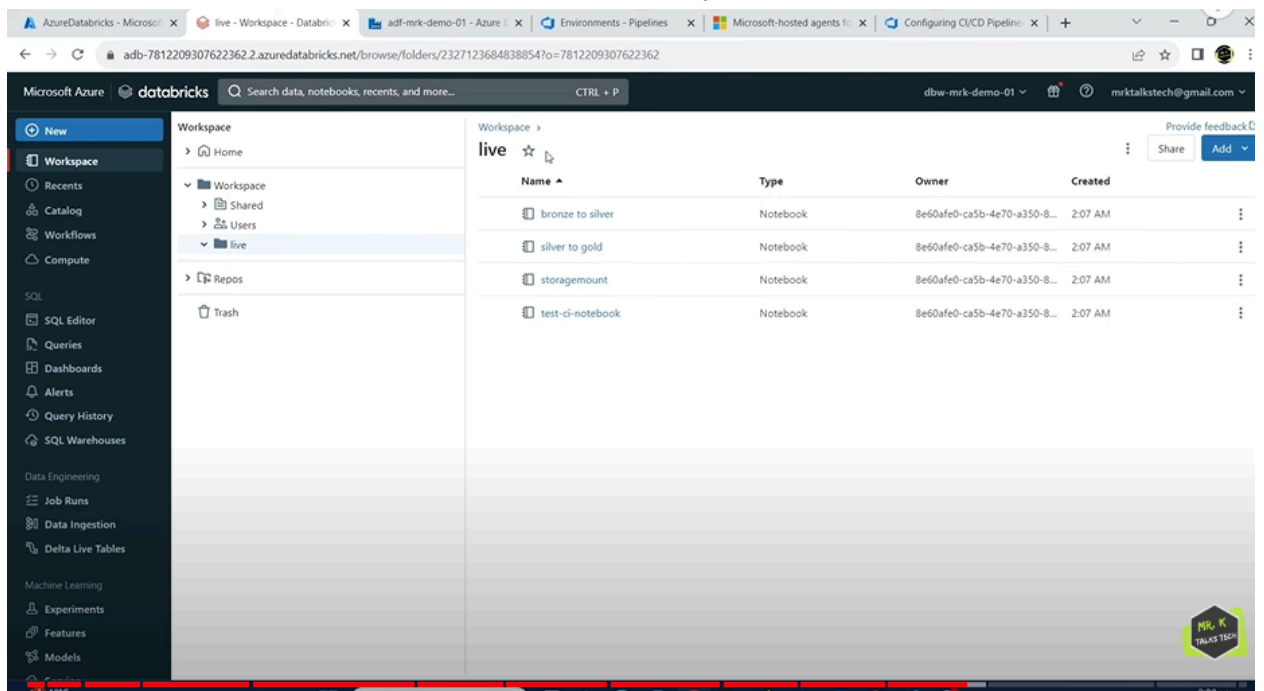
4. Next we have to give access to dev env for our pipeline..for that go to environment→ permissions
5. And we select our pipeline



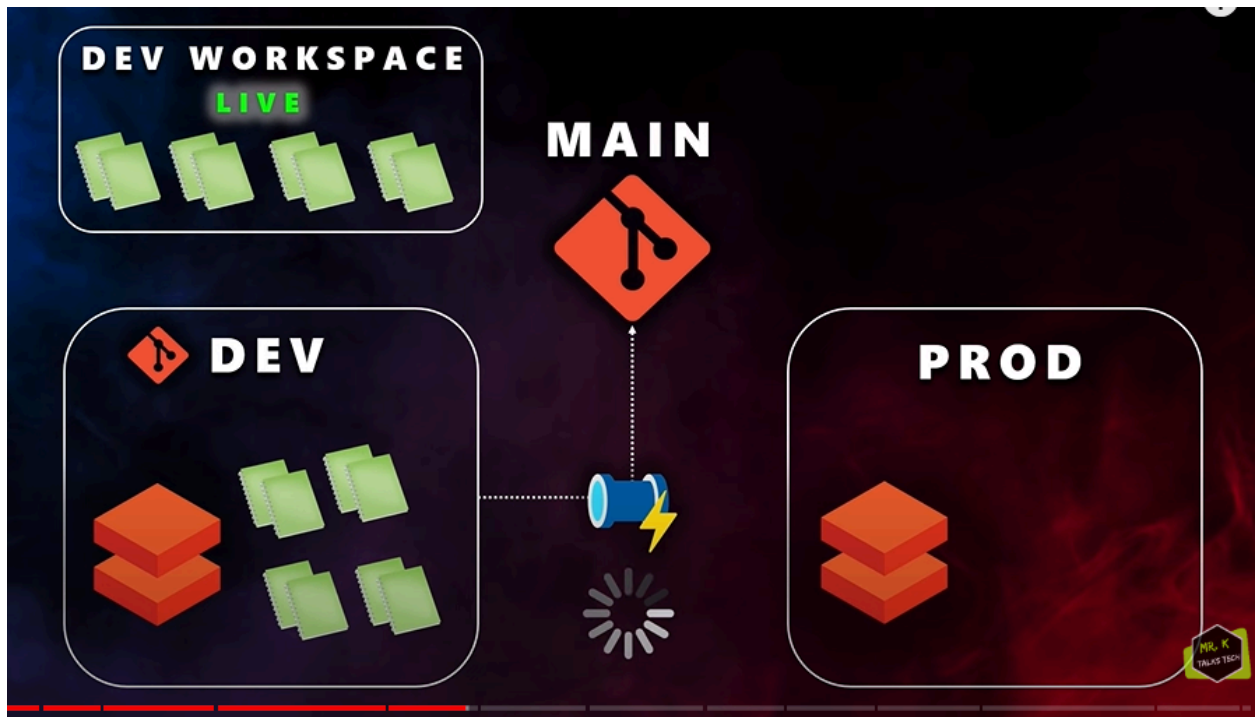
- Our pipeline must also have access to variable group as well



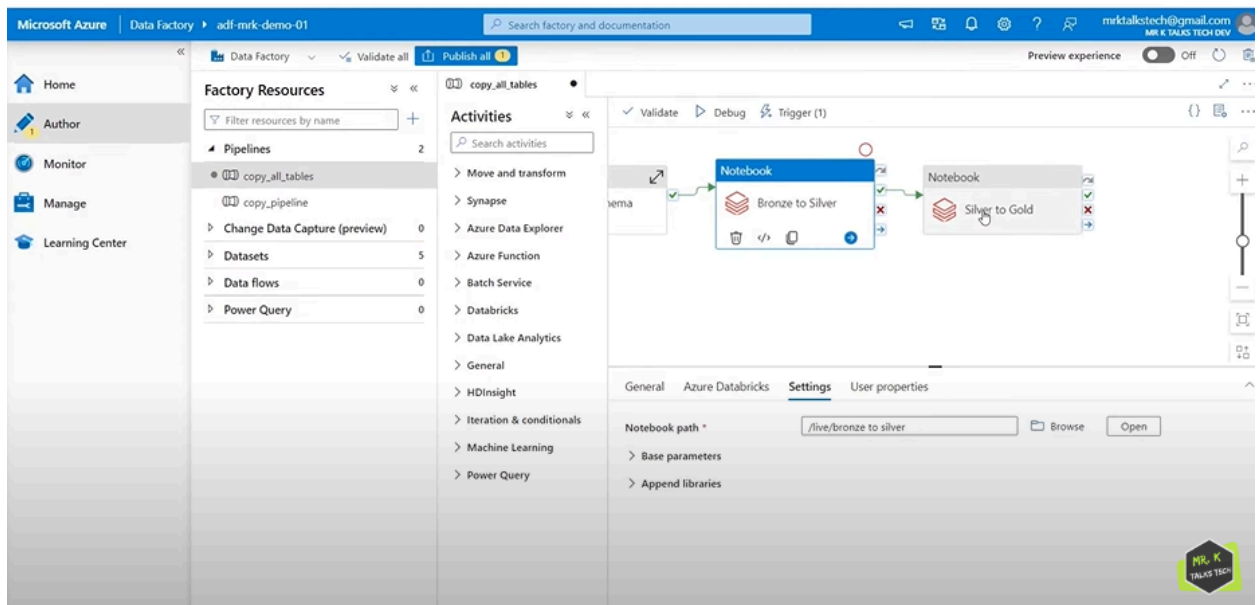
- The final permission for our pipeline ..is service connection permission
- Now our pipeline has every permission that is needed
- Next we'll test our pipeline ..by creating a branch and making a PR..then our CI must gets triggered
- Now here we can see a live folder has been created by the CI



basically we did this



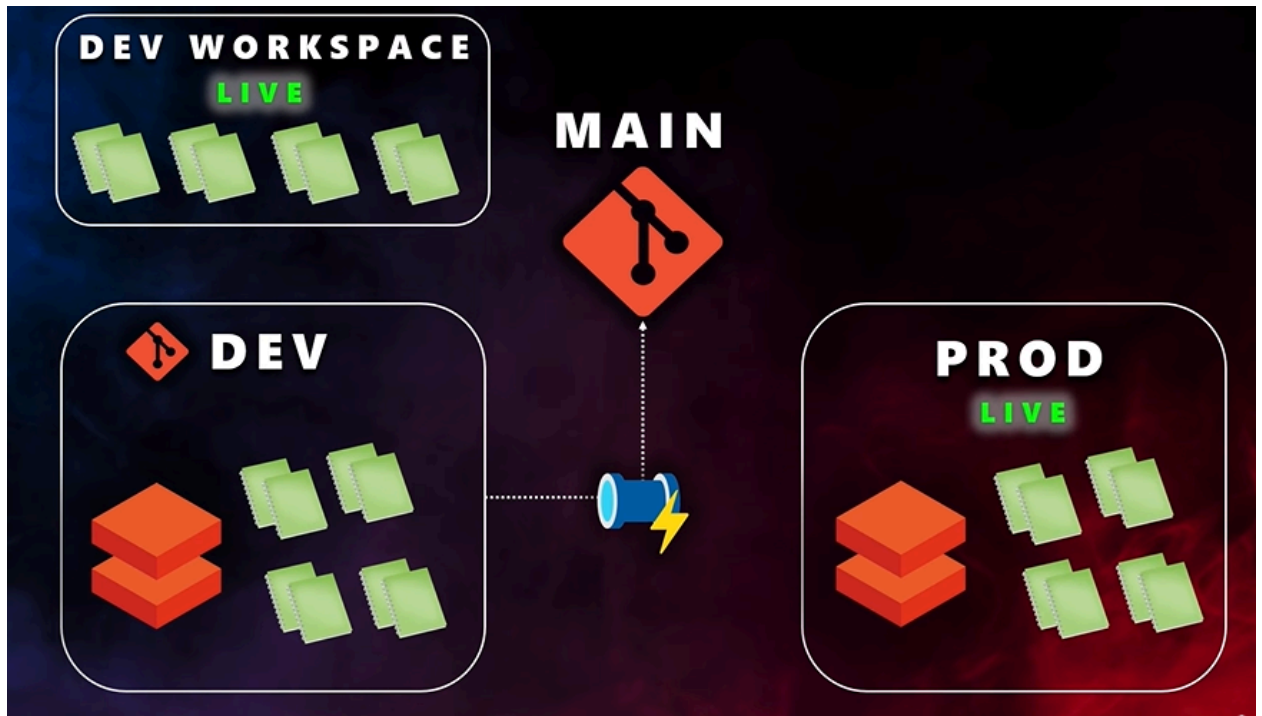
11. We can also integrate the live folder's notebooks to the ETL pipeline



which ensures our containers deal with the latest data

Continuous Deployments Pipeline

1. Here as soon as the continuous Integration gets completed...CD must get triggered and copies the notebooks from live workspace



2. We need to make few changes in our YAML code
3. Now we'll create a new feature branch from the main branch for CD ..for that we pull from the main branch and create it
4. Now the changes in the code is..we need to give prod env instead of dev env

```
variables:  
  - group: dbw-cicd-dev  
  - group: dbw-cicd-prod
```

5. We have created a new variable group for and also we create a

```
stages:  
  - template: templates/deploy-notebooks.yml  
    parameters:  
      stageId: "Deploy_to_Dev_Environment"  
      env: "dev"  
      environmentName: $(dev-environment-name)  
      resourceGroupName: $(dev-resource-group-name)  
      serviceConnection: $(dev-service-connection-name)  
      notebooksPath: $(notebooksPath)  
  
  - template: templates/deploy-notebooks.yml  
    parameters:  
      stageId: "Deploy_to_Prod_Environment"  
      env: "prod"  
      environmentName: $(prod-environment-name)  
      resourceGroupName: $(prod-resource-group-name)  
      serviceConnection: $(prod-service-connection-name)  
      notebooksPath: $(notebooksPath)
```

stage for prod

6. Next we will create the variables groups for prod as we did for dev env

The screenshot shows the Azure DevOps interface for the 'Databricks CICD' project. The left sidebar contains navigation links: Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The 'Library' link is selected. The main content area shows the 'dbw-cicd-prod' variable group configuration. The 'Properties' tab is active, showing the 'Variable group name' as 'dbw-cicd-prod' and a description field. Below the properties, there is a section for 'Variables' with a table listing three variables: 'prod-environment-name' with value 'prod-environment-databricks-cicd', 'prod-resource-group-name' with value 'rg-data-engineering-project-prod', and 'prod-service-connection-name' with value 'prod-service-connection'. The 'prod-service-connection-name' variable is highlighted with a lock icon.

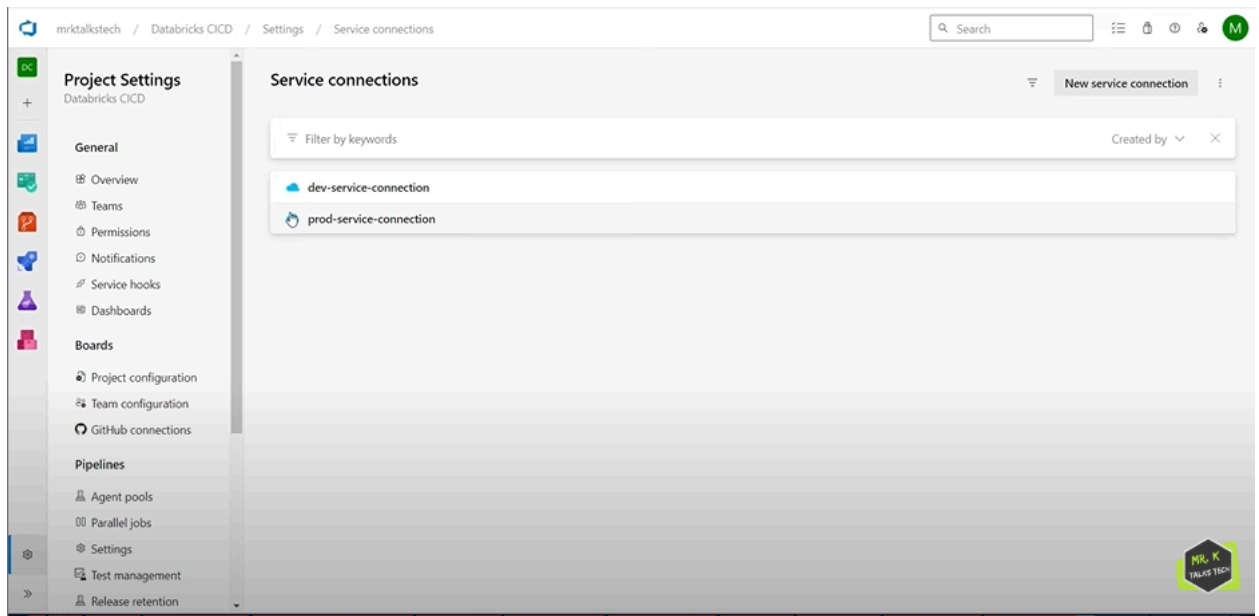
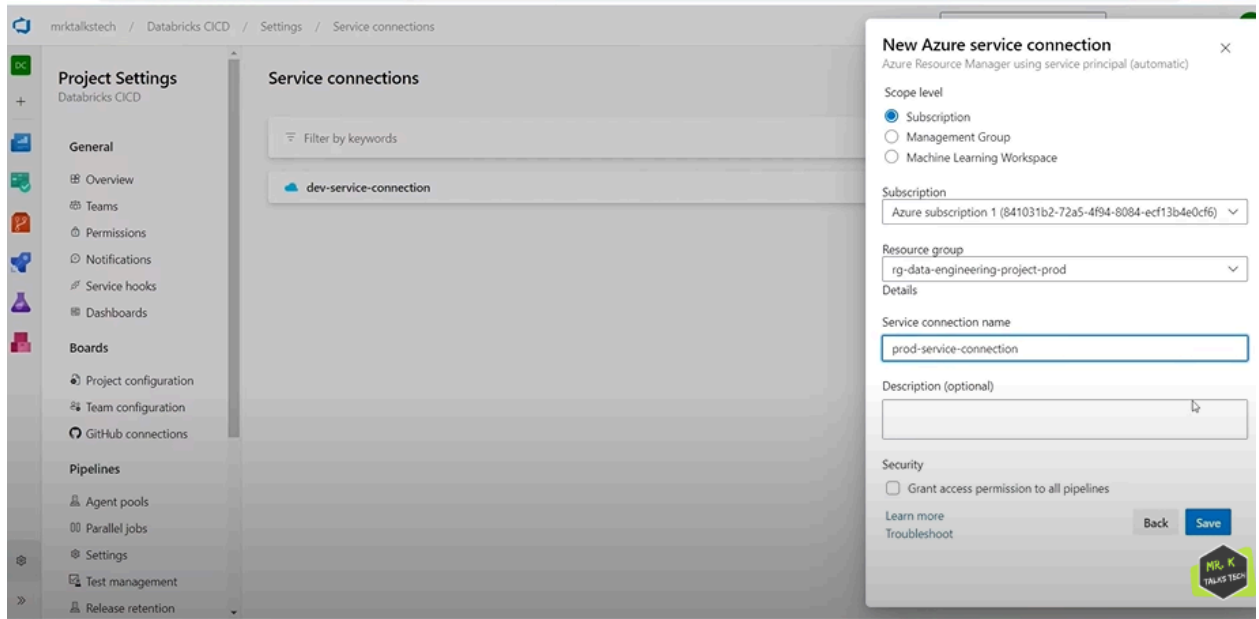
Name	Value
prod-environment-name	prod-environment-databricks-cicd
prod-resource-group-name	rg-data-engineering-project-prod
prod-service-connection-name	prod-service-connection

7. Next we will add our env for our prod

The screenshot shows the Azure DevOps interface for the 'Databricks CICD' project. The left sidebar contains navigation links: Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The 'Environments' link is selected. The main content area shows the 'Environments' page with a table listing two environments: 'dev-environment-databricks-cicd' and 'prod-environment-databricks-cicd'. The 'dev-environment-databricks-cicd' environment is shown as deployed with a green checkmark and the status '#20230911.1 on Databricks CICD Tutorial', with a last activity of '21h ago'. The 'prod-environment-databricks-cicd' environment is shown as 'Never deployed' with a last activity of 'Just now'. A 'New environment' button is visible in the top right corner.

Environment	Status	Last activity
dev-environment-databricks-cicd	✓ #20230911.1 on Databricks CICD Tutorial	21h ago
prod-environment-databricks-cicd	Never deployed	Just now

8. Next we will create a service connection for our prod env



9. Next we will give all the required permissions for our CI/CD pipeline