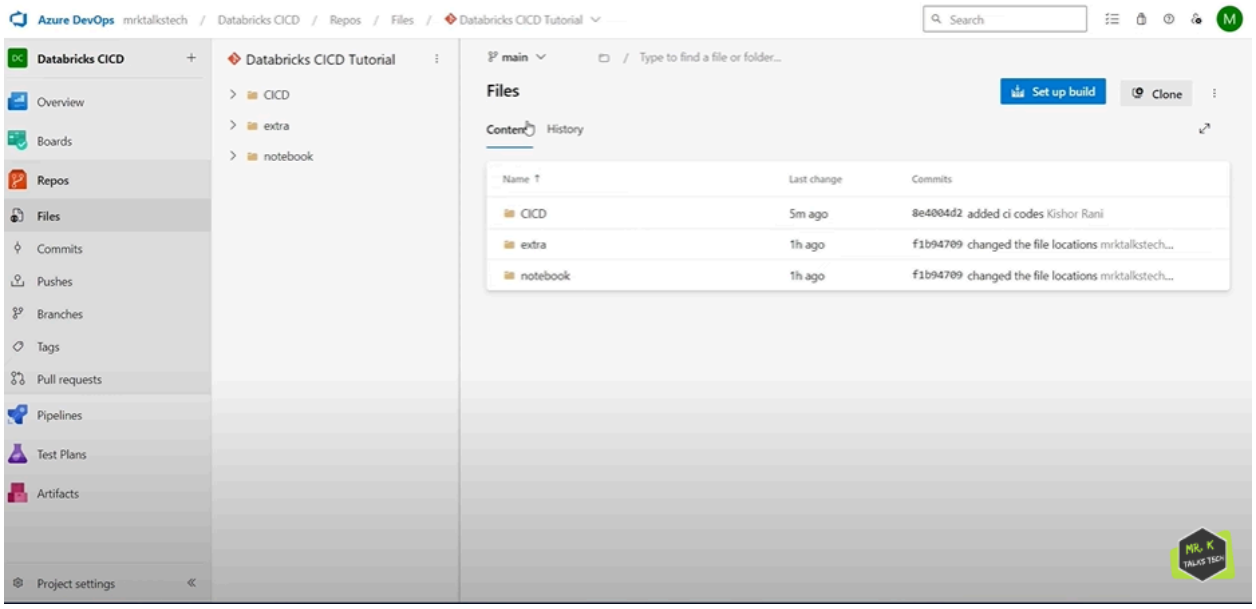
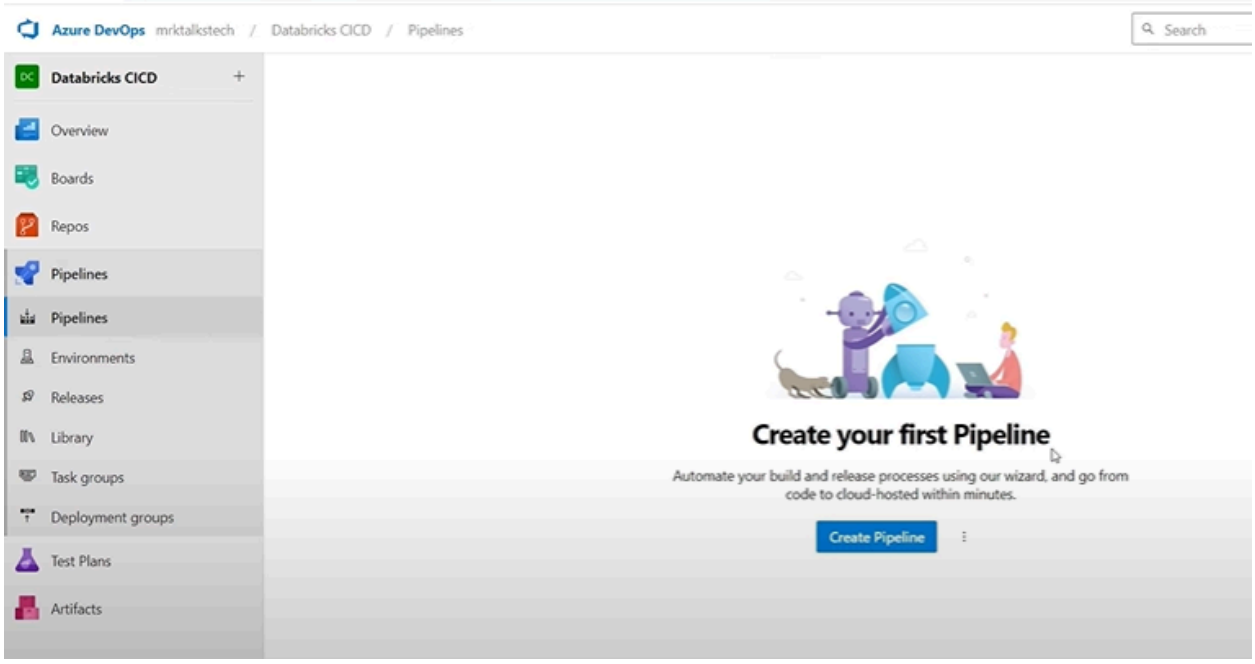
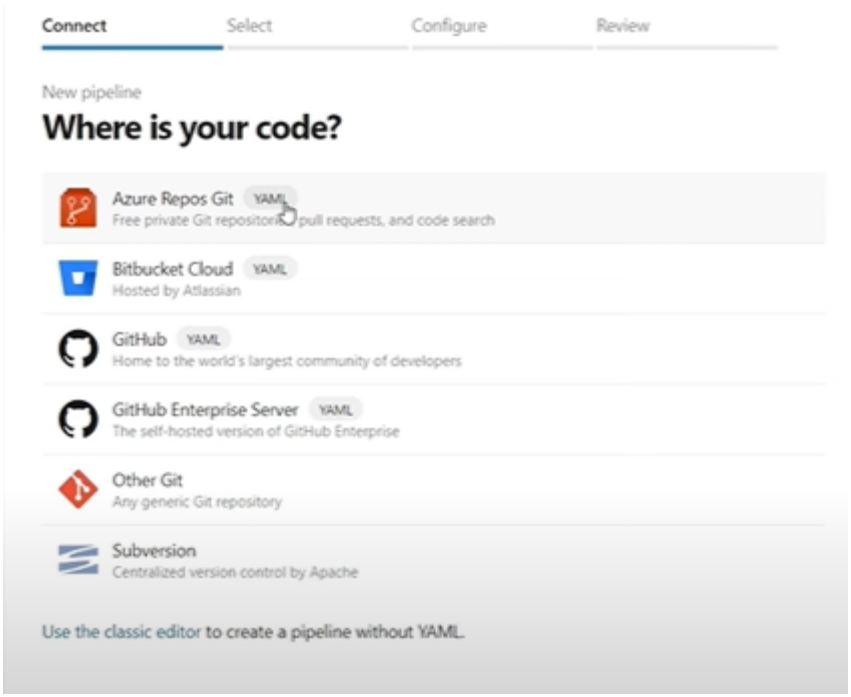


Here we can see all the CICD which we created is in the main branch

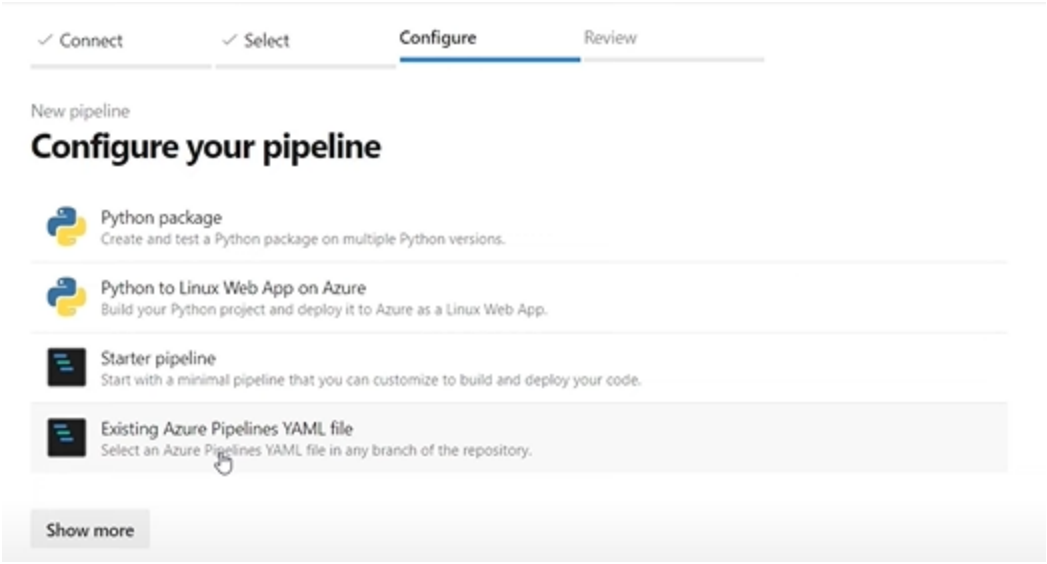
Creating Pipeline in Azure Devops

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

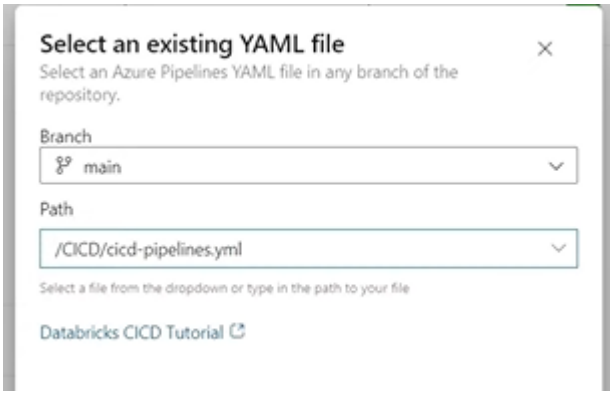


2. Select azure repos git 

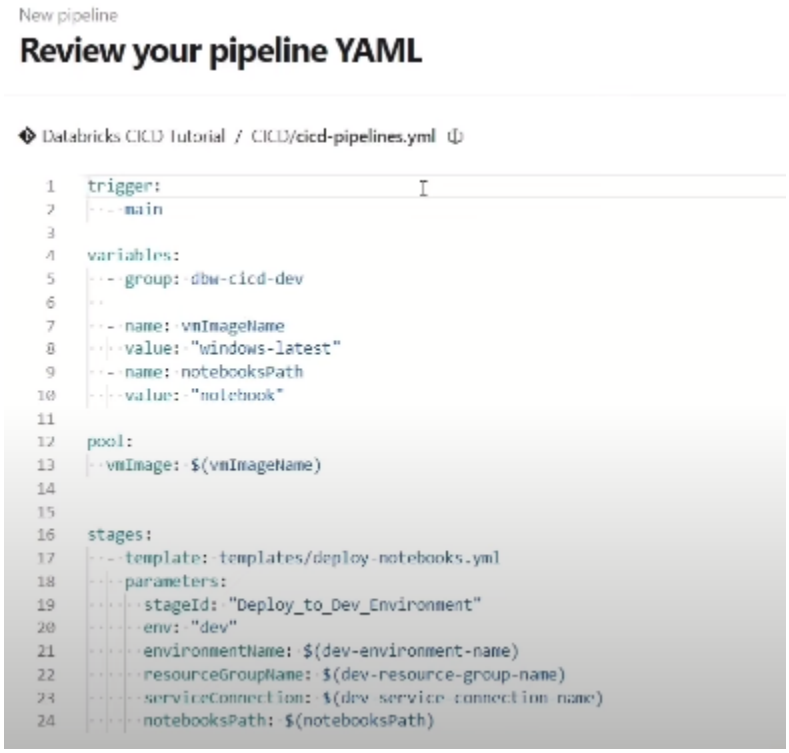
next we'll



then we'll choose the existing YAML file

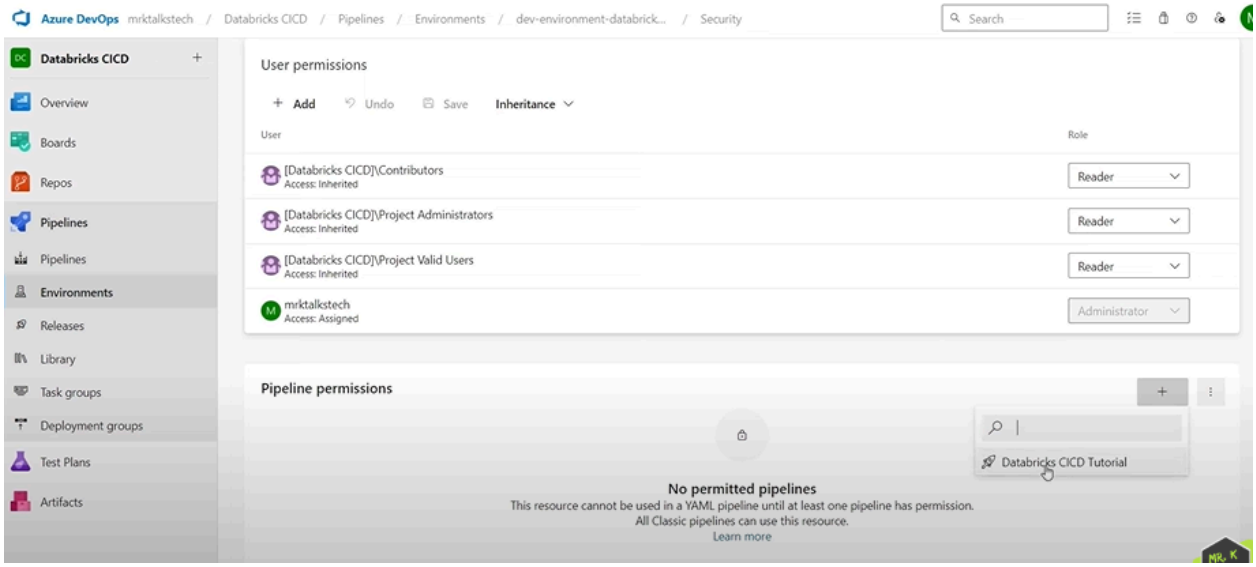


3. Lets review our code..before creating the pipeline

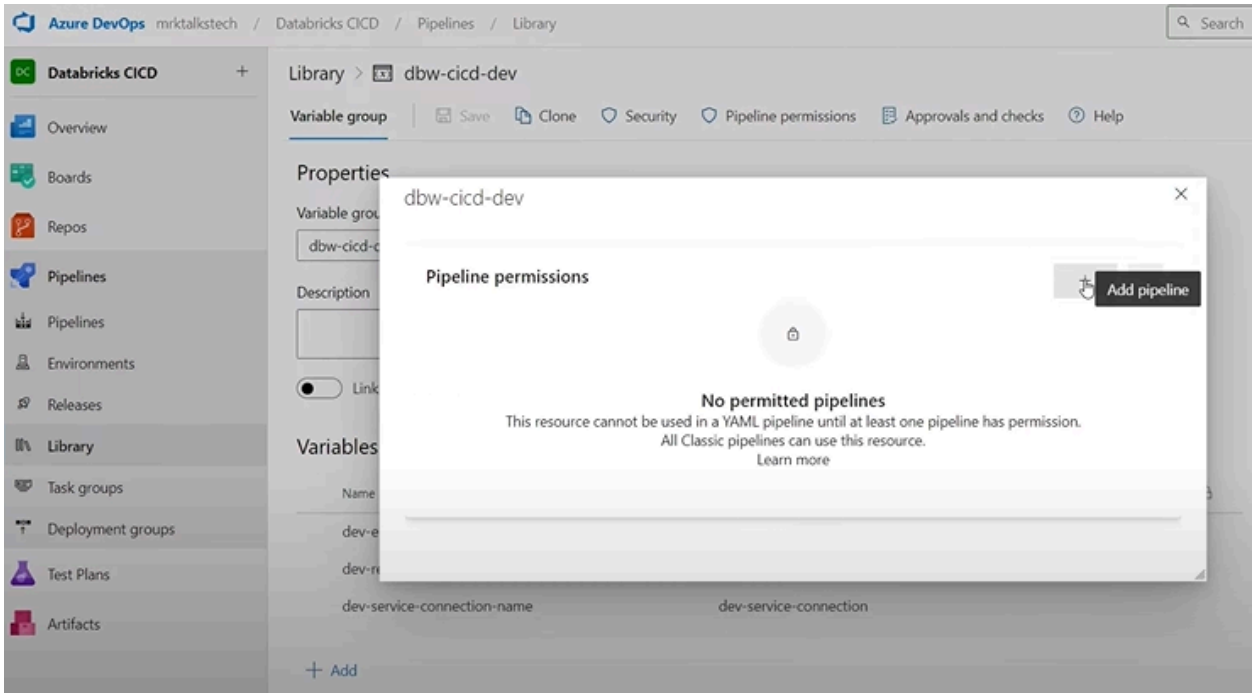


and click on save pipeline

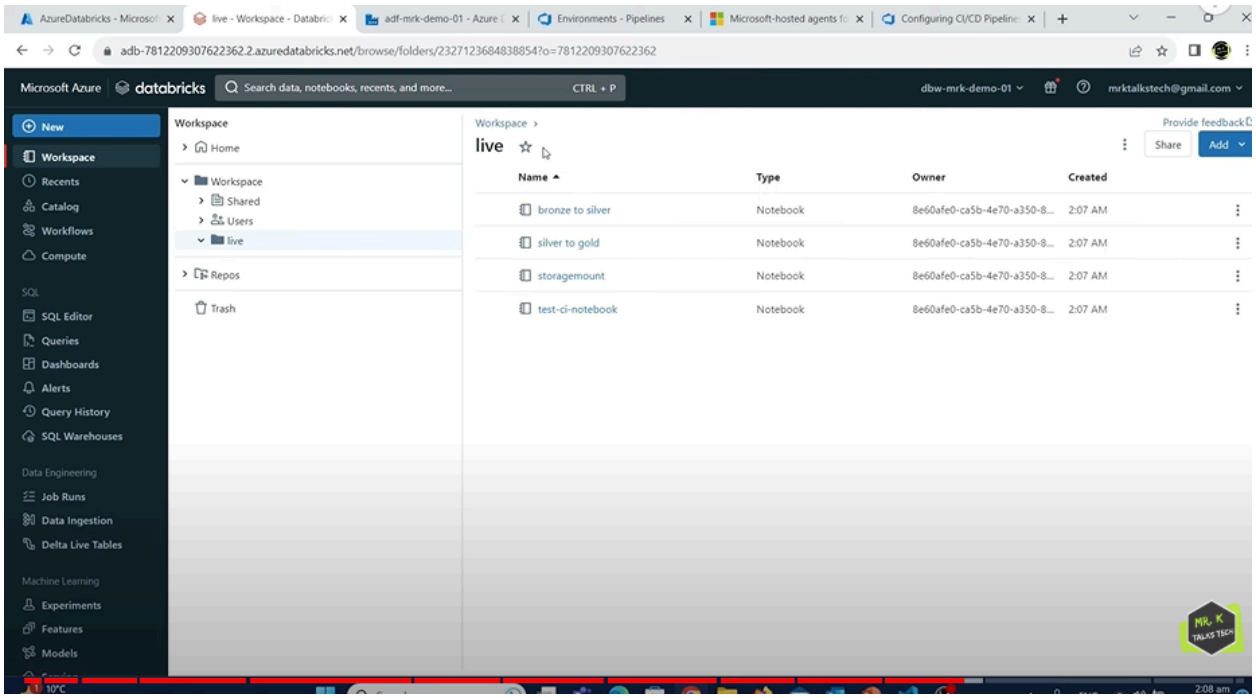
4. Next we have to give access to dev env for our pipeline..for that go to environment→ permissions
5. And we select our pipeline



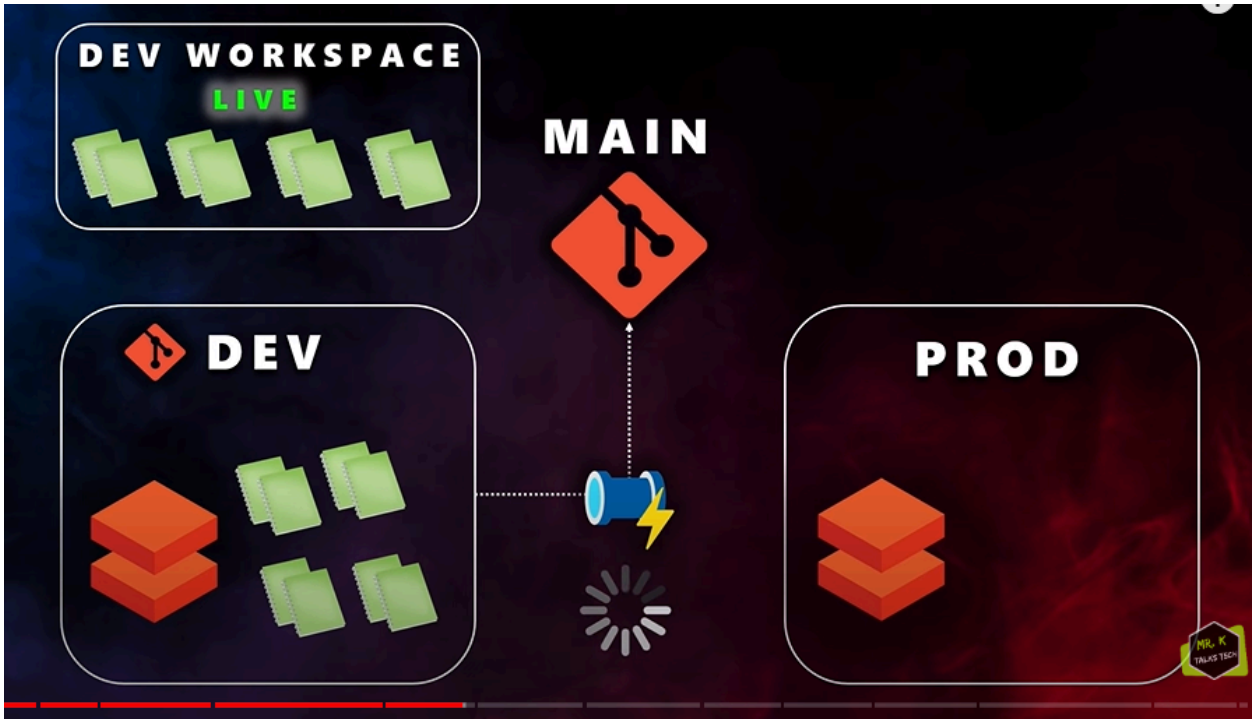
6. Our pipeline must also have access to variable group as well



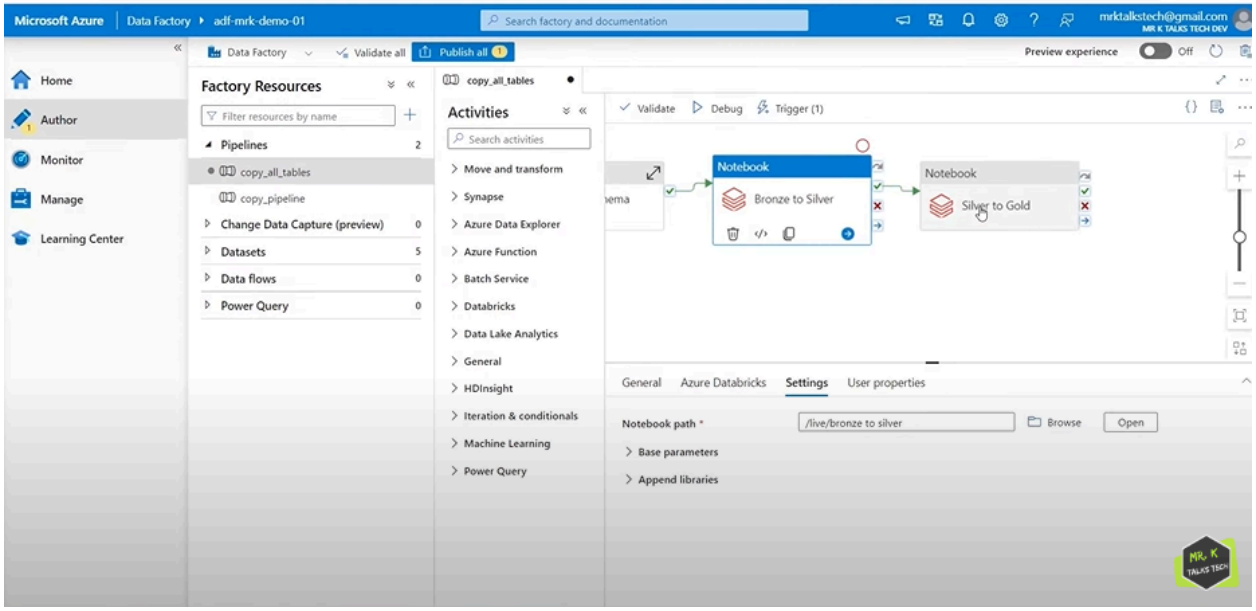
7. The final permission for our pipeline ..is service connection permission
8. Now our pipeline has every permission that is needed
9. Next we'll test our pipeline ..by creating a branch and making a PR..then our CI must gets triggered
10. Now here we can see a live folder has been created by the CI



basically we did this



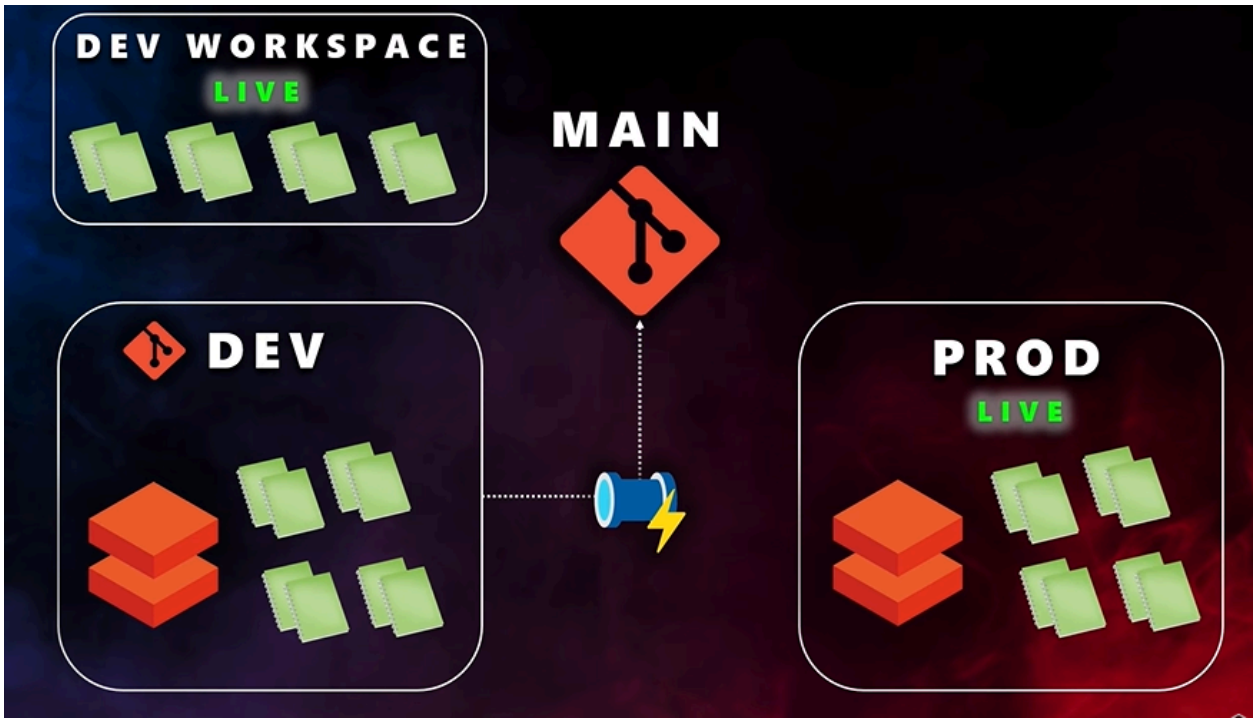
11. We can also integrate the live folder's notebooks to the ETL pipeline



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
5. We have created a new variable group for

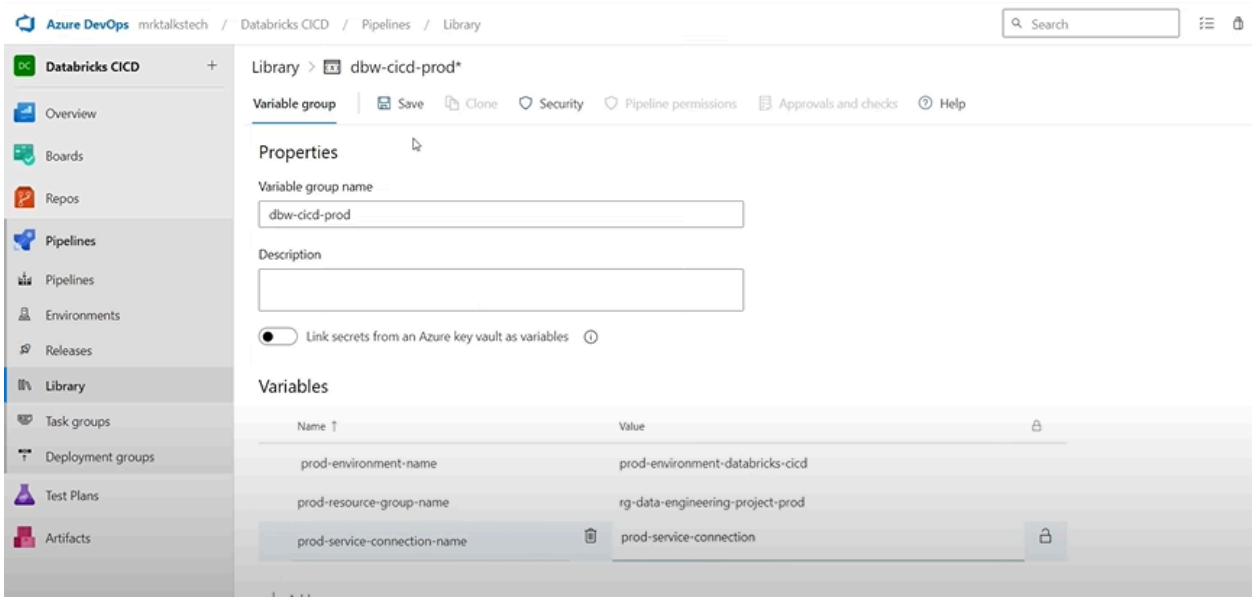
```yaml
variables:
  - group: dbw-cicd-dev
  - group: dbw-cicd-prod
```

and also we create a stage for prod

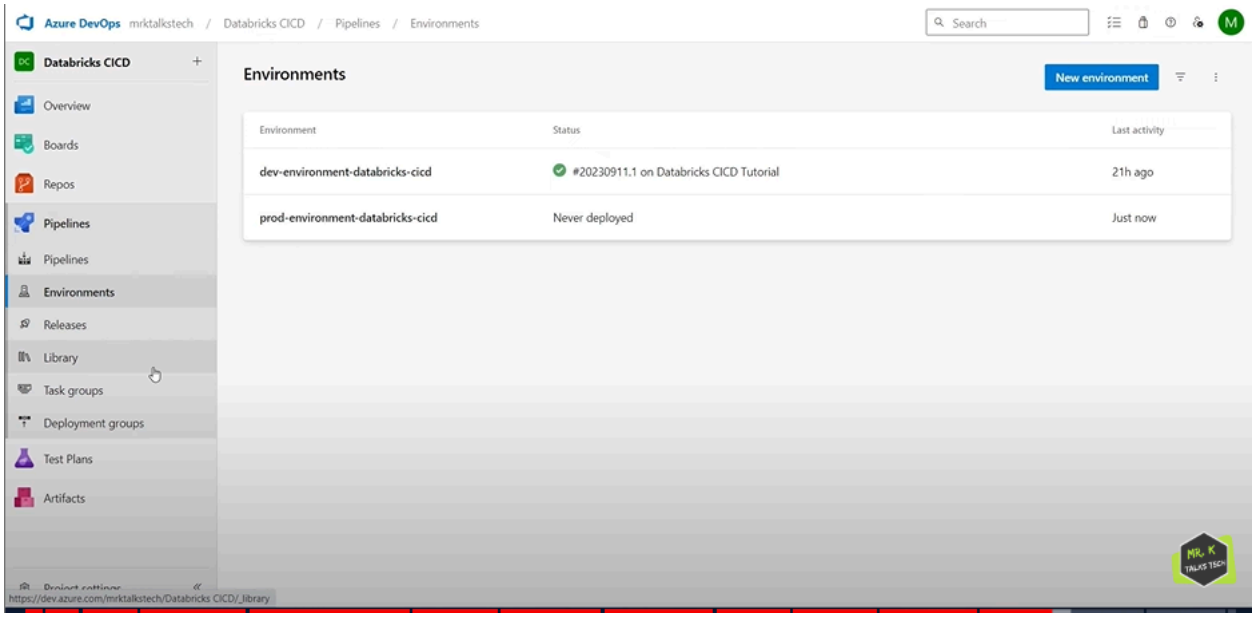
```yaml
stages:
  - template: templates/deploy-notebooks.yml
    parameters:
      stageId: "Deploy_to_Dev_Environment"
      env: "dev"
      environmentName: $(dev-environment-name)
      resourceGroupName: $(dev-resource-group-name)
      serviceConnection: $(dev-service-connection-name)
      notebooksPath: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

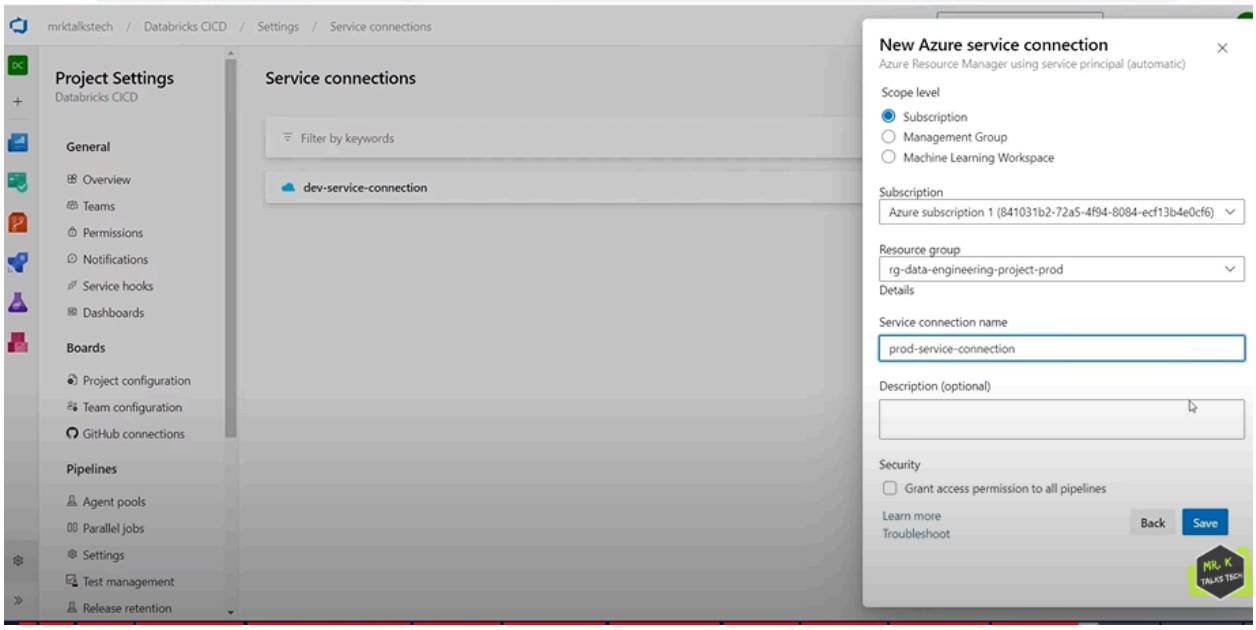
6. Next we will create the variables groups for prod as we did for dev env

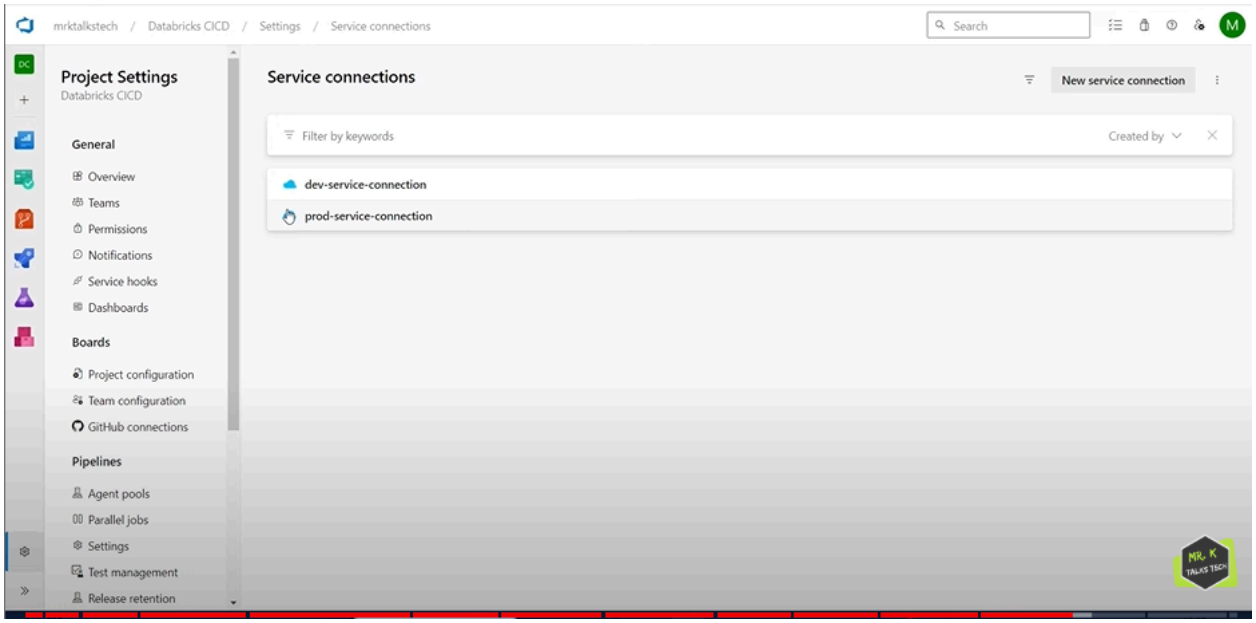


7. Next we will add our env for our prod



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





9. Next we will give all the required permissions for our CICD pipeline