**<https://www.youtube.com/watch?v=iLYIx3QS508&list=PLv2BtOtLblH1RhbtfTpp9ovi3Y-3HiRO2&index=22>**

**Step 1: Enable Git Integration in Fabric**

1. Go to your **Microsoft Fabric workspace**.
2. On the top-right, click the **workspace settings (⚙️)**.
3. Under **Git integration (Preview)**, choose **Connect to a Git repository**.
4. Select **Azure DevOps Git** → Sign in.
5. Choose your **Azure DevOps organization, project, and repo**.
6. Pick the **branch** (usually main).

👉 Now, your workspace items (Notebooks, Pipelines, Dataflows, Semantic Models, Reports, etc.) will sync as **JSON/YAML files** into that repo.

**🔹 Step 2: Verify Files in Azure DevOps**

* Go to your **Azure DevOps repo** → you should see Fabric artifacts exported as code (JSON/YAML).
* Example structure:
* /dataflows/
* /pipelines/
* /notebooks/
* /semanticmodels/
* /reports/

**🔹 Step 3: Sync Azure DevOps Repo → GitHub**

Since Fabric doesn’t connect directly to GitHub yet, you set up a **mirror sync**. Two common options:

**✅ Option A: Azure DevOps Service Hook (Automatic Sync)**

1. In Azure DevOps → go to **Project Settings → Service Hooks**.
2. Add a new service hook → select **GitHub**.
3. Authenticate with GitHub and select the target repo.
4. Choose trigger: **When code is pushed**.
5. Configure action: **Push commit to GitHub repo**.

**✅ Option B: Pipeline Sync (More Control)**

1. Create a pipeline in Azure DevOps (azure-pipelines.yml) like this:
2. trigger:
3. branches:
4. include:
5. - main
6. pool:
7. vmImage: 'ubuntu-latest'
8. steps:
9. - checkout: self
10. - script: |
11. git config --global user.email "you@example.com"
12. git config --global user.name "Azure DevOps Bot"
13. git clone https://github.com/YOUR\_GITHUB\_USER/YOUR\_REPO.git github-repo
14. cd github-repo
15. git remote add upstream https://$(System.AccessToken)@dev.azure.com/YOUR\_ORG/YOUR\_PROJECT/\_git/YOUR\_REPO
16. git pull upstream main --rebase
17. git push origin main
18. displayName: 'Sync Azure DevOps to GitHub'
    * Replace placeholders with your org/project/repo details.
    * Store your **GitHub PAT** as a pipeline secret.

**🔹 Step 4: Confirm in GitHub**

* Check your GitHub repo → you should see all the Fabric content in folders (/notebooks, /pipelines, /reports, etc.).
* Now you can manage with **pull requests, branching, and CI/CD** directly in GitHub.

✅ End Result:

* Fabric workspace → syncs to Azure DevOps (official integration).
* Azure DevOps → auto-mirrors to your GitHub account.