CODING ASSESMENT

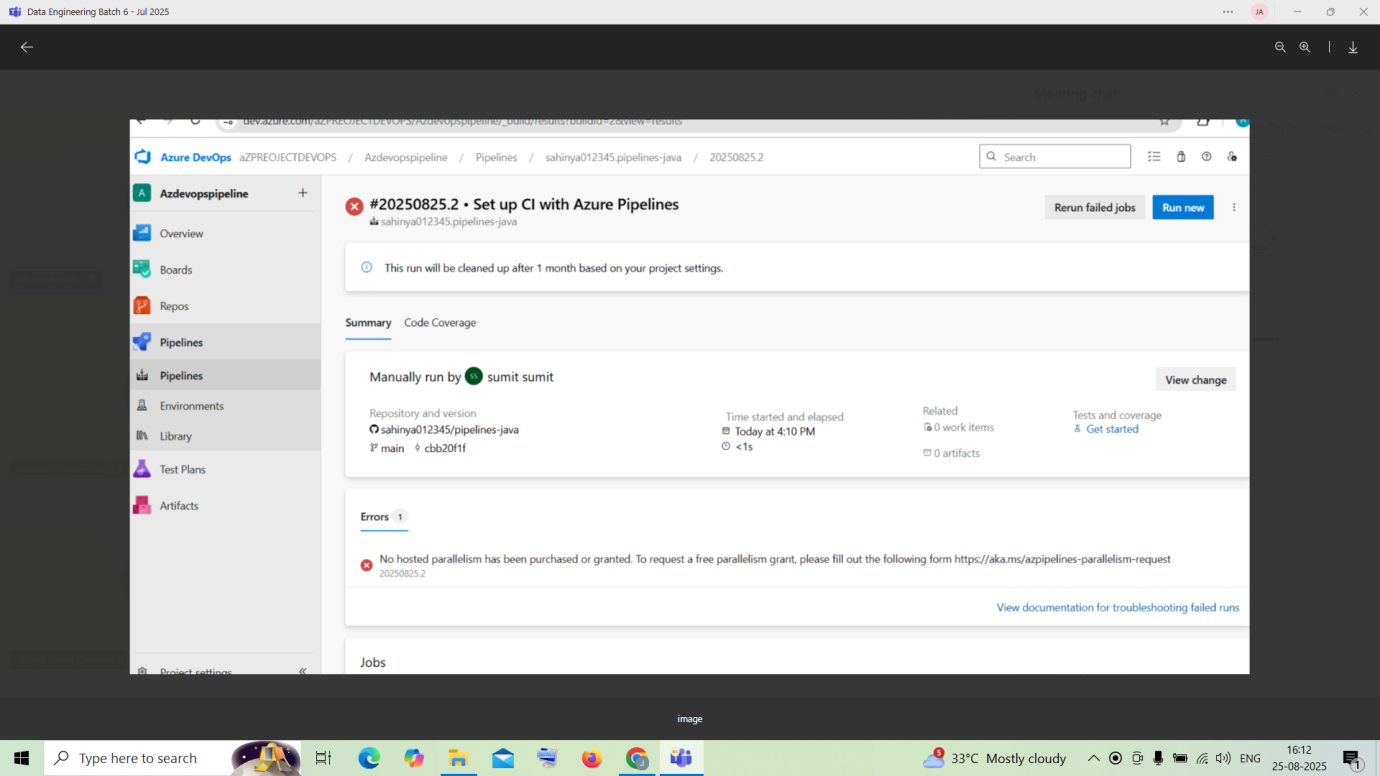
Implement one of cicd pipeline activity in azure devops /azure data factory, azure databricks

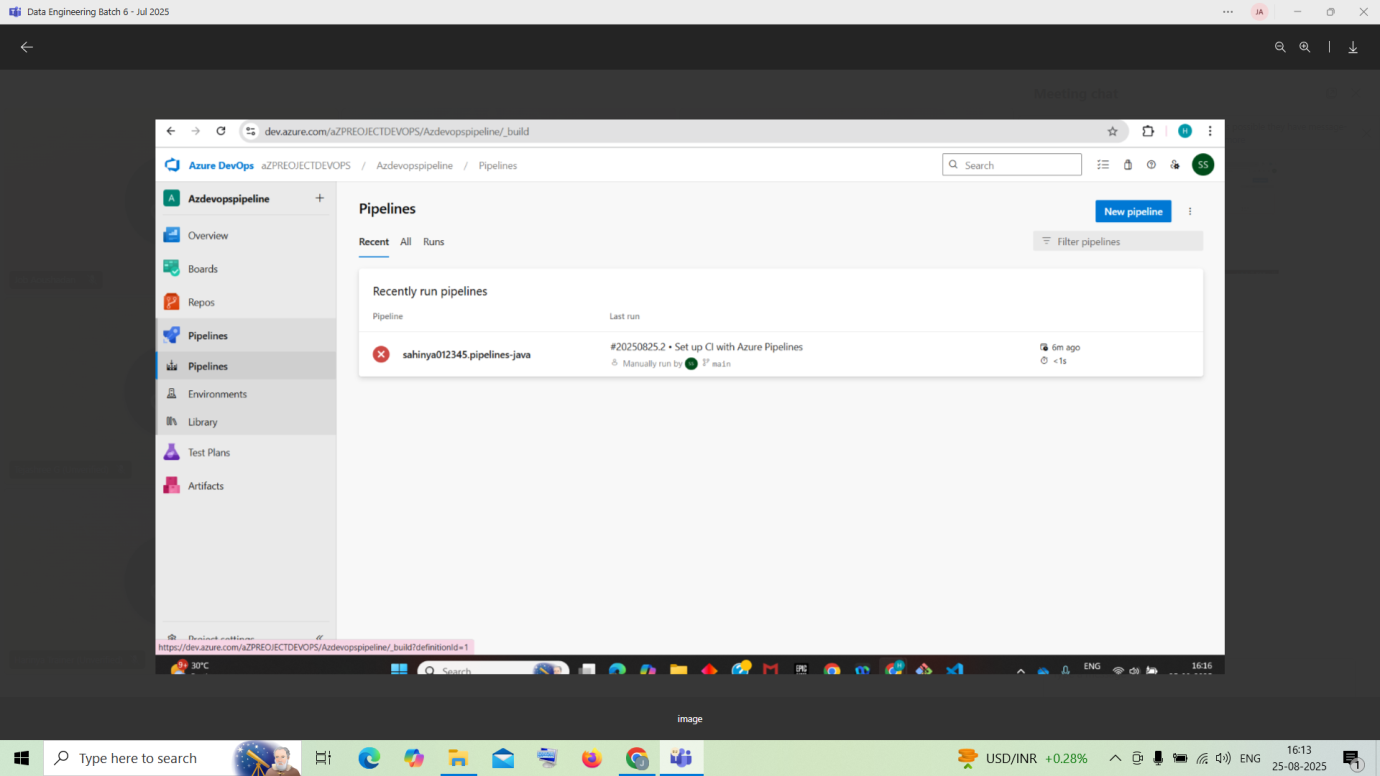
[Jobaoushadancse2021@jerusalemengg.ac.in](mailto:Jobaoushadancse2021@jerusalemengg.ac.in)

Job Aoushadan N – 29.08.2025

**Steps to Create Pipeline in Azure DevOps**

1. **Go to Pipelines** → In your Azure DevOps project, click **Pipelines** → **New Pipeline**.
2. **Select Code Source** → Choose repo (Azure Repos, GitHub, etc.).
3. **Choose Pipeline Type**:
   * **YAML** → Modern, code-based (azure-pipelines.yml).
   * **Classic Editor** → GUI drag-and-drop tasks.
4. **Define Pipeline**:
   * In YAML → Add build/test/deploy steps.
   * In Classic → Add tasks (Build, Test, Publish).
5. **Save & Run** → Commit YAML or save pipeline, then run it.
6. **Monitor** → Check logs, artifacts, and results.





**Detailed Overview:**

* ADF supports **Git integration** (Azure Repos or GitHub) for your development data factory. Once you author pipelines and datasets in your dev environment and merge changes, ARM templates are auto-generated for deployment.
* The pipeline typically follows this flow:
  1. Develop in feature branches and merge to a main/collaboration branch.
  2. ADF publishes and auto-generates ARM templates into an adf\_publish branch.
  3. CI/CD pipelines in Azure DevOps pick up those ARM templates to deploy to higher environments—UAT/test, then production.
* **Best practices**:
  1. Only the development factory should be Git-integrated; UAT and prod are deployed via pipelines to avoid direct editing.
  2. Use pre- and post-deployment scripts (PowerShell) to manage triggers or parameter overrides.
  3. Automate ARM template generation using the @microsoft/azure-data-factory-utilities npm package to sidestep manual publishes.

### Step-by-Step Example

#### 1. ****Set up Git integration in ADF****

* In ADF Studio, go to Manage → Git Configuration, choose Azure DevOps Git, and connect to your repo with collaboration branch (e.g., main) and a publish branch (adf\_publish).

#### 2. ****Develop and publish****

* Author pipelines in feature branches, push changes → create PR → merge to main.
* Upon publishing, ARM templates and parameter files are generated automatically in adf\_publish.

#### 3. ****Build Pipeline (CI)****

* Create an Azure Pipelines YAML or classic build pipeline that:
  + Triggers on changes to the adf\_publish branch.
  + Uses Azure DevOps **ARM template deployment task** to validate or bundle templates.
* Optionally integrate the npm auto-publish utility to generate artifacts without using the ADF UI.

#### 4. ****Release Pipeline (CD)****

* Create stages for **UAT** and **Production**.
* Deploy ARM templates via AzureResourceManagerTemplateDeployment tasks to each environment's resource group.
* Insert approval gates before UAT or production deployments using Azure DevOps Environments.

#### 5. ****Parameterization & scripts****

* Use pre-deployment scripts to disable triggers or update connections; post-deployment to re-enable them or add global parameters.
* Parameterize environment-specific settings (e.g., connection strings) via ARM parameter files