CODING CHALLENGE

IMPLEMENTING CICD PIPELINE IN AZURE DEVOPS

1. Introduction

CI/CD (Continuous Integration / Continuous Deployment) automates the process of:

- Building, testing, and deploying code
- Ensuring fast and reliable delivery of software changes

In this demo, we implement a simple CI/CD pipeline in Azure DevOps that runs a Python script whenever code is pushed to the repository.

2. Python Script

File: hello_world.py

```
# hello_world.py

print("Hello CI/CD from Azure DevOps Pipeline!")
```

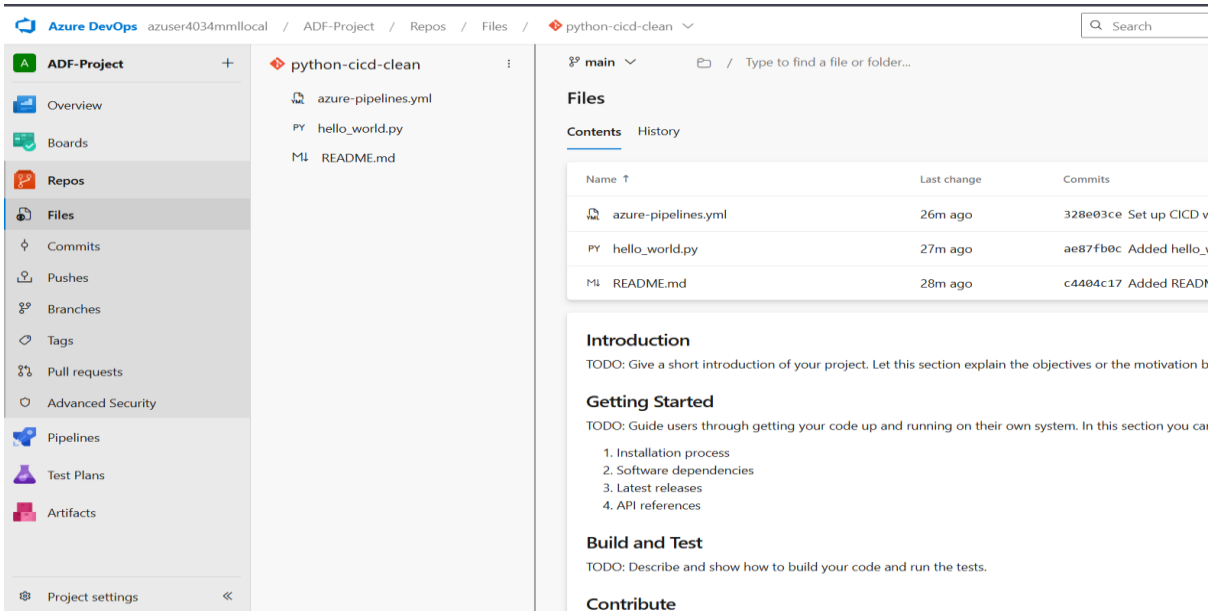
Purpose:

- Demonstrate automatic code execution in the pipeline.

3. Repository Setup

Steps:

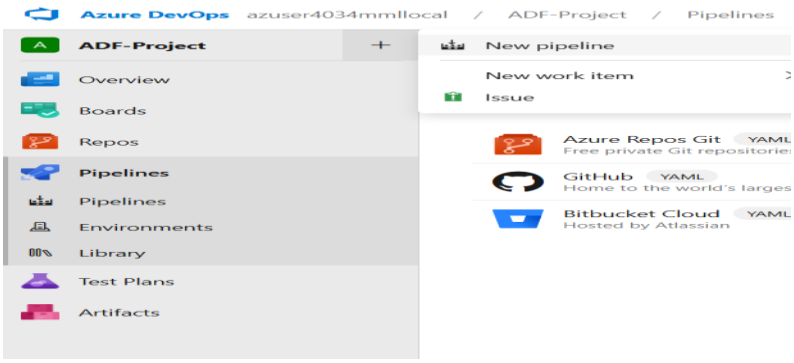
1. Open Azure DevOps → Repos → Files
2. Create a repository: python-cicd-demo
3. Upload hello_world.py
4. Commit to main branch



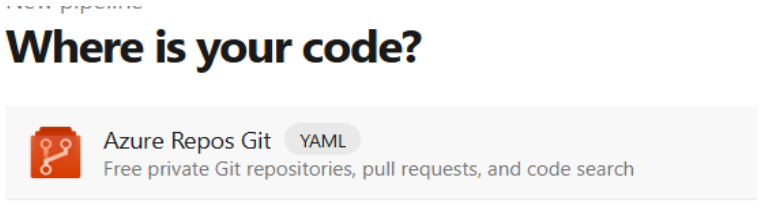
4.Azure Pipeline Setup

Steps:

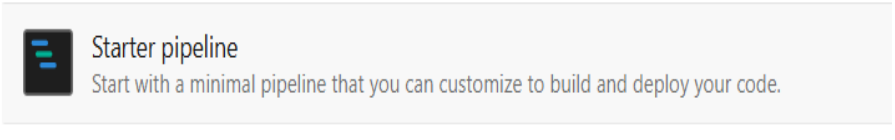
1. Navigate to Pipelines → New Pipeline



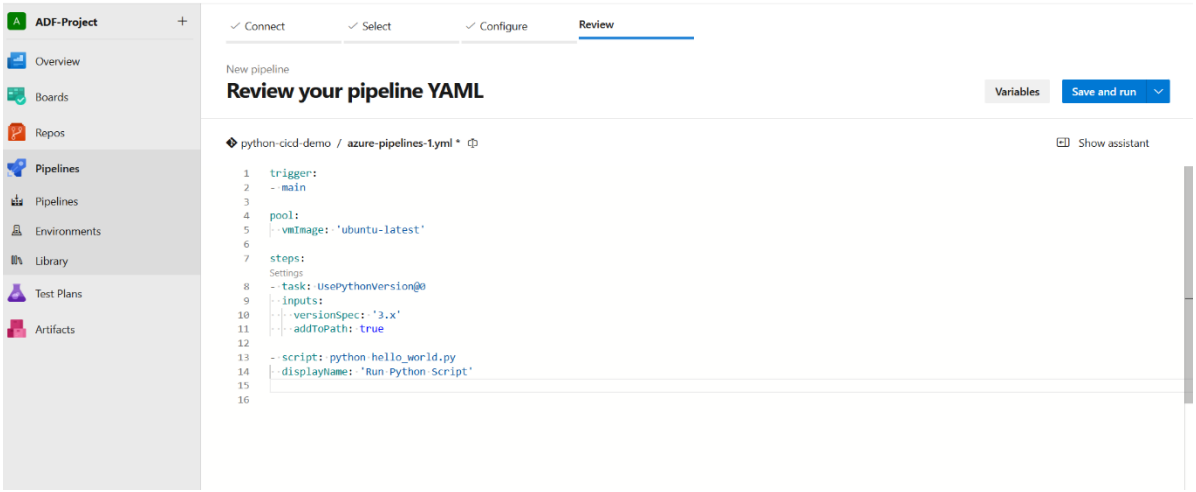
2. Choose Azure Repos Git → Your repository



3. Select Starter Pipeline (YAML)



4. Replace the YAML with:




```
trigger:
- main

pool:
  vmImage: 'ubuntu-latest'

steps:
- task: UsePythonVersion@0
  inputs:
    versionSpec: '3.x'
    addToPath: true

- script: python hello_world.py
  displayName: 'Run Python Script'
```

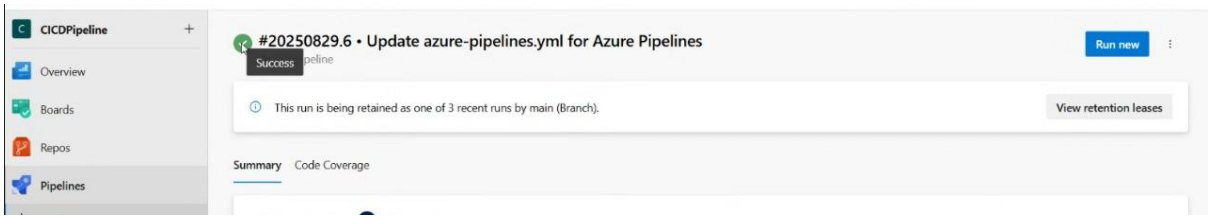
Explanation:

- trigger: main → runs pipeline automatically on code push
- UsePythonVersion@0 → installs Python 3.x
- script → runs Python script

5. Pipeline Execution

Steps:

1. Click **Save → Run**
2. Pipeline executes:
    - Installs Python
    - Runs Python script
    - Displays output in logs



6. Key Takeaways

- CI/CD pipelines automate code execution on push events
- Reduces manual testing and deployment errors
- Pipelines can be extended to run unit tests, build artifacts, and deploy applications

7. Conclusion

The CI/CD pipeline implemented in Azure DevOps successfully demonstrates automation of Python code execution. Key points:

- Automation: The pipeline automatically runs whenever code is pushed to the repository, ensuring consistent execution.
- Efficiency: Manual steps such as running scripts and verifying output are eliminated.
- Scalability: The same pipeline structure can be extended to run complex Python projects, include unit tests, or deploy applications.
- Flexibility: Free-tier or self-hosted agents allow running pipelines even without a paid Azure DevOps subscription.

- Practical Learning: This demo provides a hands-on understanding of CI/CD concepts, YAML pipeline setup, and integration with version control.

Overall: The project illustrates how CI/CD pipelines streamline software workflows, improve reliability, and provide a foundation for more advanced DevOps and automation practices.