

# Implementing a CI/CD Pipeline in Azure DevOps

**Name:** Priyeshwar

**Mail:** [priyesh2664@gmail.com](mailto:priyesh2664@gmail.com)

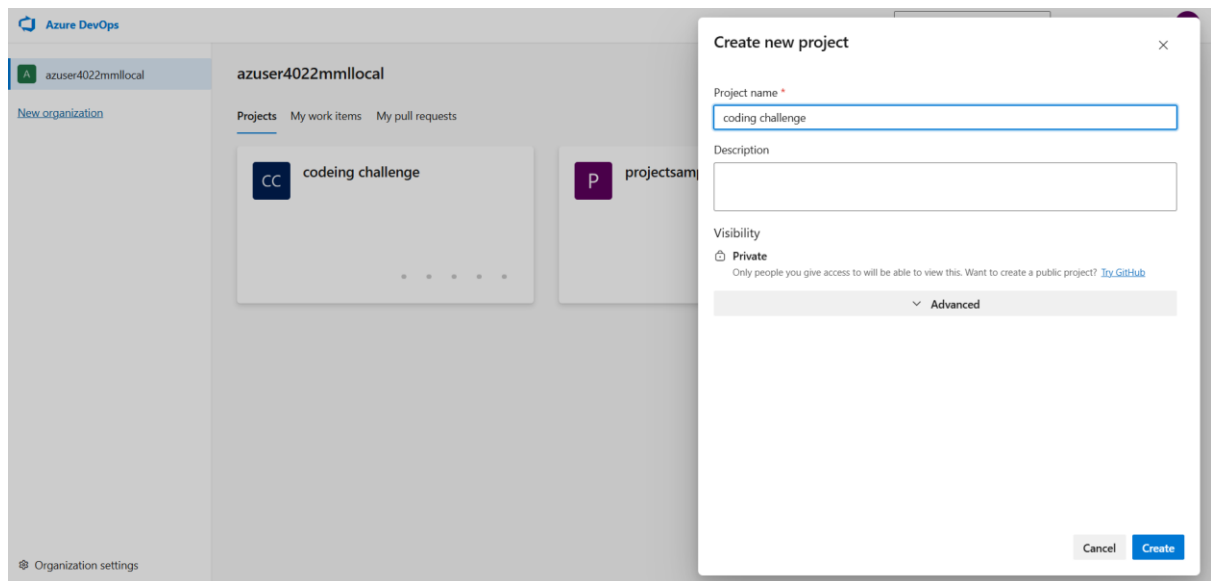
---

A CI/CD pipeline in Azure DevOps automates the process of building, testing, and deploying applications. It enables teams to deliver software faster, with improved quality and reliability.

---

## Step 1: Create a Project in Azure DevOps

1. Go to <https://dev.azure.com>.
2. Click New Project and provide a name
3. Choose visibility (private or public).



4. Select Git for version control.

---

## Step 2: Connect Source Code Repository

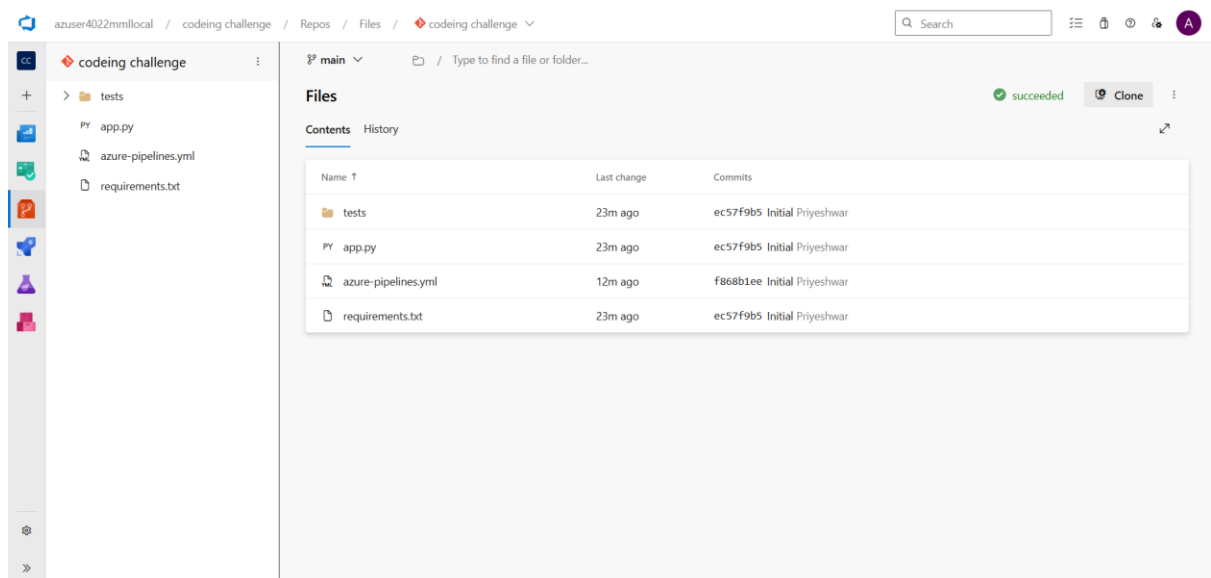
- Push your application code to Azure Repos Git, or connect to an external repository such as GitHub or Bitbucket.

```
MINGW64:/d/DE-Hexaware/Coding Challenge/Devops
$ git commit -m "Initial"
[main f868b1e] Initial
1 file changed, 4 insertions(+), 4 deletions(-)

Priyeshwar@Priyeshwar MINGW64 /d/DE-Hexaware/Coding Challenge/Devops (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 314 bytes | 314.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Analyzing objects... (3/3) (17 ms)
remote: Validating commits... (1/1) done (0 ms)
remote: Storing packfile... done (30 ms)
remote: Storing index... done (35 ms)
remote: Updating refs... done (185 ms)
To https://dev.azure.com/azuser4022mm/local/codeing%20challenge/_git/codeing%20c
challenge
2fcb807..f868b1e main -> main
branch 'main' set up to track 'origin/main'.

Priyeshwar@Priyeshwar MINGW64 /d/DE-Hexaware/Coding Challenge/Devops (main)
$
```

- Ensure your main (or master) branch is set as the default branch for builds.



### Step 3: Configure Continuous Integration (CI) with Azure Pipelines

1. In your project, go to Pipelines → Create Pipeline.
2. Select your source (Azure Repos, GitHub, Bitbucket, etc.).
3. Choose Starter Pipeline or point to an existing azure-pipelines.yml file in your repository.

Pipeline.YAML

trigger:

branches:

include:

- main

pool:

name: 'local-agent' # use your self-hosted agent pool

steps:

- script: python --version

displayName: 'Check Python version'

- script: pip install -r requirements.txt

displayName: 'Install dependencies'

- script: pytest

displayName: 'Run tests'

This pipeline triggers on every commit, automatically builds the application, and executes unit tests.

---

#### Step 4: Add Continuous Testing

- Use Azure Test Plans for manual or exploratory testing.
- Add automated test runs within your CI pipeline (e.g., dotnet test, pytest, or npm test).
- Configure approvals and release gates before production deployment

The screenshot shows the Azure DevOps Pipelines interface. The top navigation bar includes the user profile 'azuser4022mmlocal' and the path 'codeing challenge / Pipelines / codeing challenge / 20250829.5'. A search bar is on the right. The left sidebar contains a 'Jobs' section with a list of jobs for 'codeing challenge'. The main area displays a 'Job' in progress, with a green checkmark and a 'View raw log' button. The job details include: Pool: local-agent, Agent: local-agent, Started: Today at 11:41 AM, Duration: 9s. The job steps are: 1. Initialize job (<1s), 2. Checkout codeing chall... (2s), 3. Install dependencies (3s), 4. Run tests (2s), 5. Post-job: Checkout co... (<1s), 6. Finalize Job (<1s), and 7. Report build status (<1s).

## Step 5: Monitor and Improve

- Enable Azure Application Insights to monitor application performance and detect issues.
- Add Release Gates to block production deployments if monitoring detects failures.
- Use Azure DevOps Dashboards to track build and deployment success, as well as overall project health.

The screenshot shows the Azure DevOps Pipelines interface. The top navigation bar includes the user profile 'azuser4022mmlocal' and the path 'codeing challenge / Pipelines'. A search bar is on the right. The left sidebar contains a 'Pipelines' section with a list of pipelines. The main area displays a 'Recently run pipelines' table. The table has two columns: 'Pipeline' and 'Last run'. The table contains one row: 'codeing challenge' with a green checkmark, '#20250829.5 • Initial', and 'Just now 17s'. The table also shows 'Individual CI for' and '2nd main'.