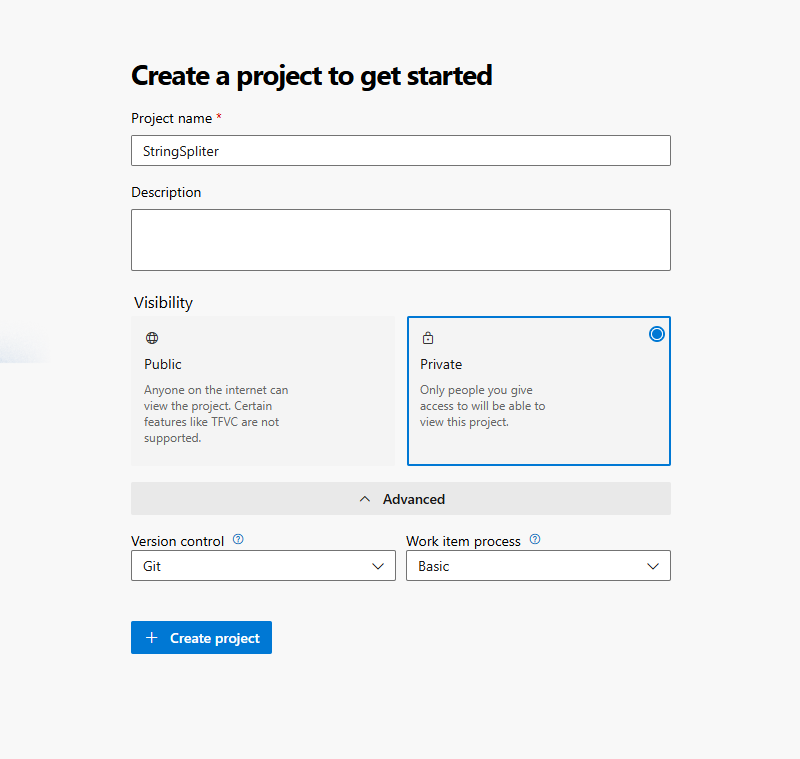
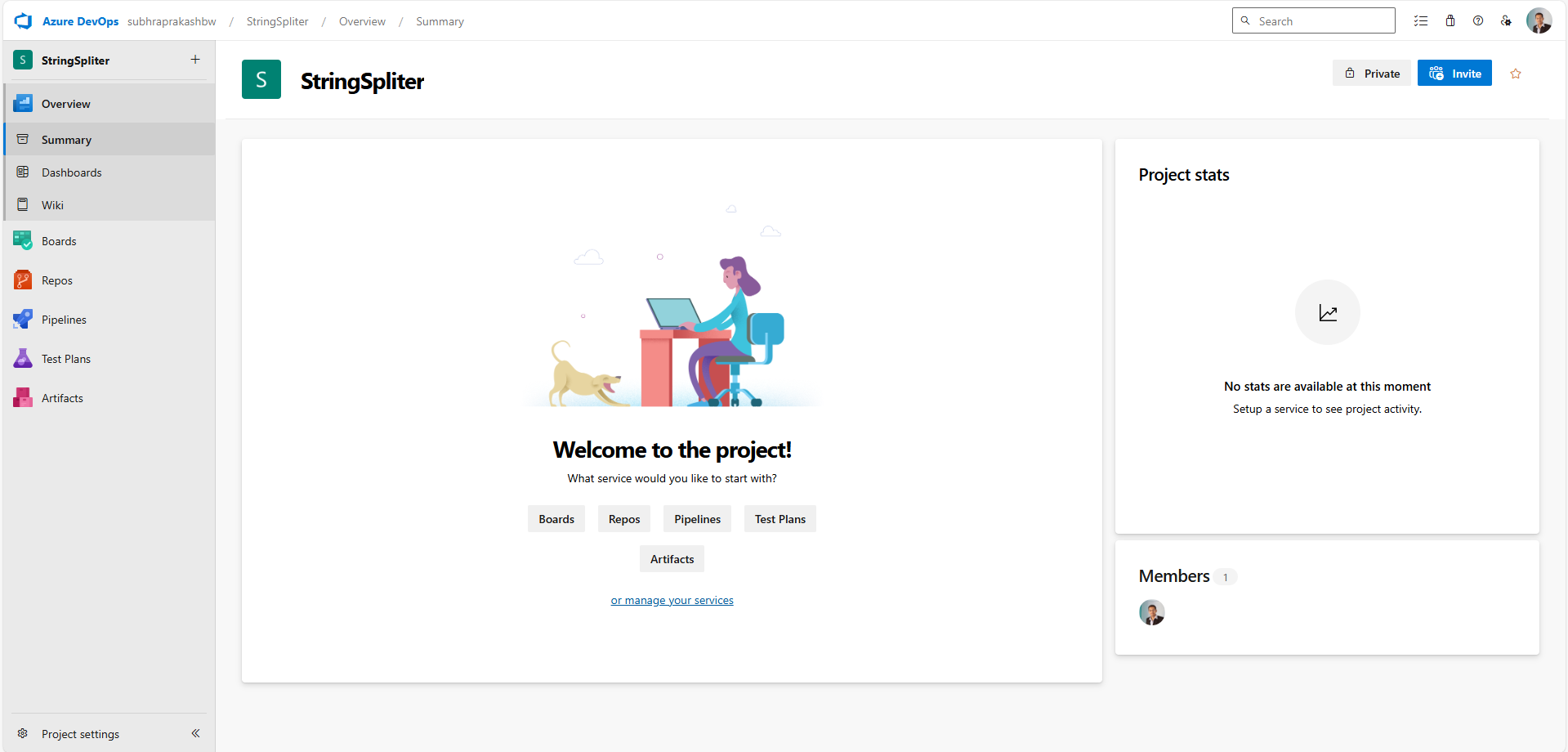
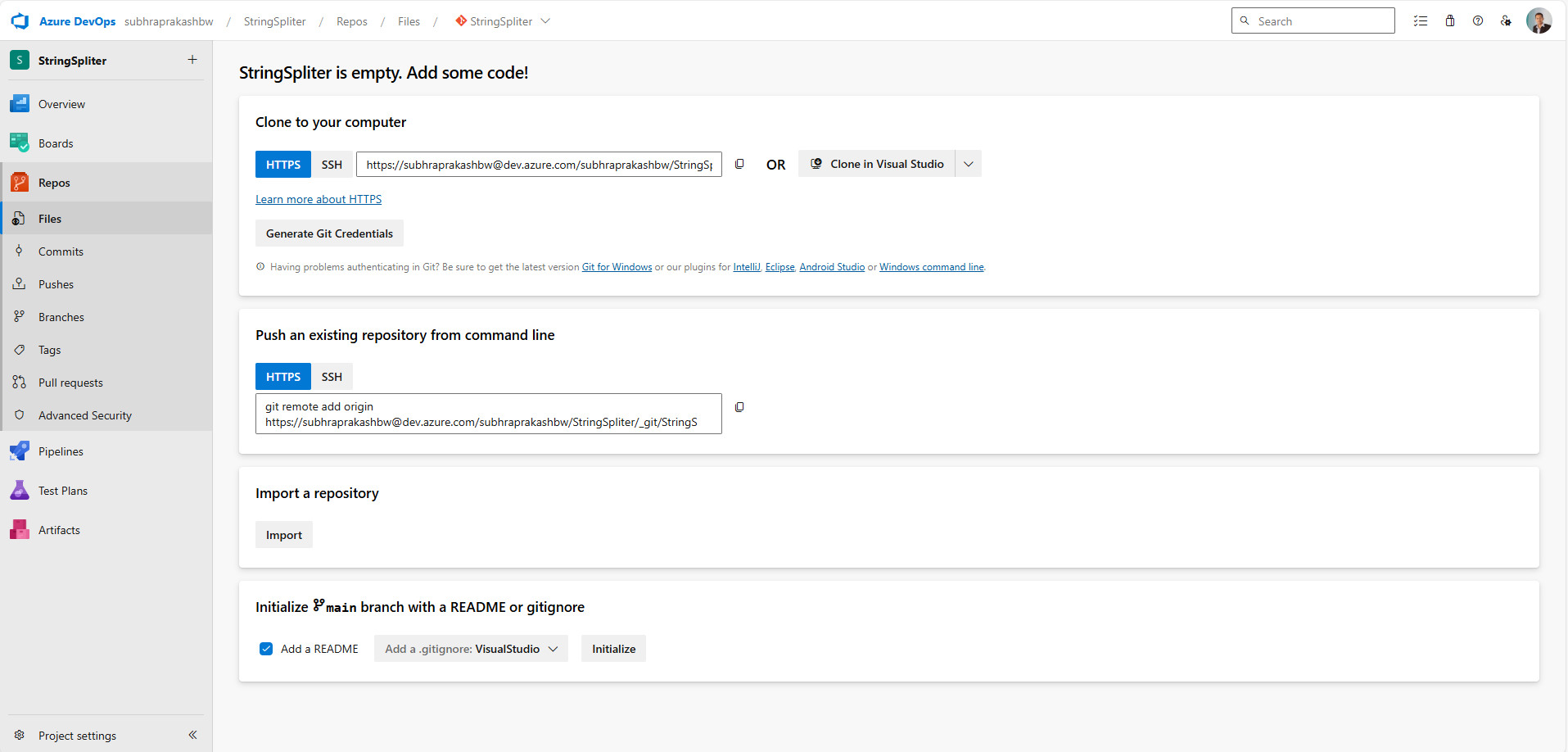
Create a Project



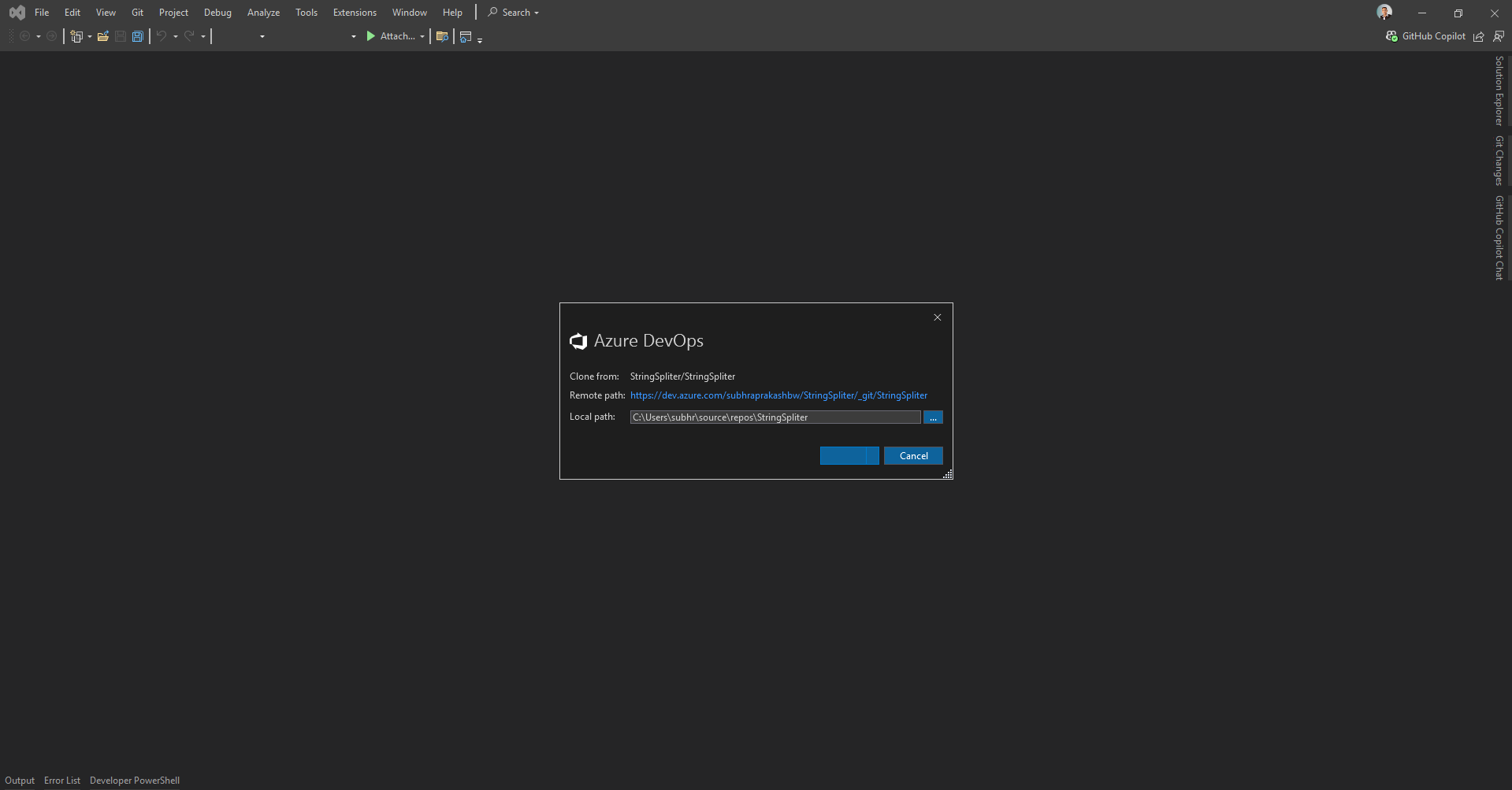
Step 2 :



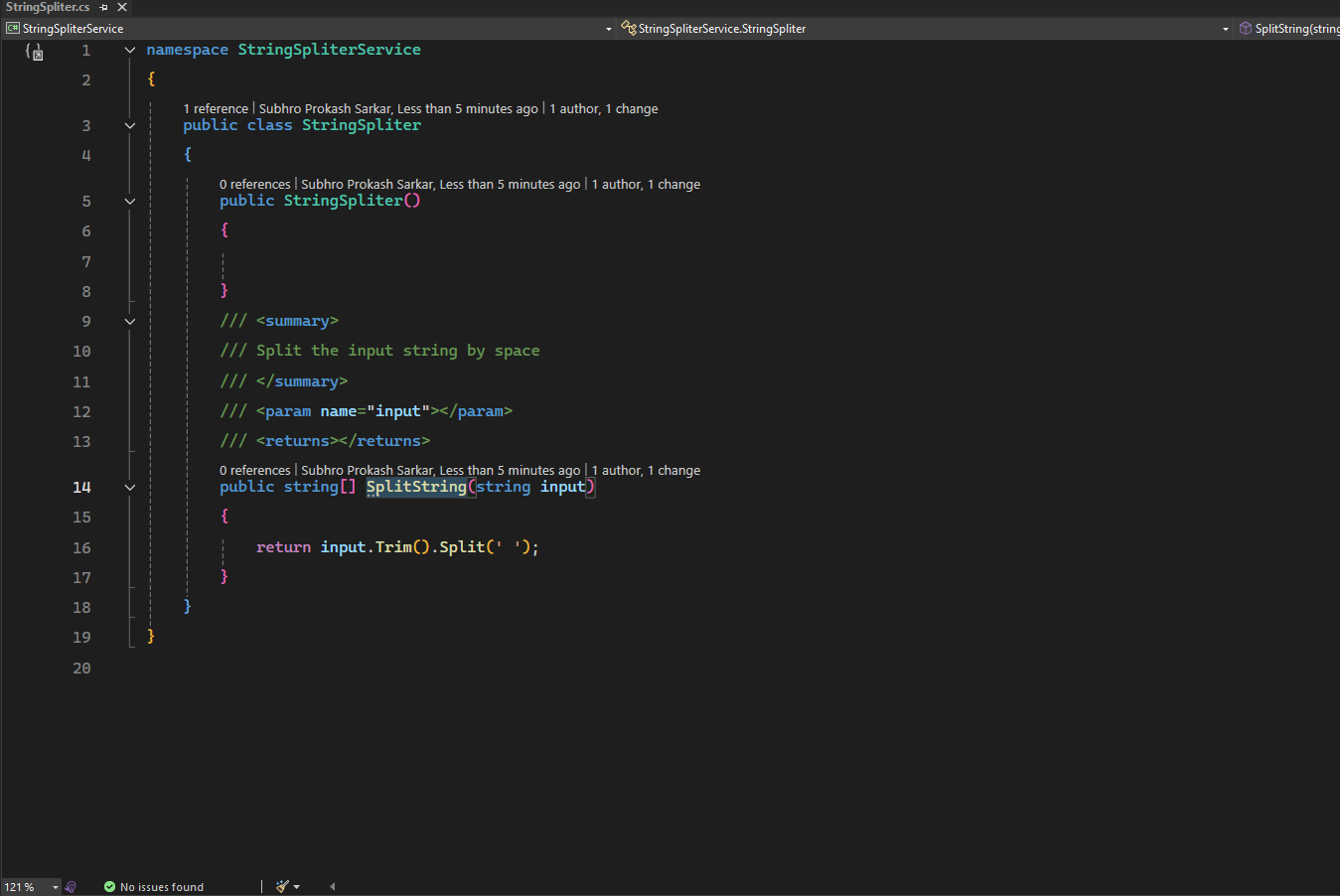
Step 3:



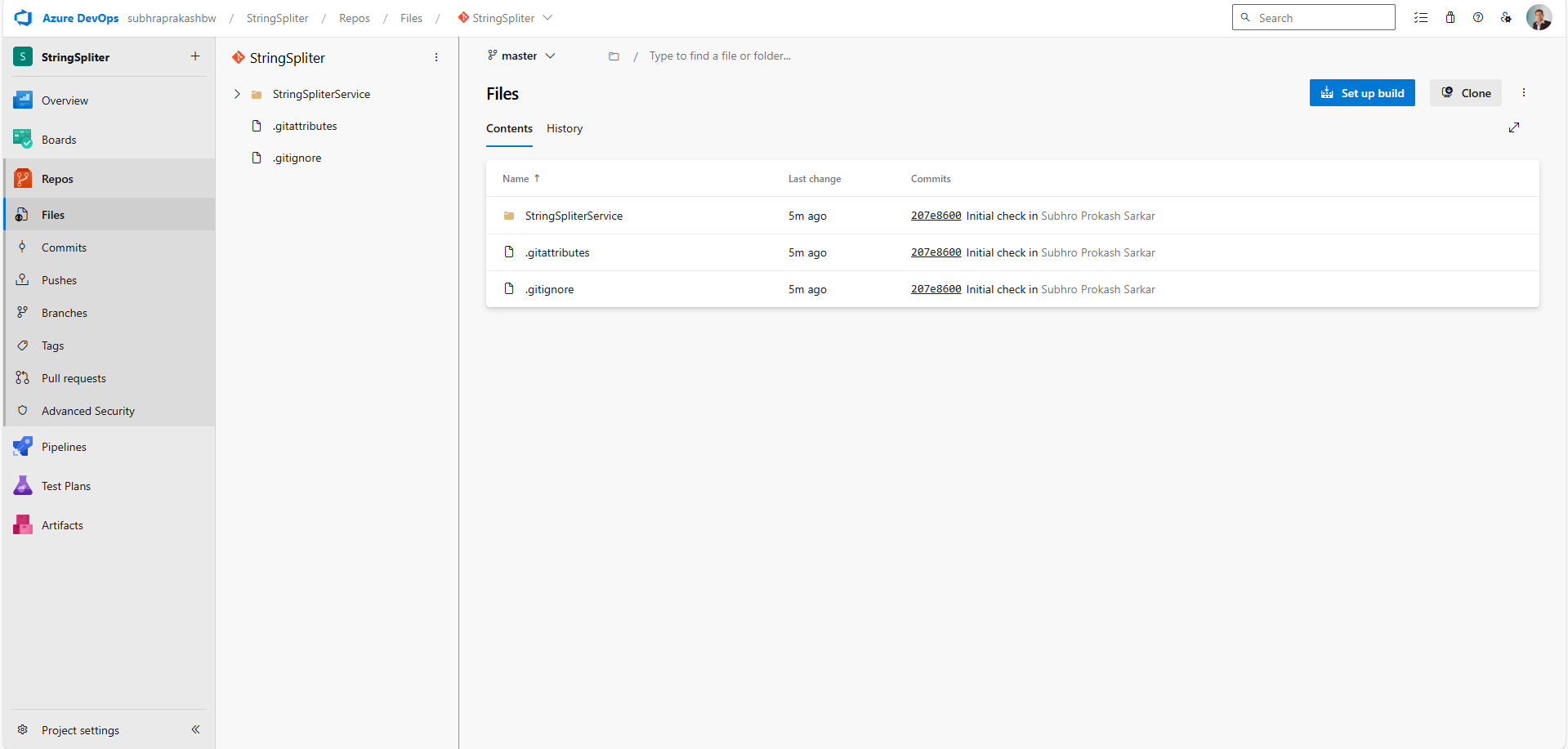
Step 4:



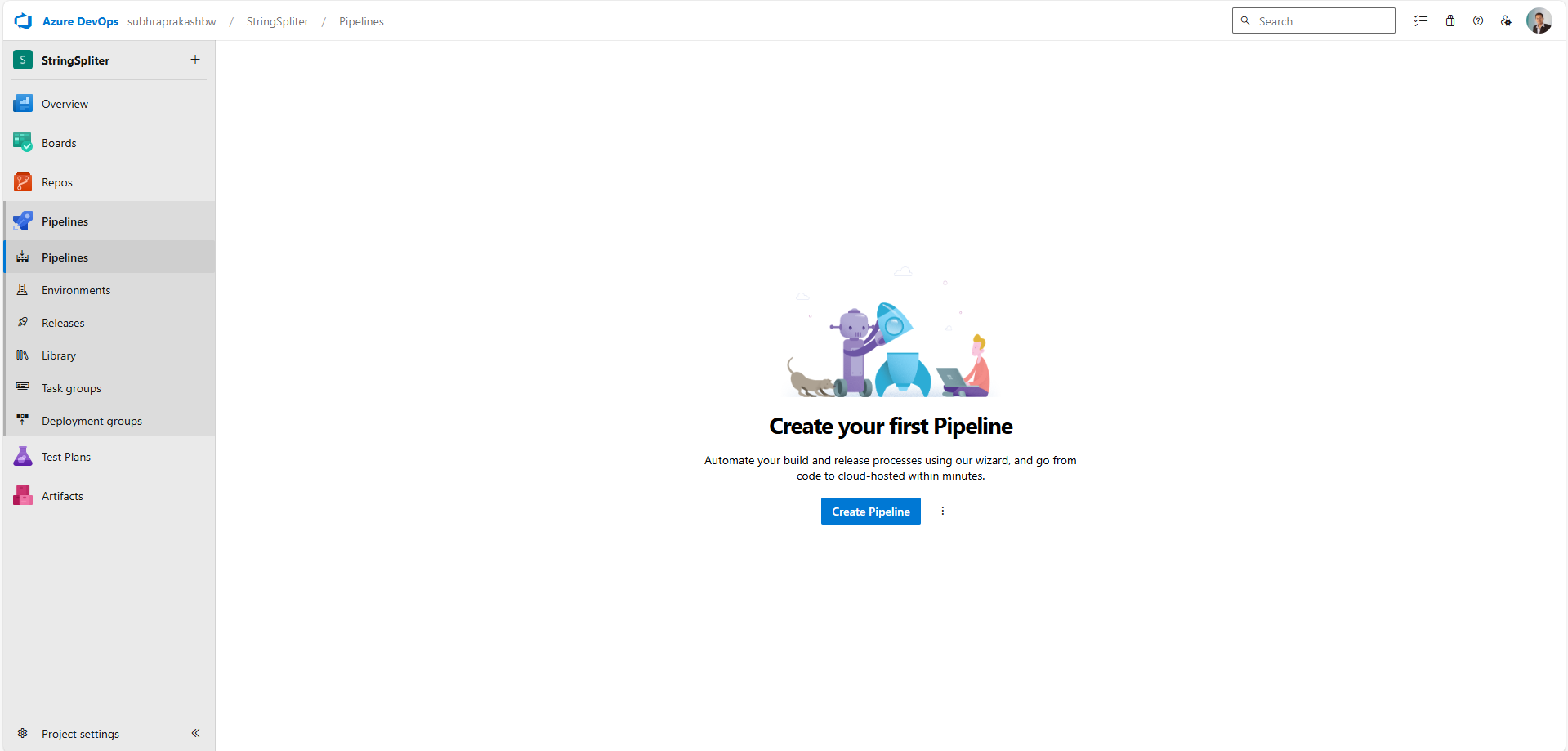
Step 5:



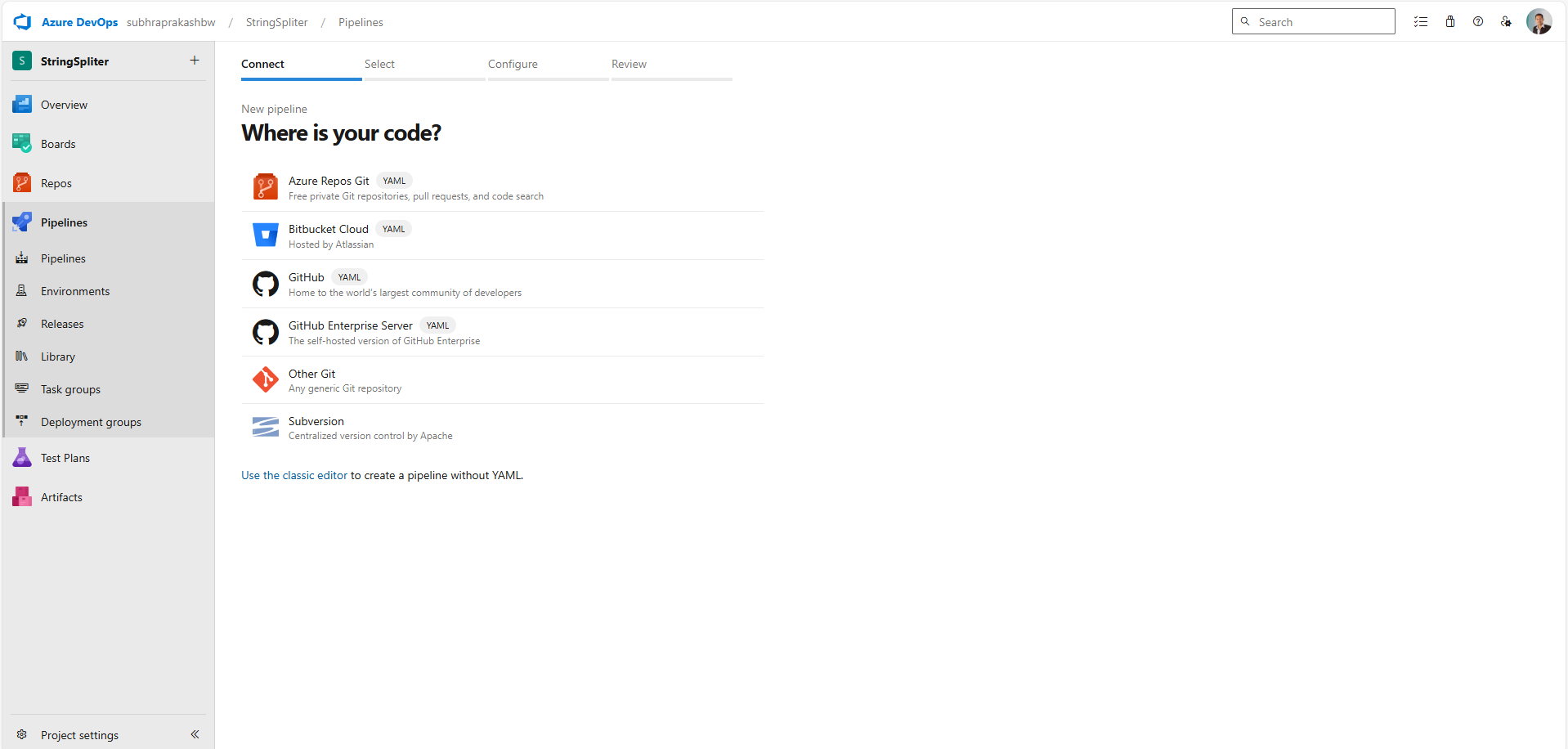
Step 6:



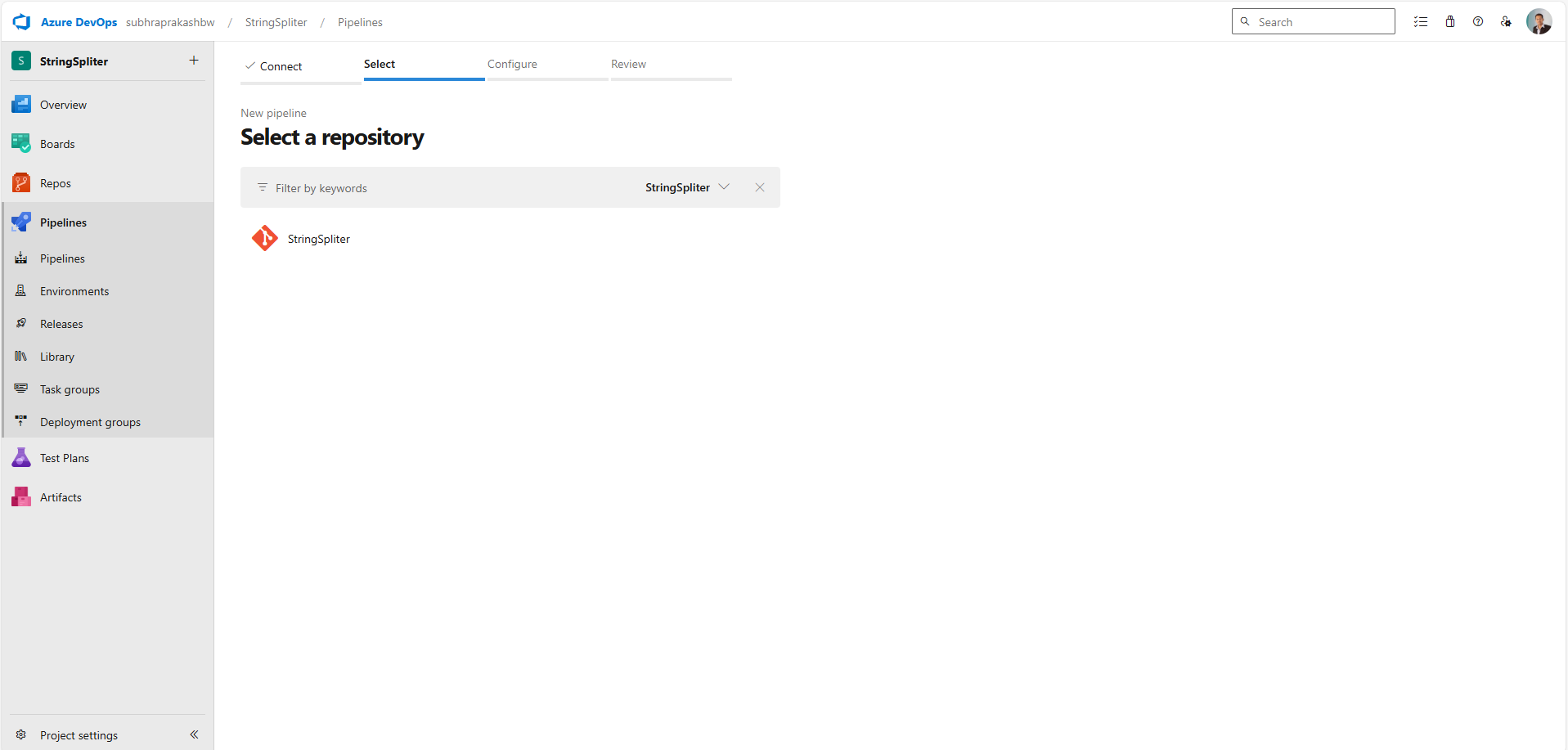
Step 7:



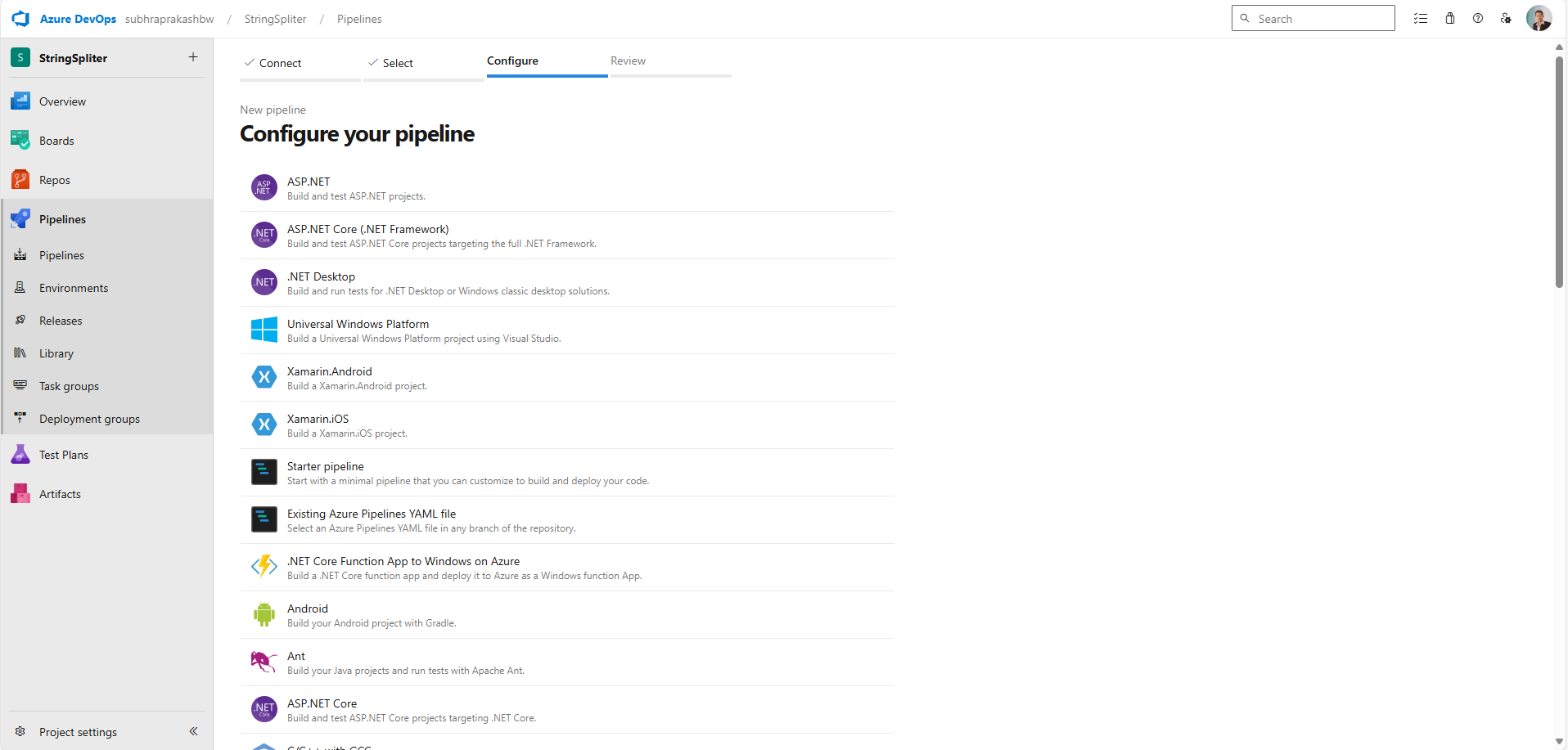
Step 8:



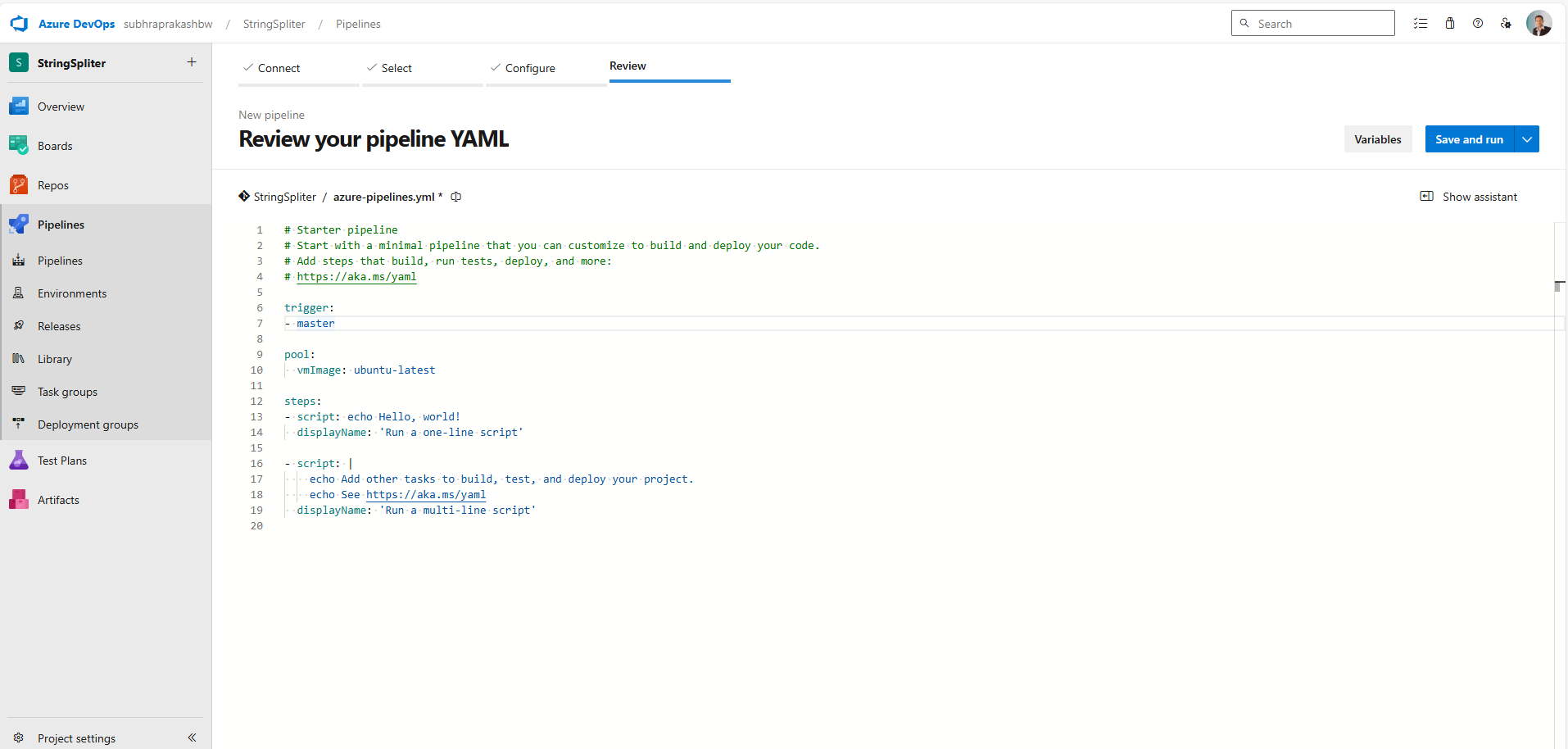
Step 9:



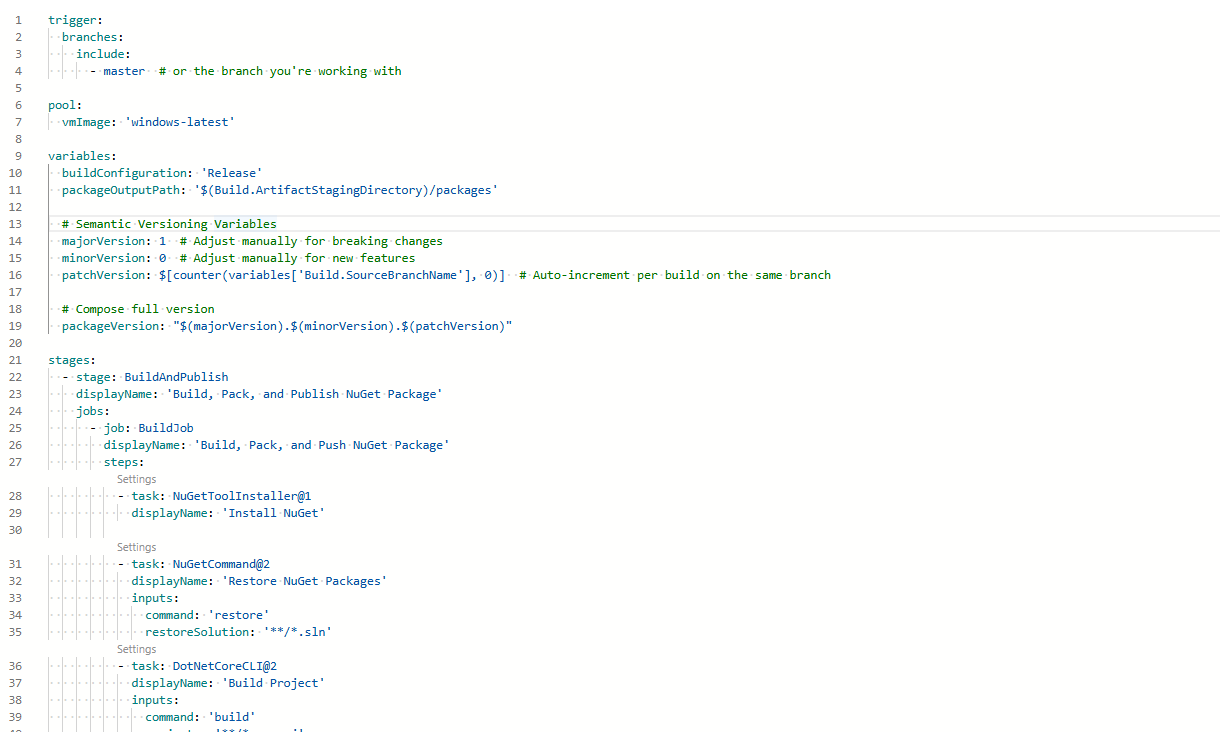
Step 10:



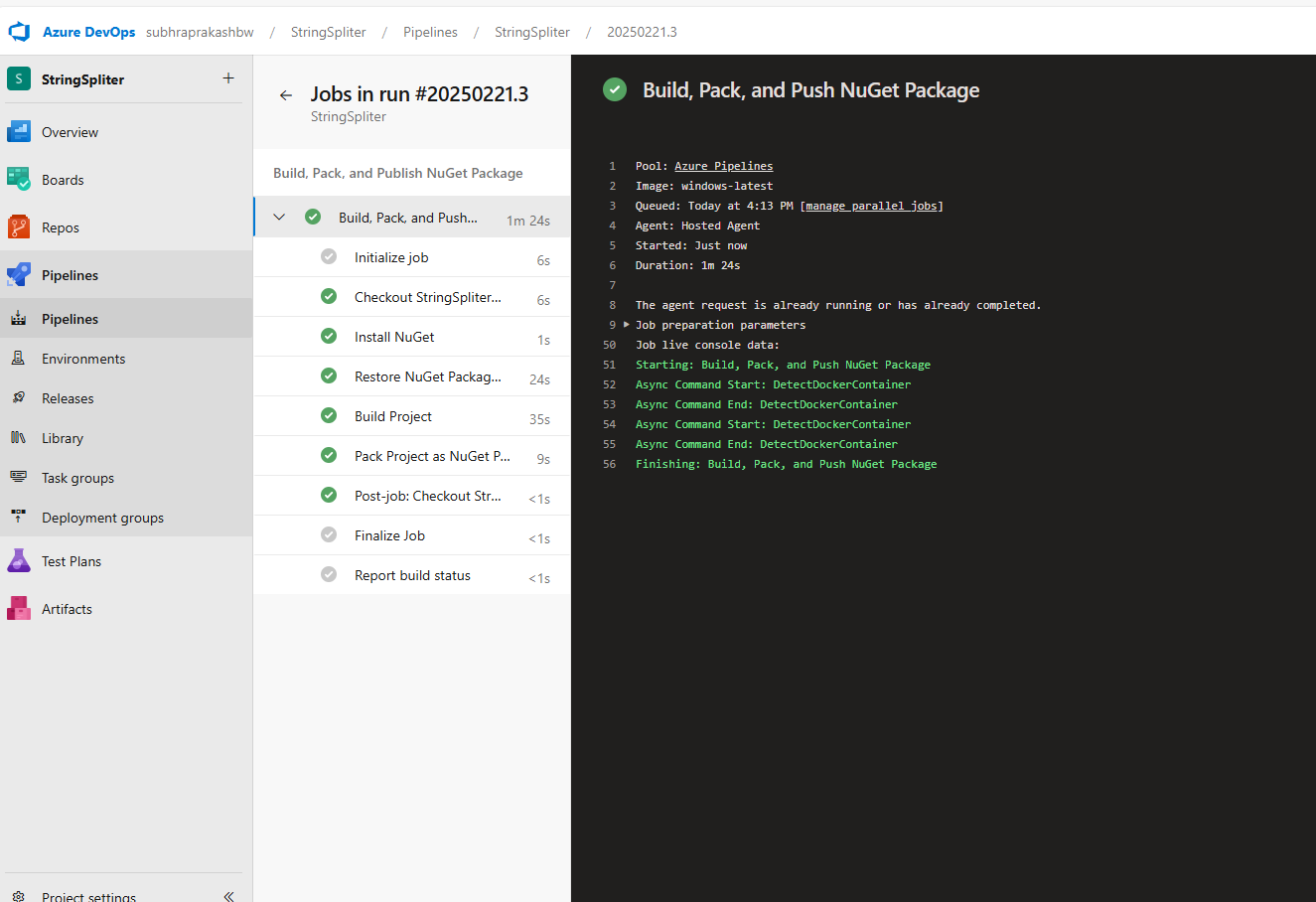
Step 11: initial pipeline yml



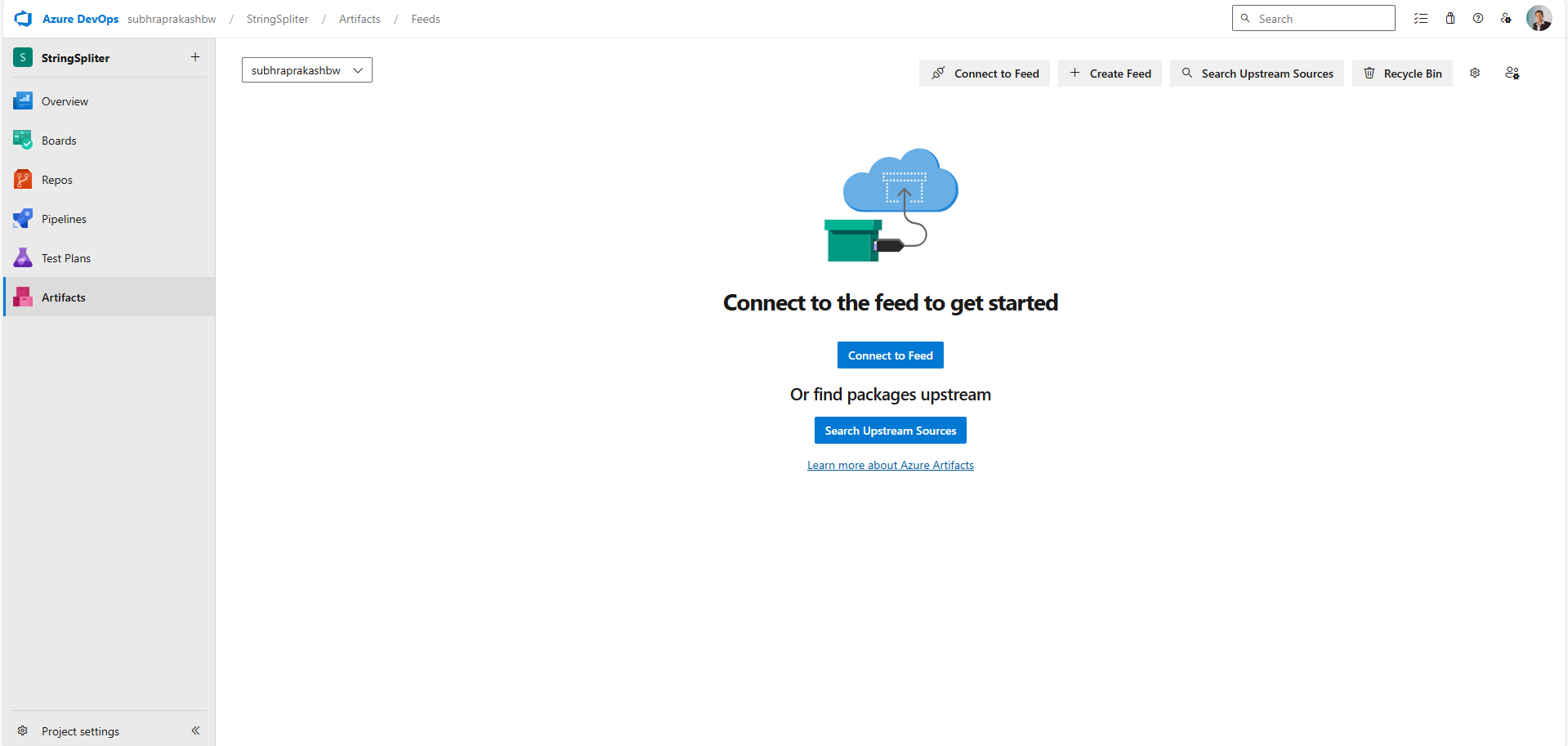
Step 12: Packing nuget package



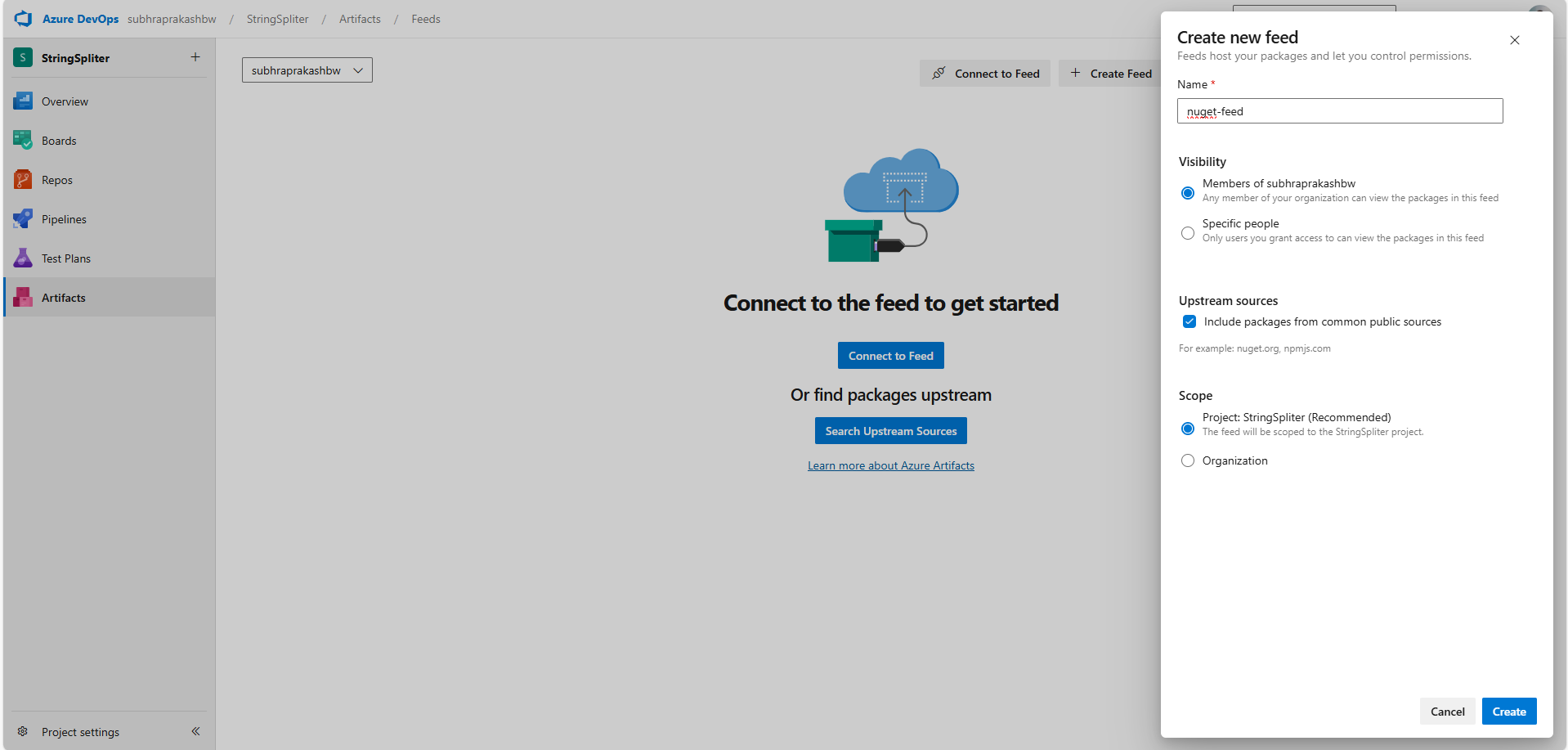
Step 12: Running the pipeline



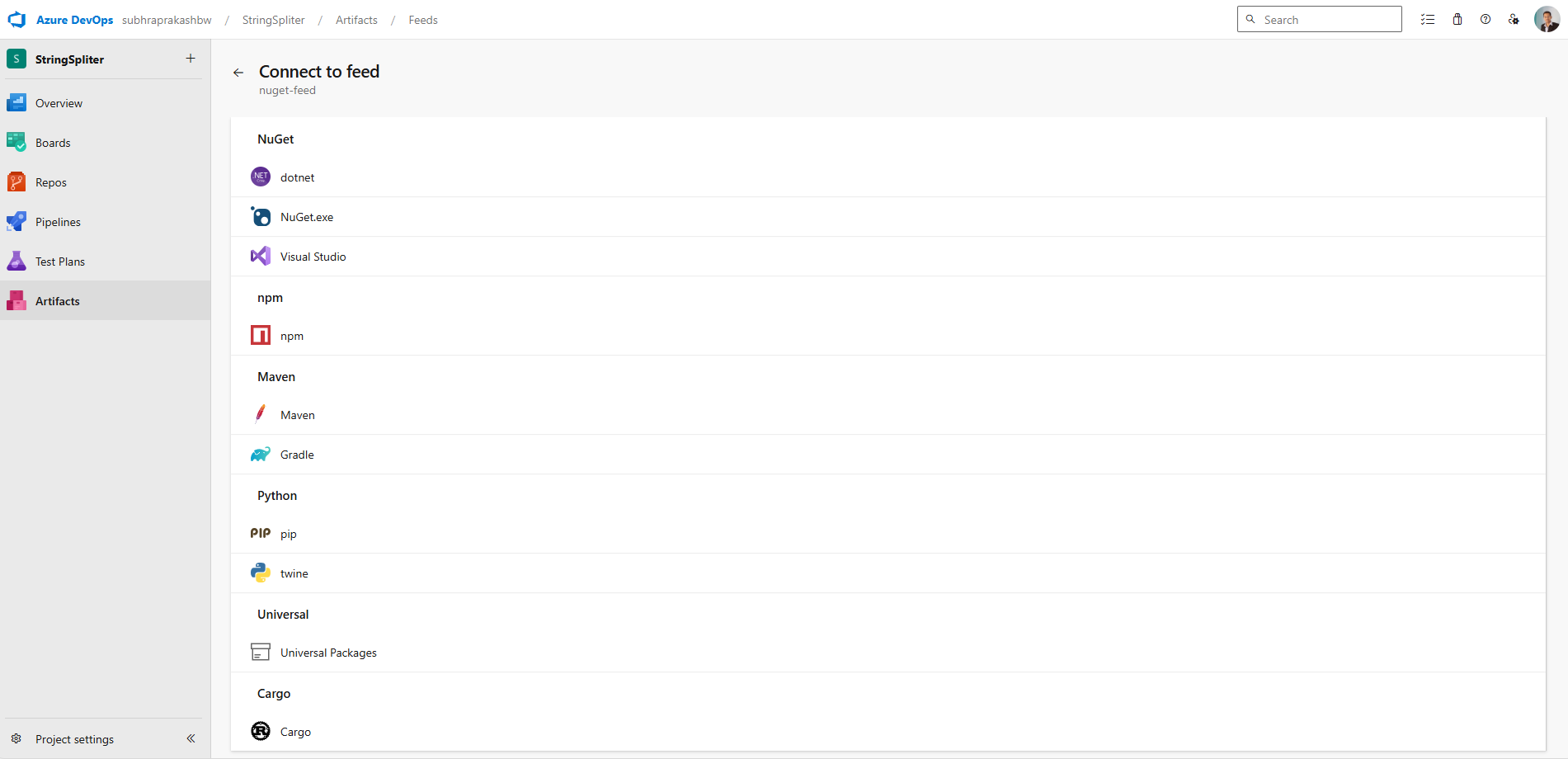
Step 13: releasing the nuget package into the project feed so that visual studio can download



Step 14: Creating feed

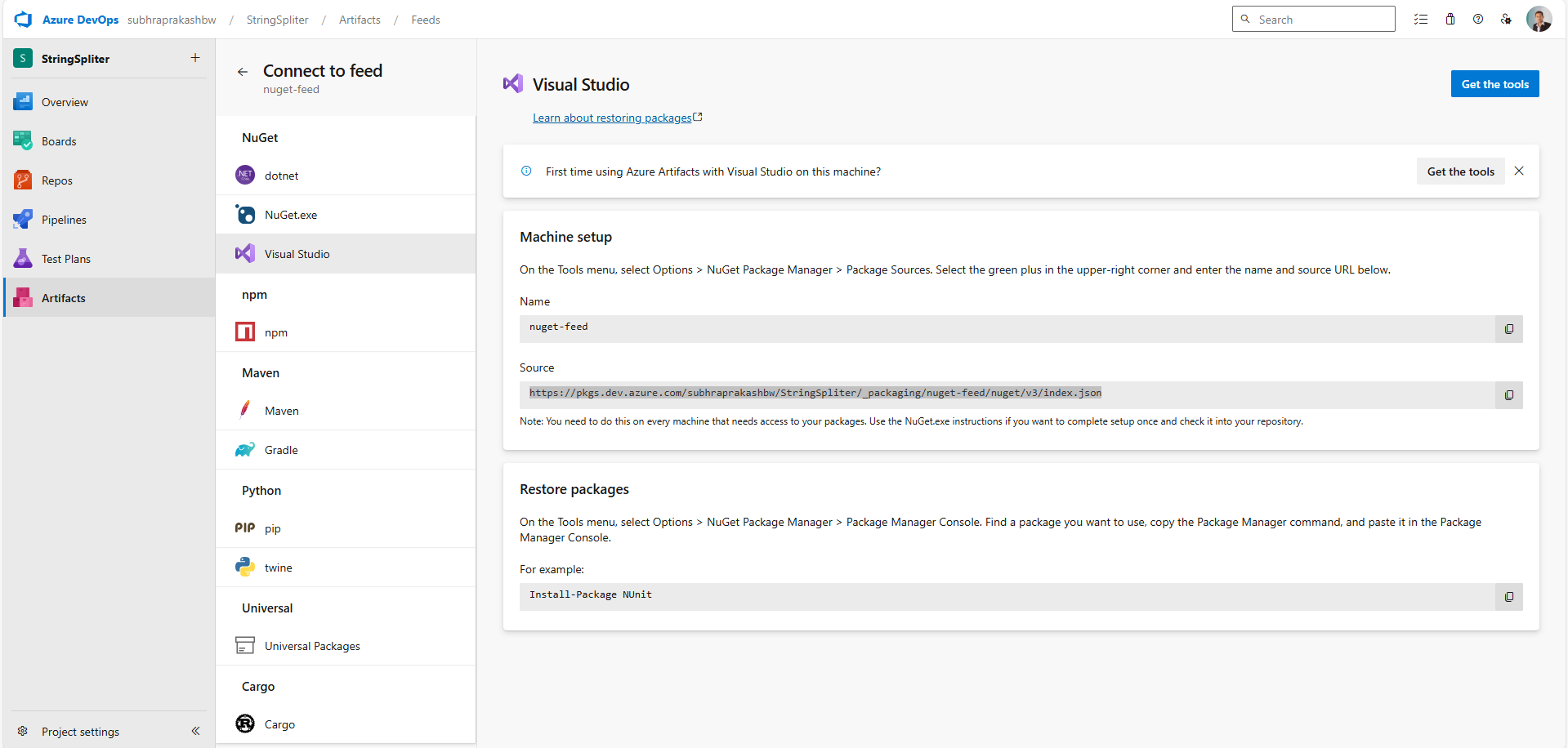


Step 15: Connect to feed

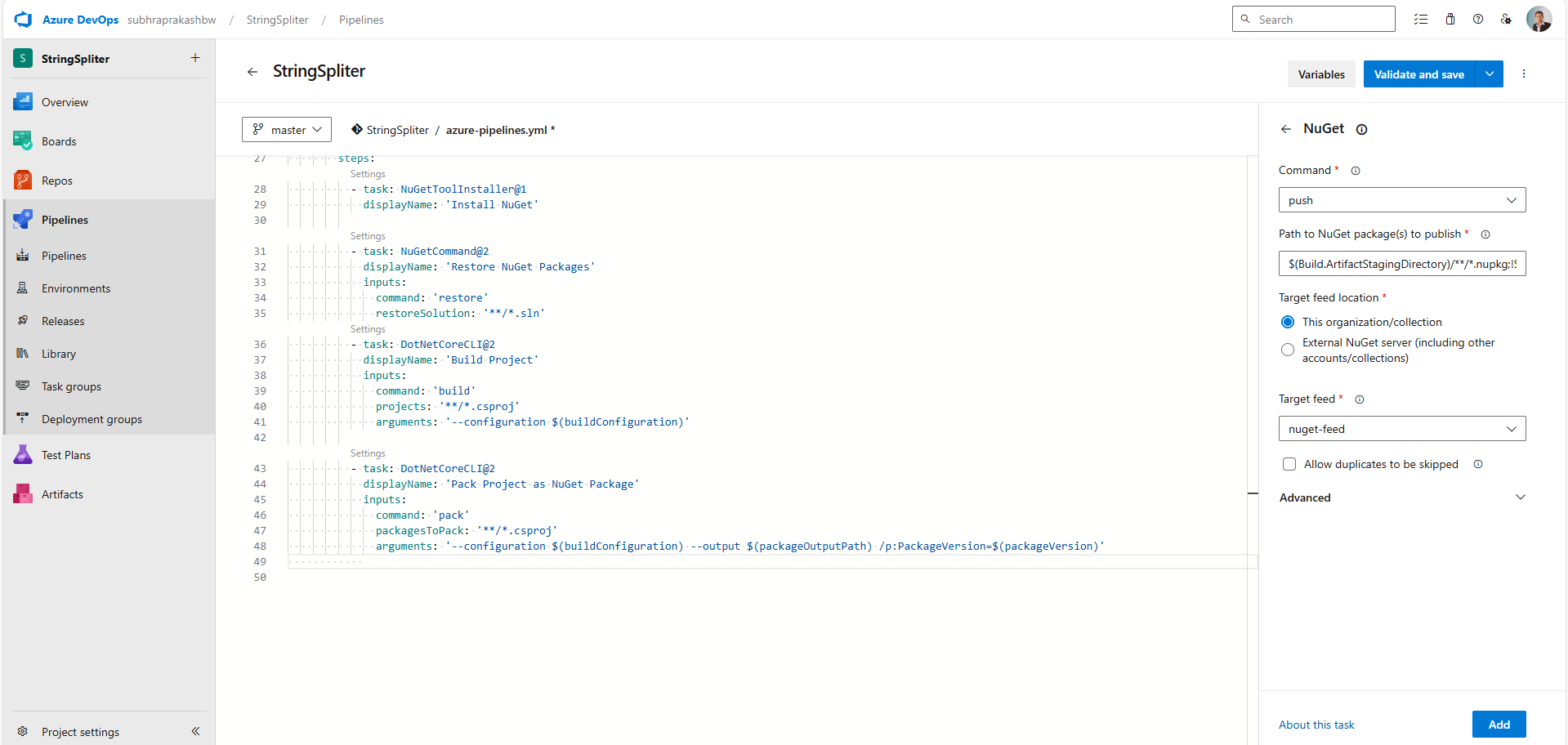


Step 16: Getting machine setup code

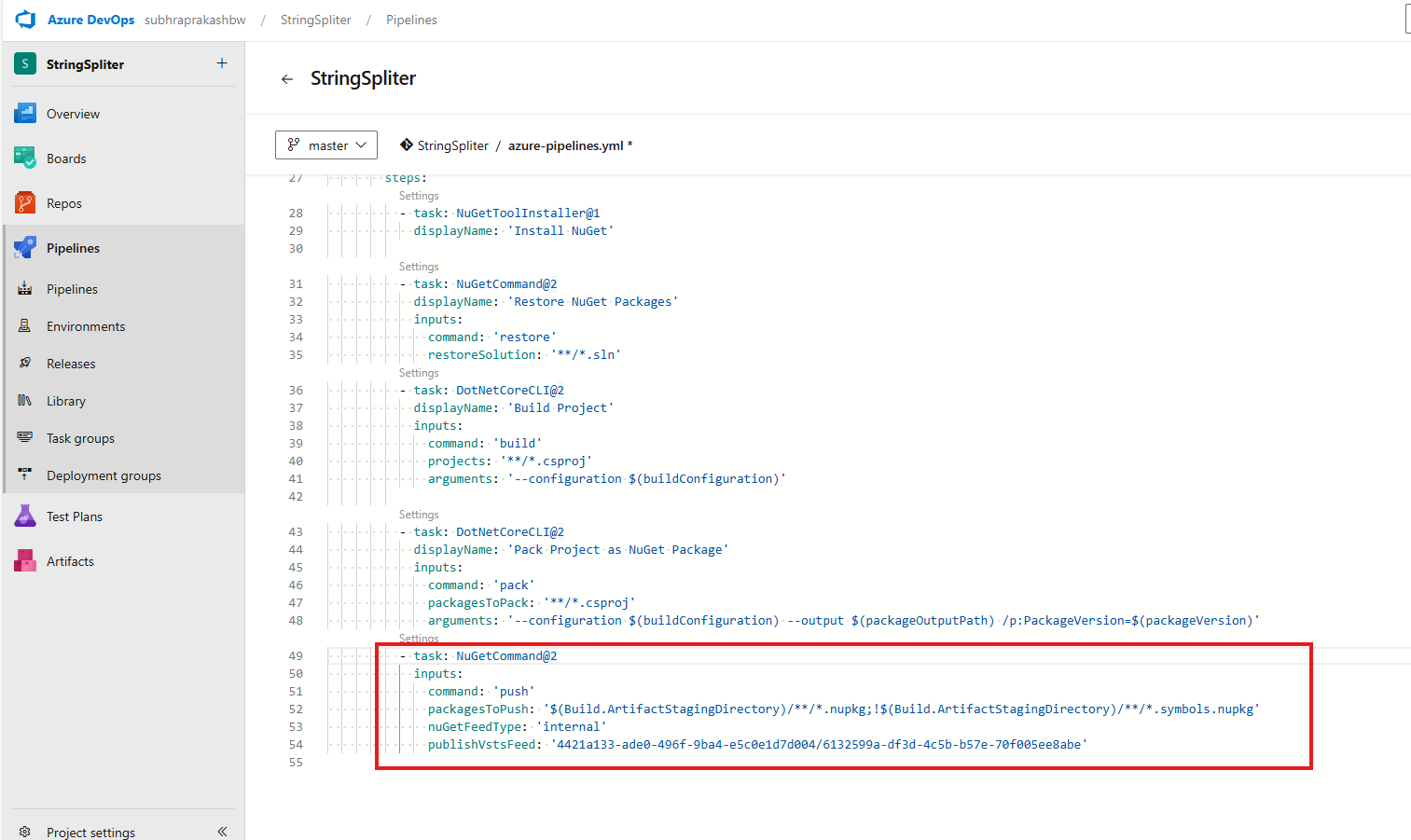
https://pkgs.dev.azure.com/subhraprakashbw/StringSpliter/\_packaging/nuget-feed/nuget/v3/index.json



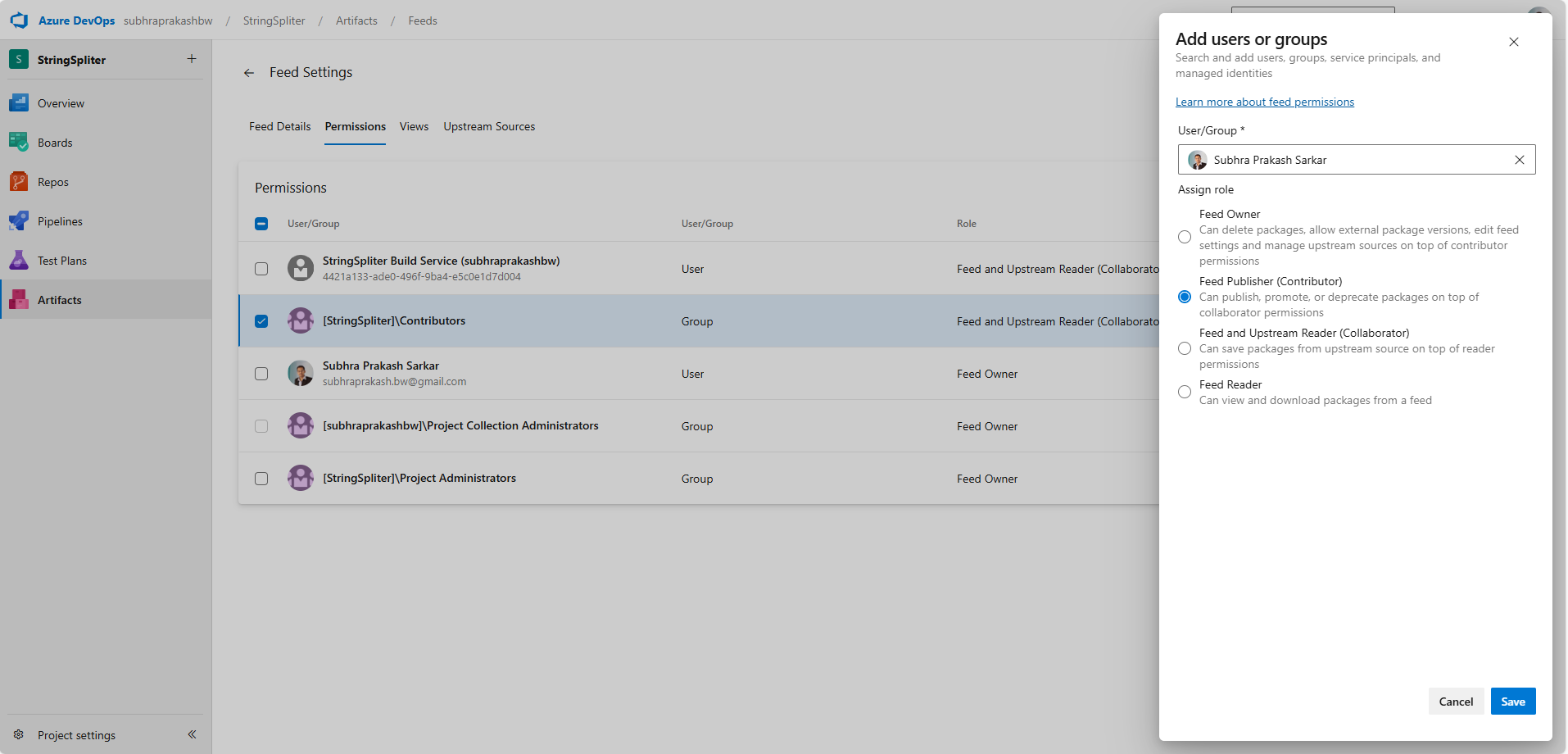
Step 17: adding package publish location



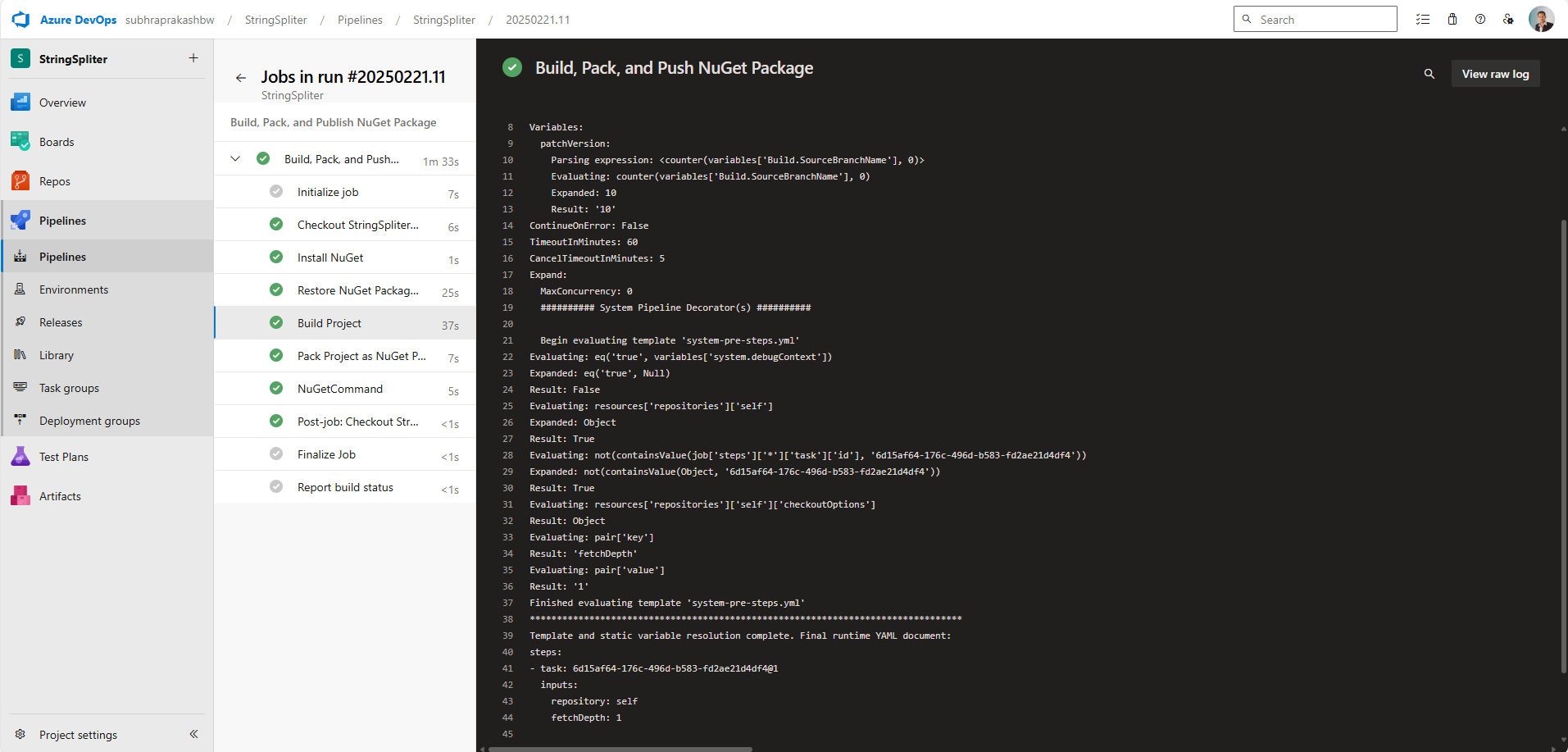
Step 18: added code



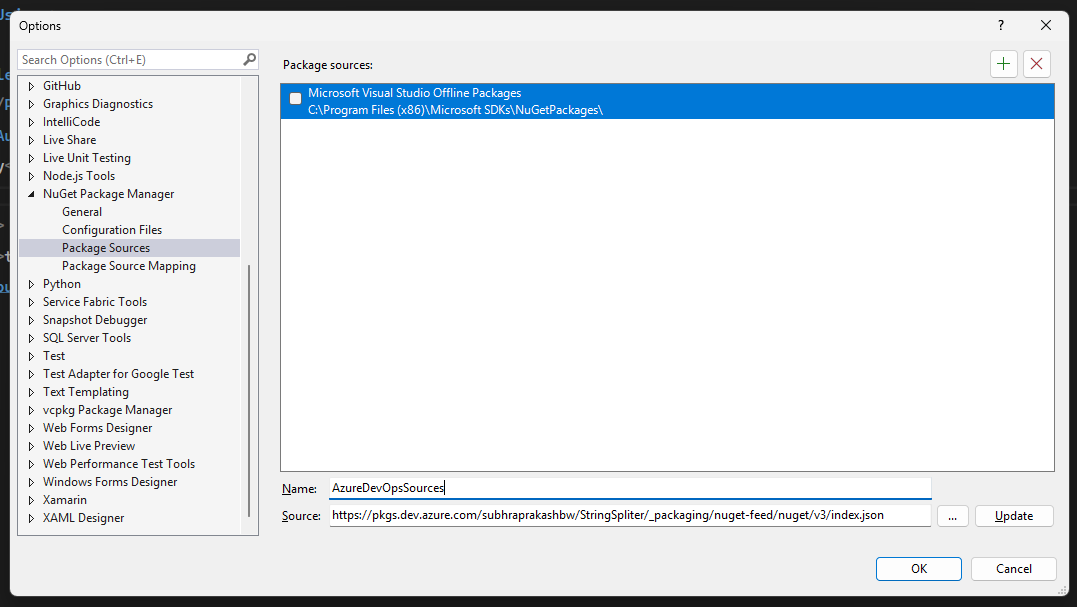
Step 19: need to set permission if add package fails



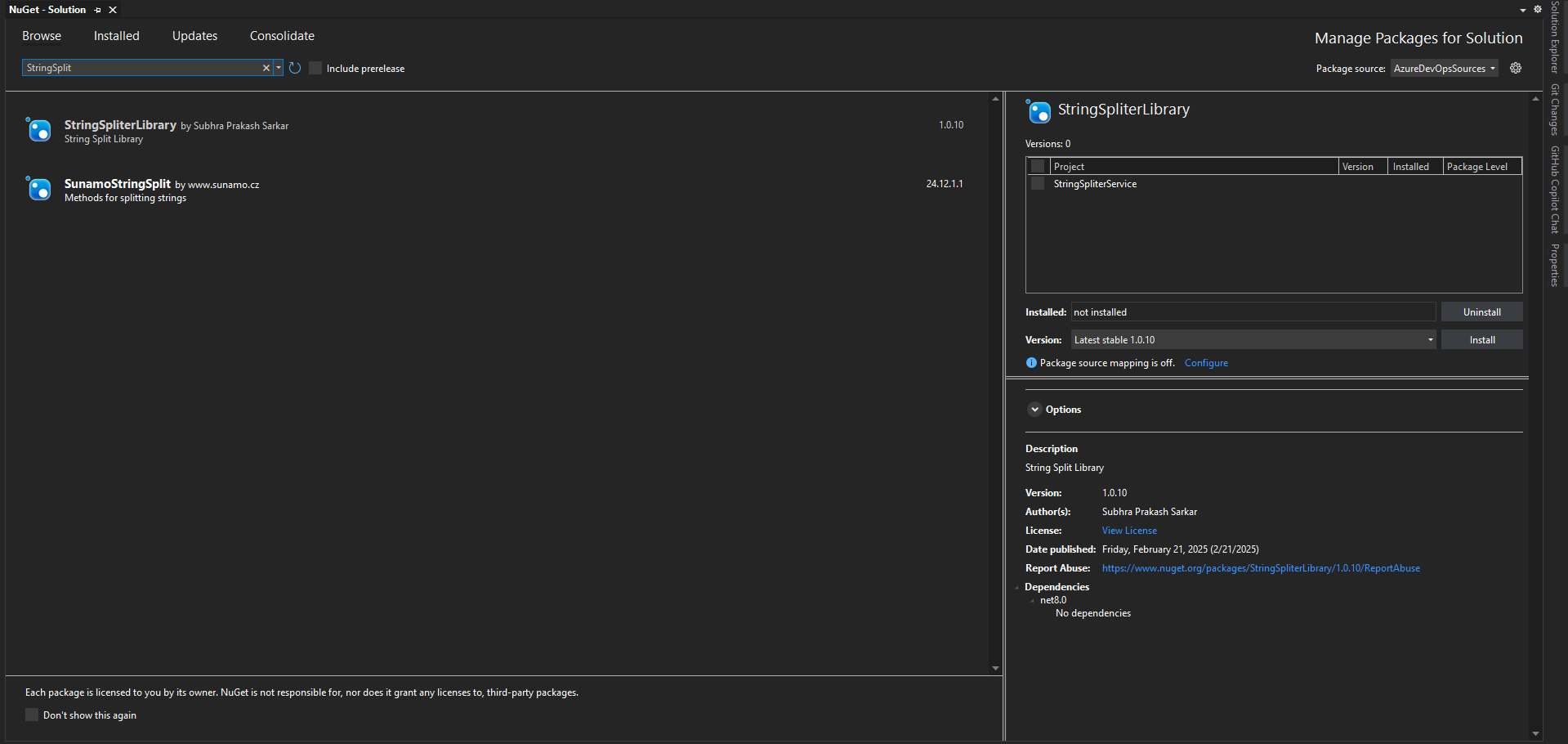
Step 20: Build successful



Step 21: adding nuget package in Visual Studio



Step 22:



# CI/CD Workflow

## 1. CI/CD Pipeline Overview

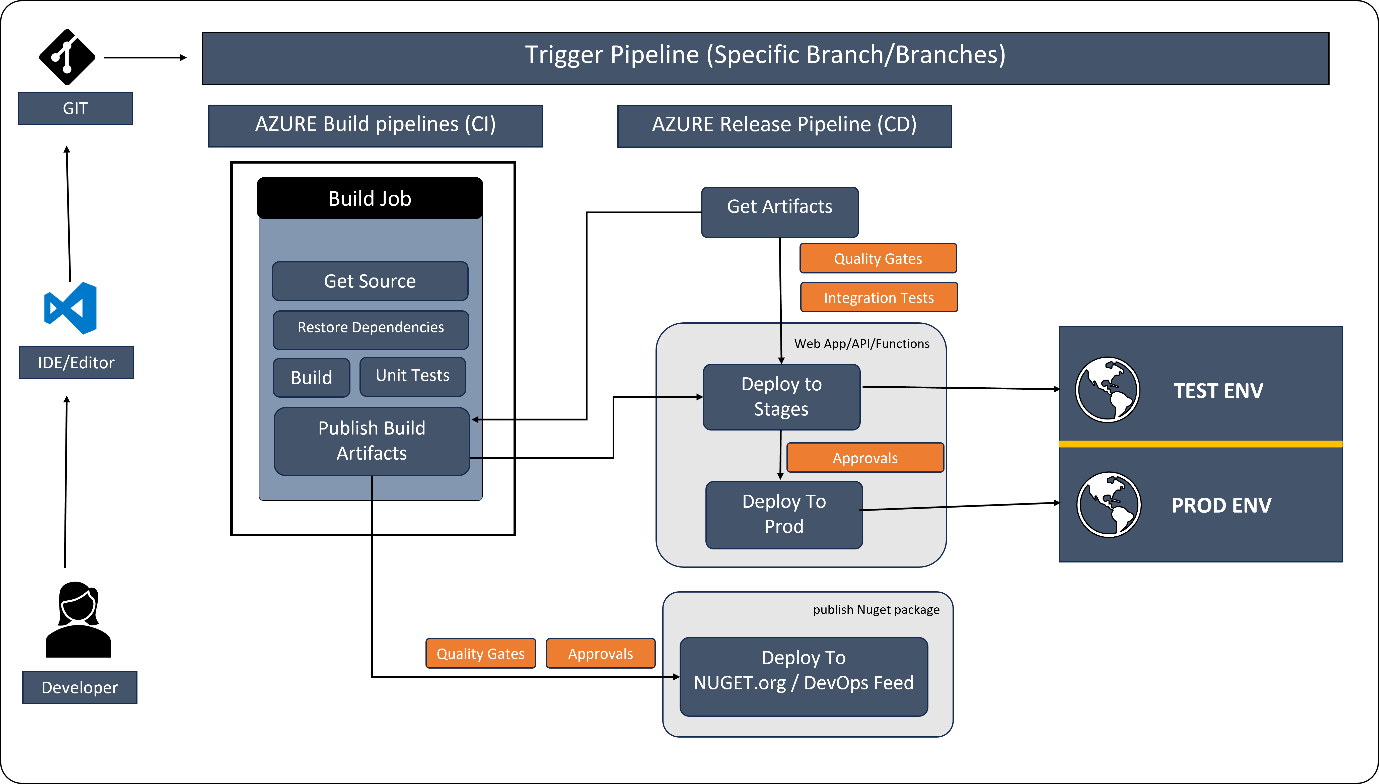
The CI/CD pipeline in **Azure DevOps** consists of the following steps:

### 1️ Continuous Integration (CI) - Build Pipeline

1. **Code Commit**: Developers push code to the **Azure Repos (Git)**.
2. **Trigger Build Pipeline**: The pipeline starts automatically on new commits.
3. **Restore Dependencies**: Install required packages (e.g., using NuGet, npm, or Maven).
4. **Build**: Compile the application code.
5. **Run Unit Tests**: Execute automated tests to ensure code quality.
6. **Publish Build Artifacts**: If the build is successful, package and store the output for deployment.

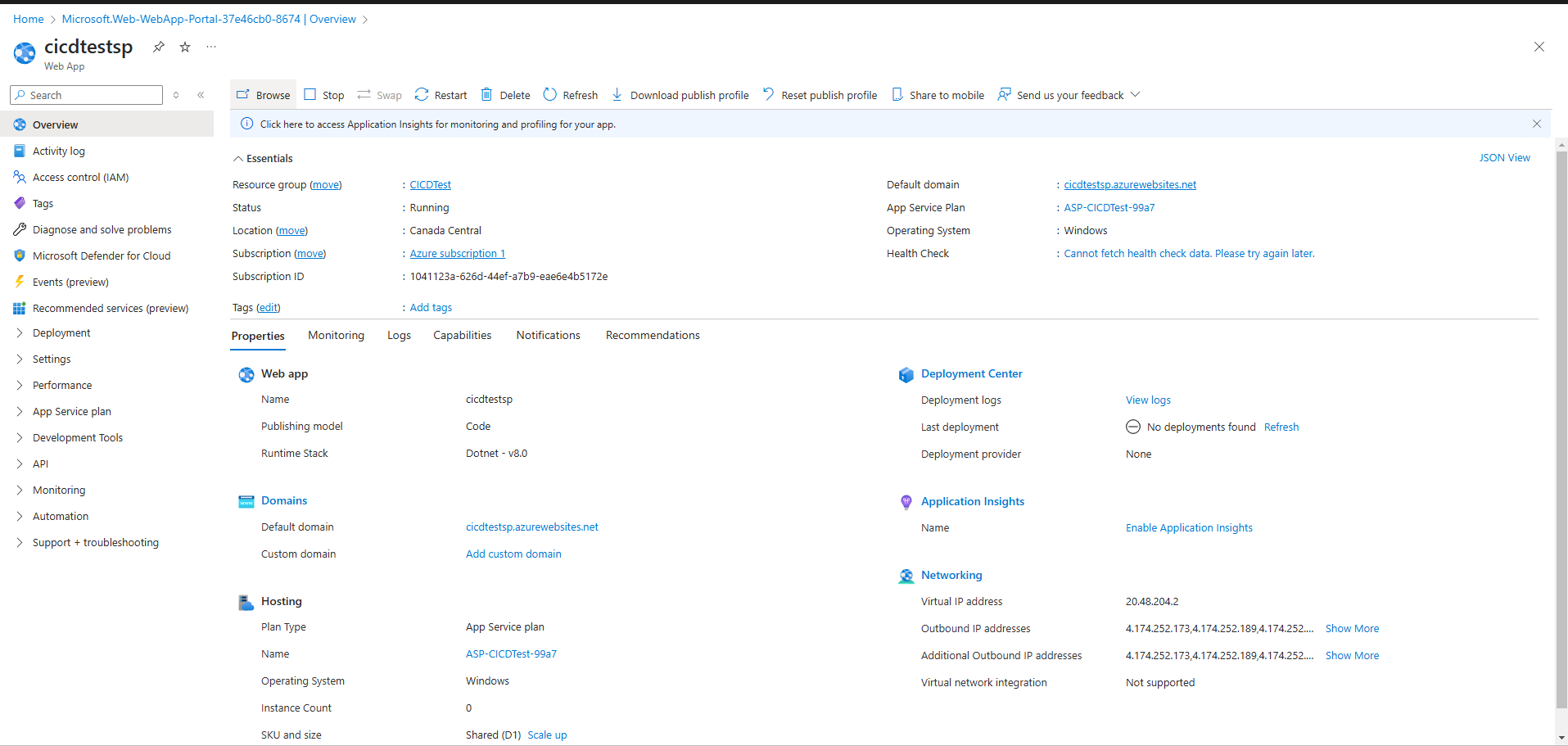
### 2️ Continuous Deployment (CD) - Release Pipeline

1. **Trigger Release Pipeline**: Deploys the build artifact from the CI pipeline.
2. **Deploy to Staging**: The application is first deployed to a **staging** environment.
3. **Run Integration Tests**: Automated tests validate the deployment.
4. **Approval Process**: Manual approval (if required) before production deployment.
5. **Deploy to Production**: The final tested build is deployed to production.
6. **Monitoring & Feedback**: Application performance and errors are monitored.

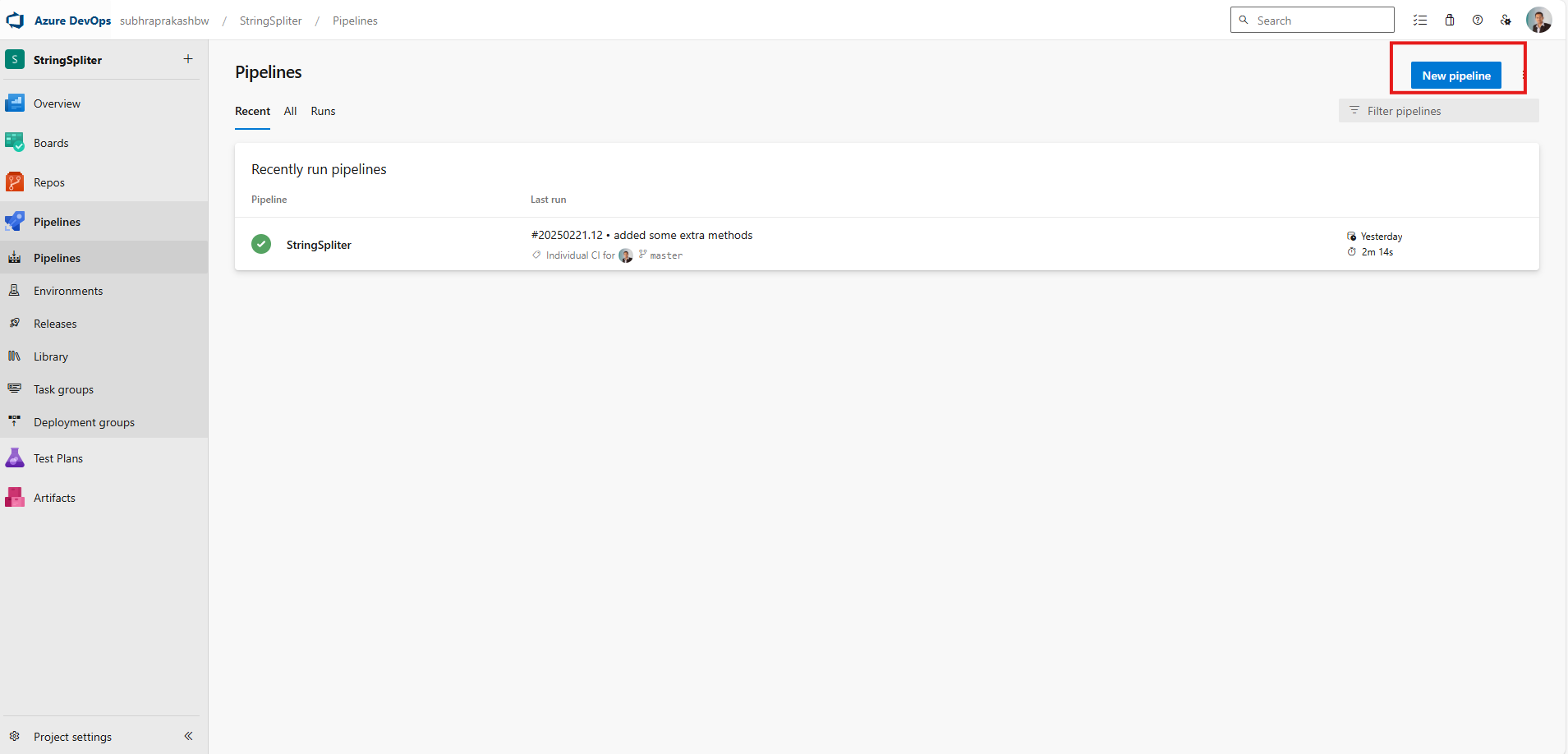


# Hosting web app

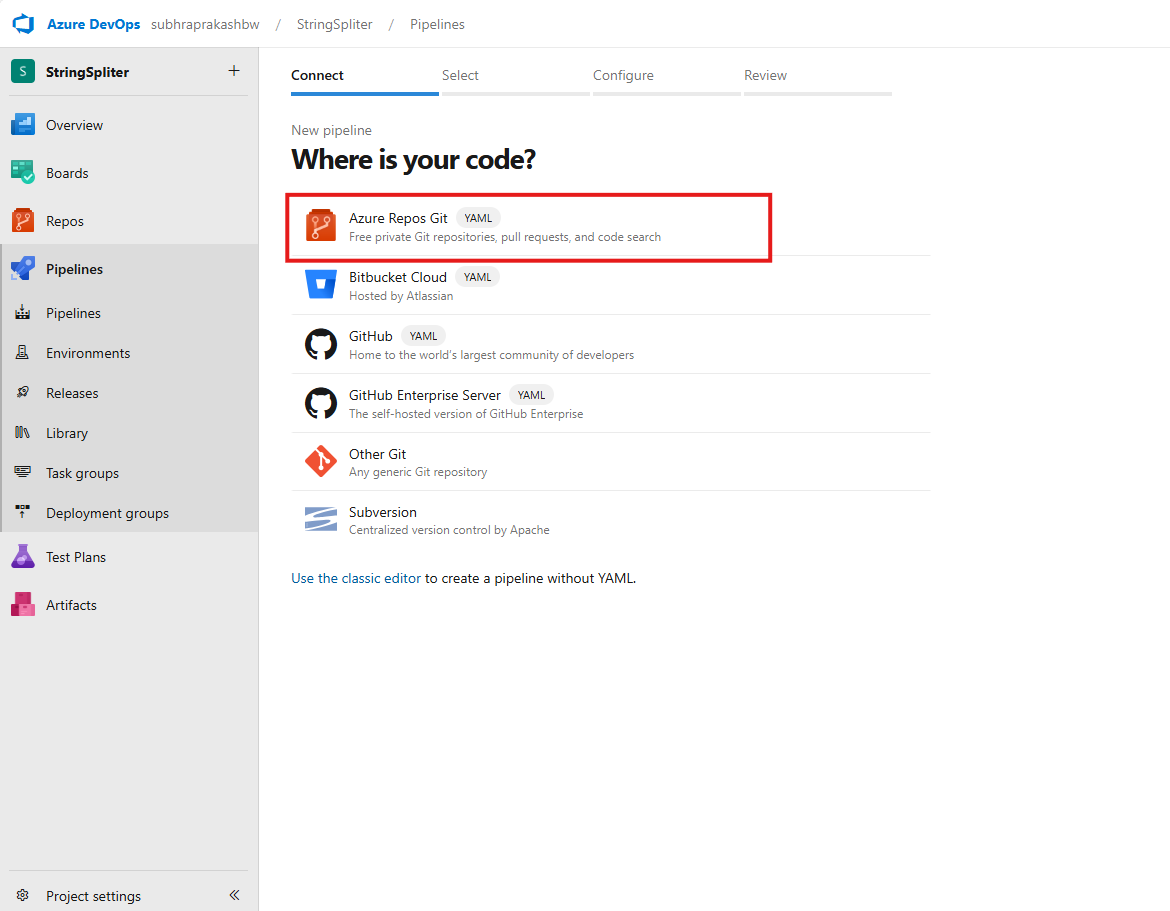
Step 1: Create App service



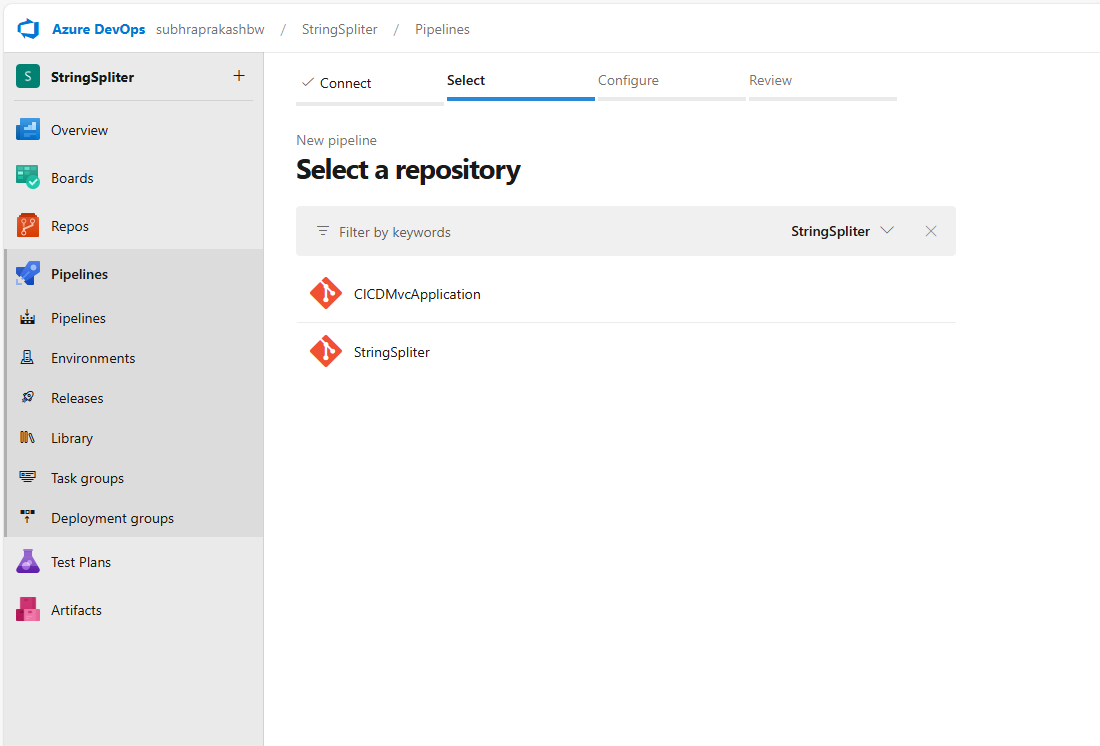
Step 2: Creating a new pipeline



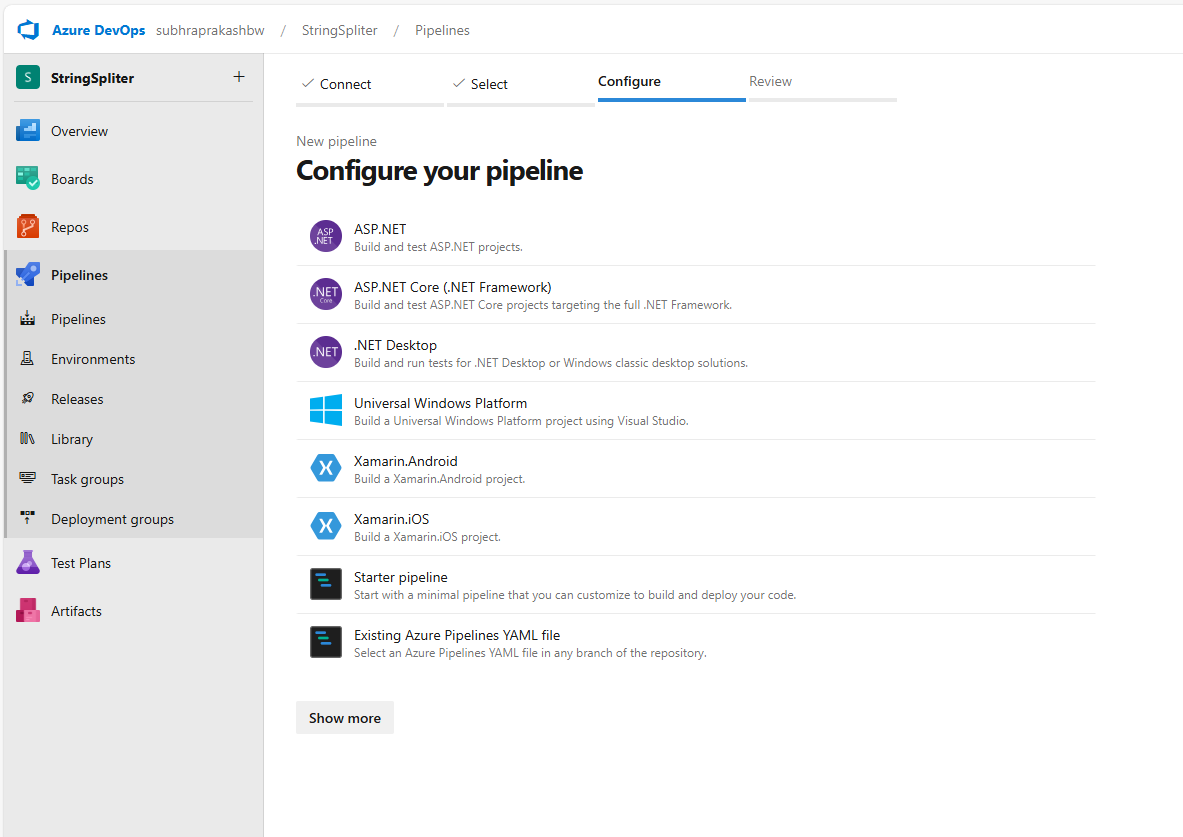
Step 3:



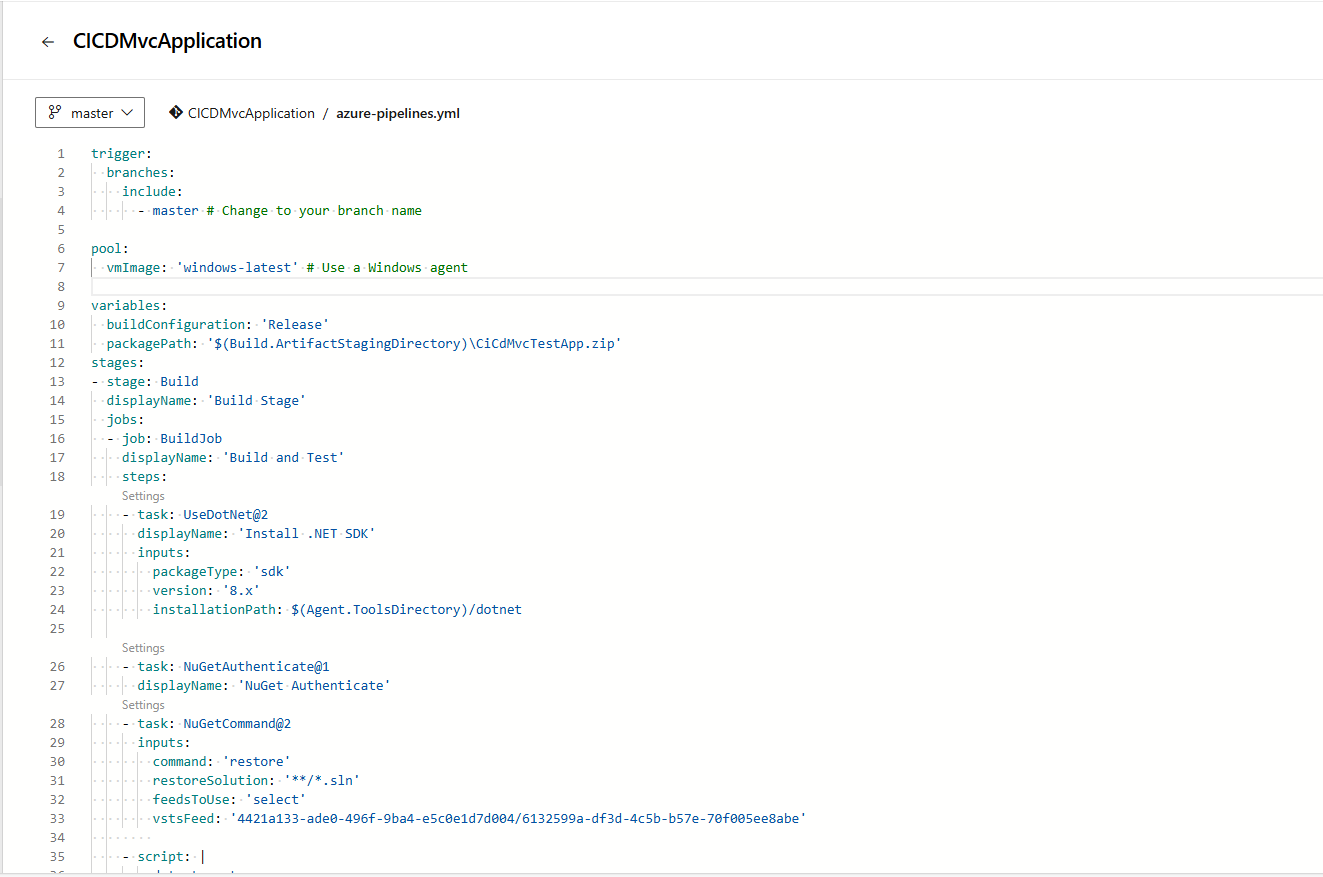
Step 4:



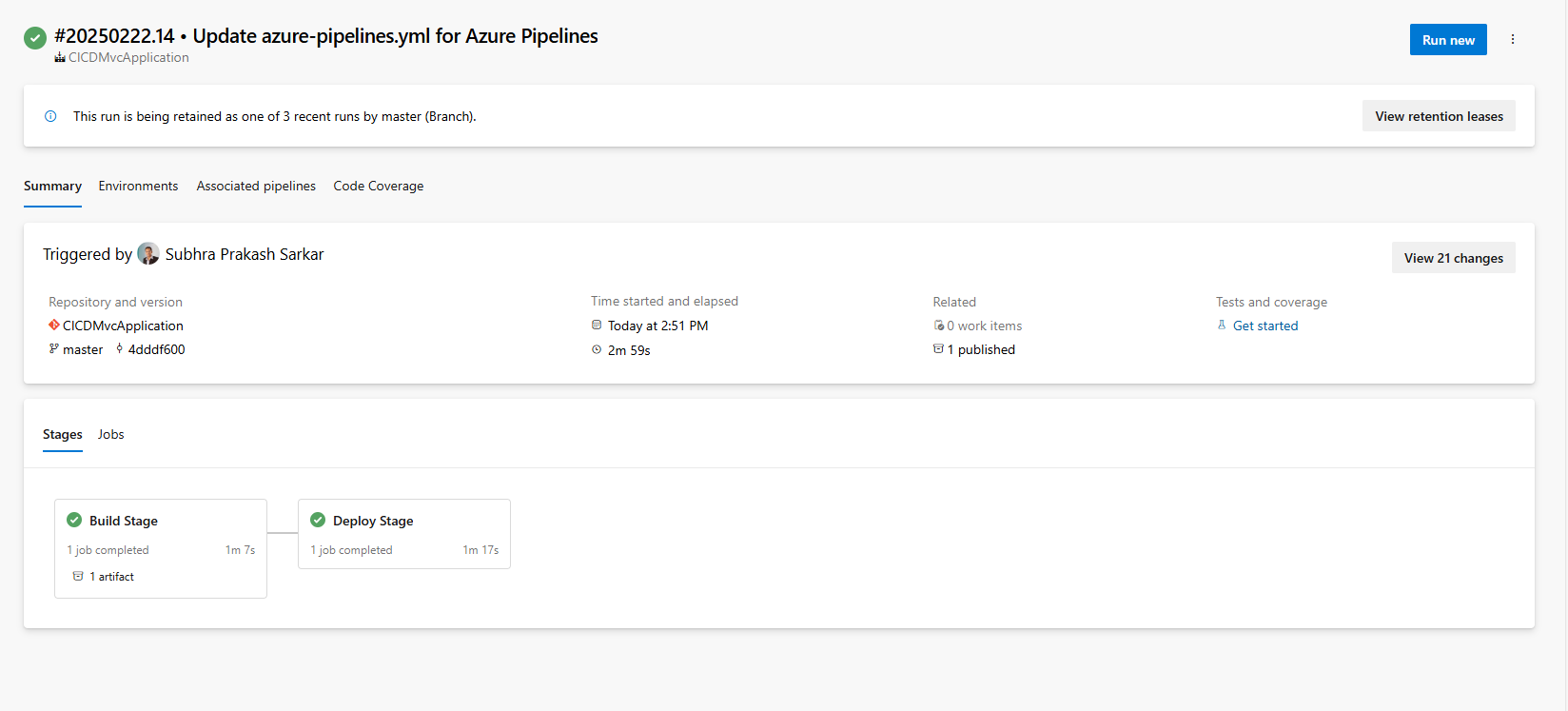
Step 5:



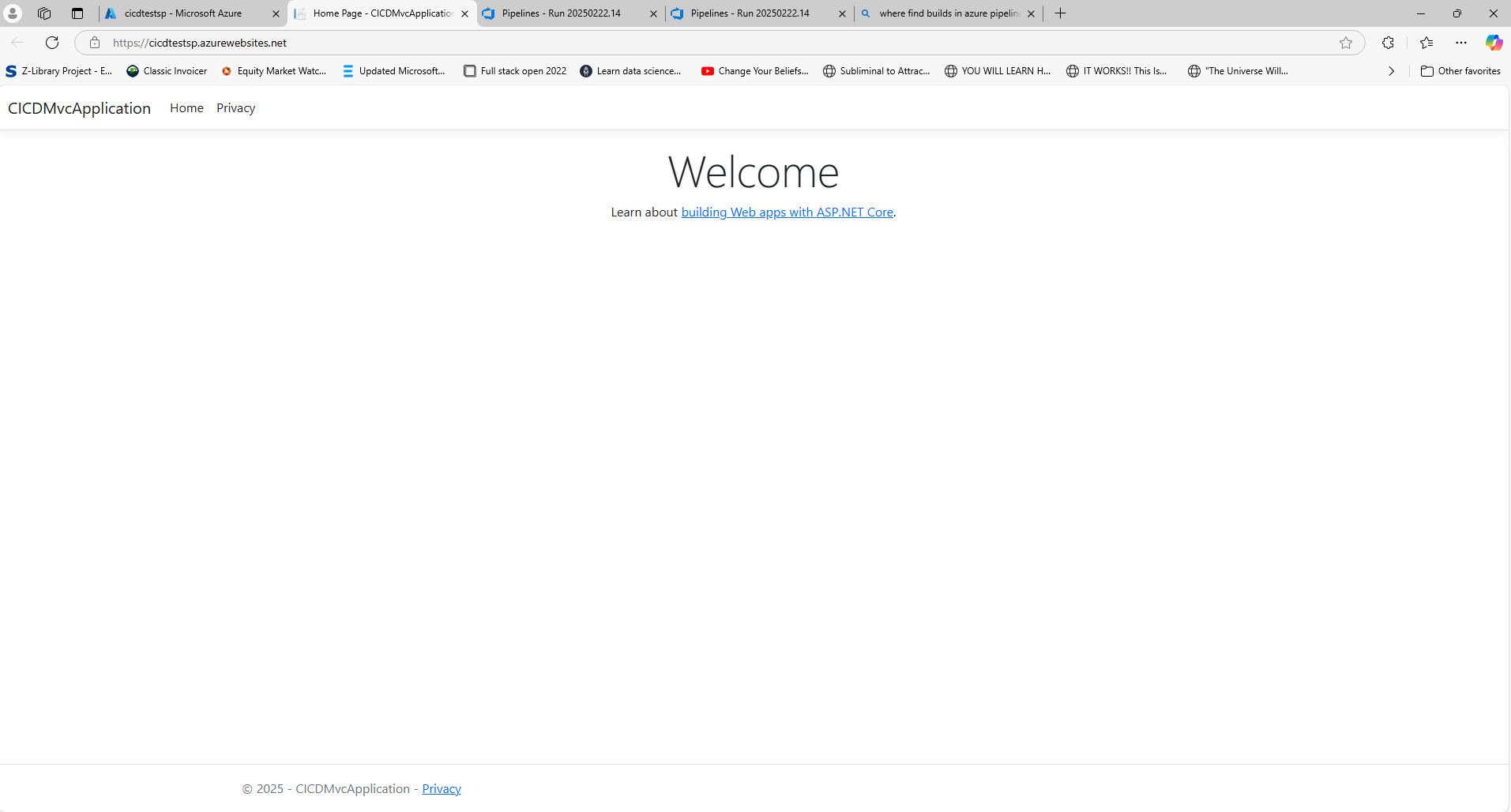
Step 6:



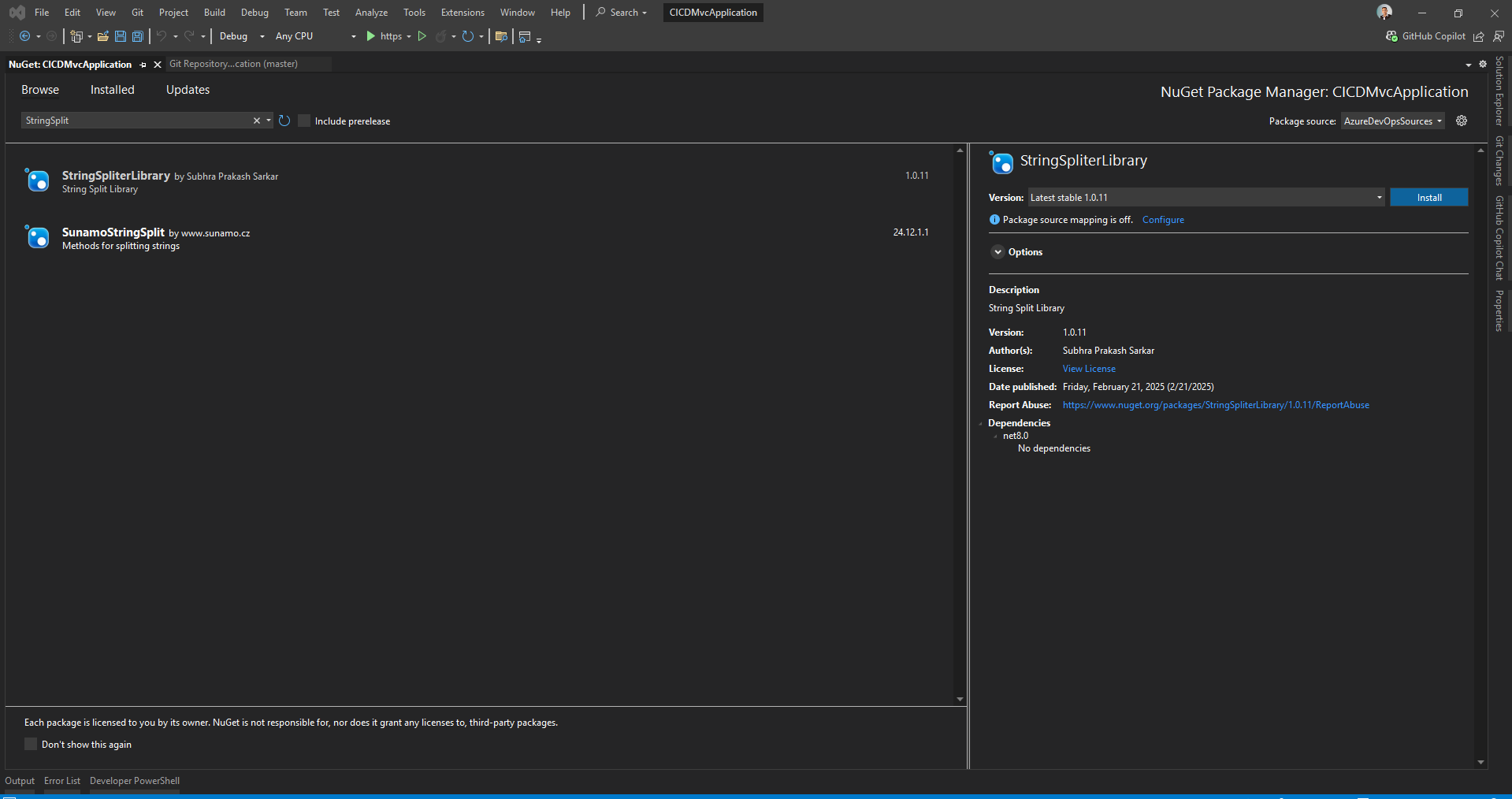
Step 7:



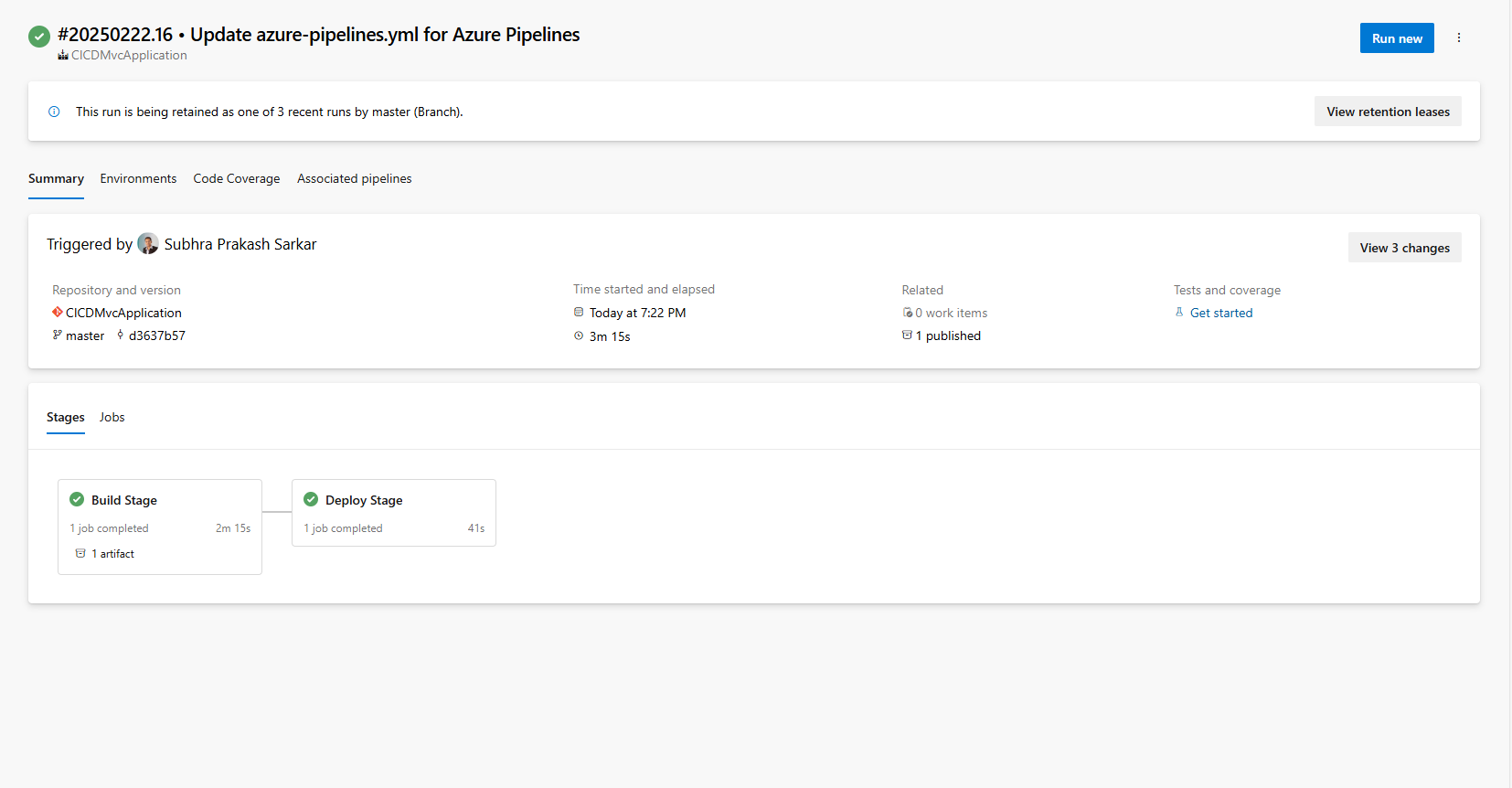
Step 8:

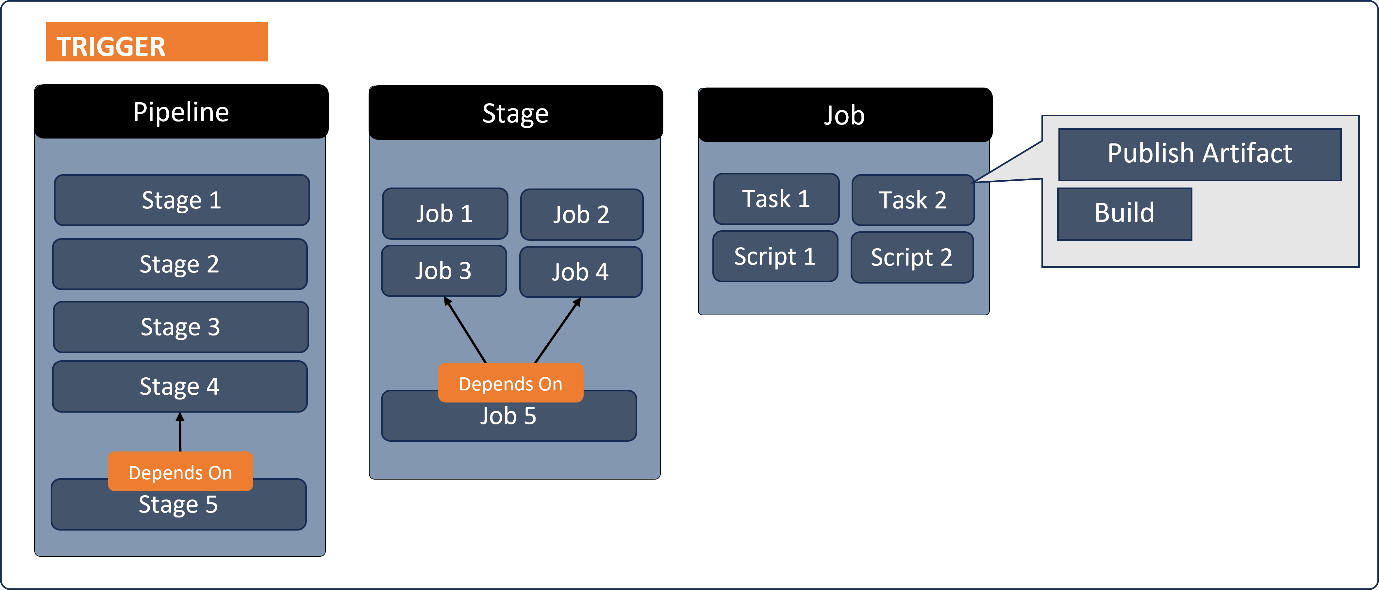


Step 9:



Step 10:





**1. Stage**

* A **stage** represents a major phase in the CI/CD pipeline (e.g., Build, Test, Deploy).
* It groups multiple jobs that need to be executed together.
* Stages can run sequentially or in parallel.

**2. Job**

* A **job** is a collection of tasks that run sequentially within a stage.
* Jobs help in organizing related steps in a logical manner.
* A stage can have **multiple jobs**, and each job can run on a different agent.

**3.a. Task**

* A **task** is a predefined action (e.g., restore dependencies, build an app, run tests, deploy).
* Azure DevOps provides built-in tasks like UseDotNet, NuGetCommand, ArchiveFiles, etc.

**3.b. Script**

* The **script** section allows running custom shell commands (PowerShell, Bash, or CMD).
* It is often used for executing .NET CLI commands, testing, or publishing.