**Links:**

YAML File Link: <https://github.com/167281/MS1Buildpipe.git>

File Name - Test Yaml File.yaml & azure-pipelines.yml

Pipeline Link: <https://dev.azure.com/ramyarajagopalan/ramyarajagopalan/_build>

Repo Link (With Project files): <https://github.com/167281/MS1Buildpipe.git>

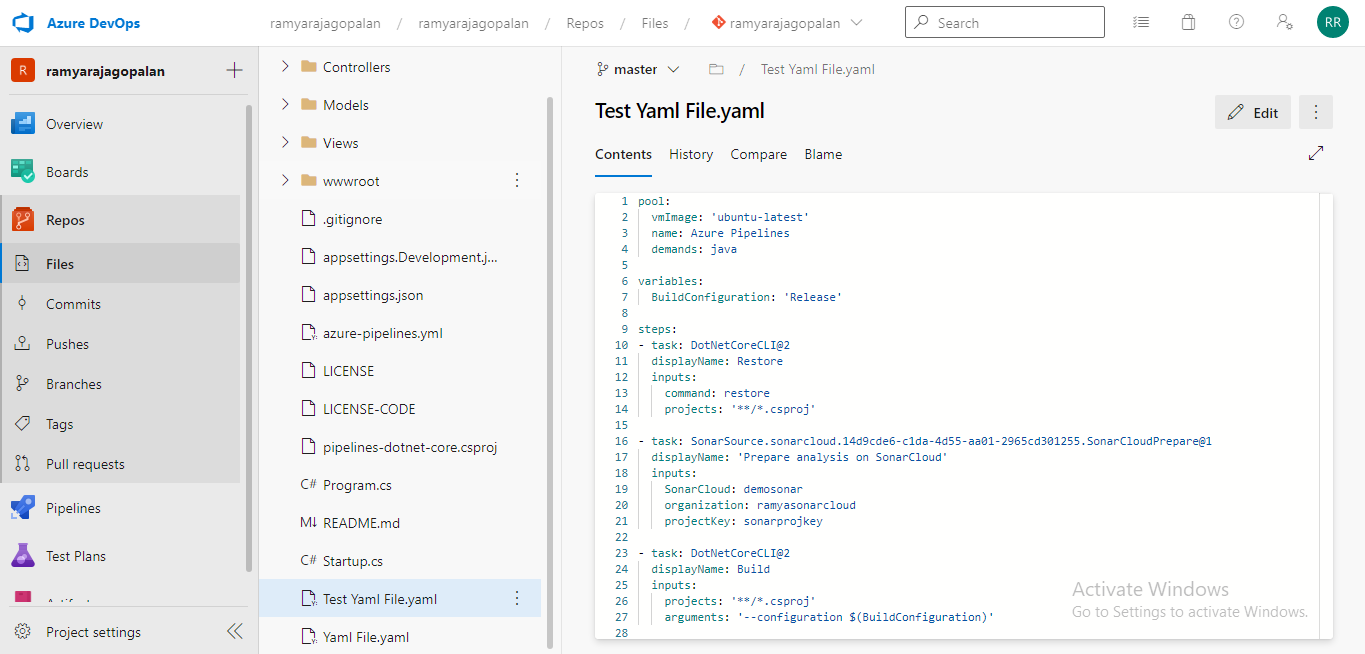
Document Link: <https://github.com/167281/MS1Buildpipe.git>

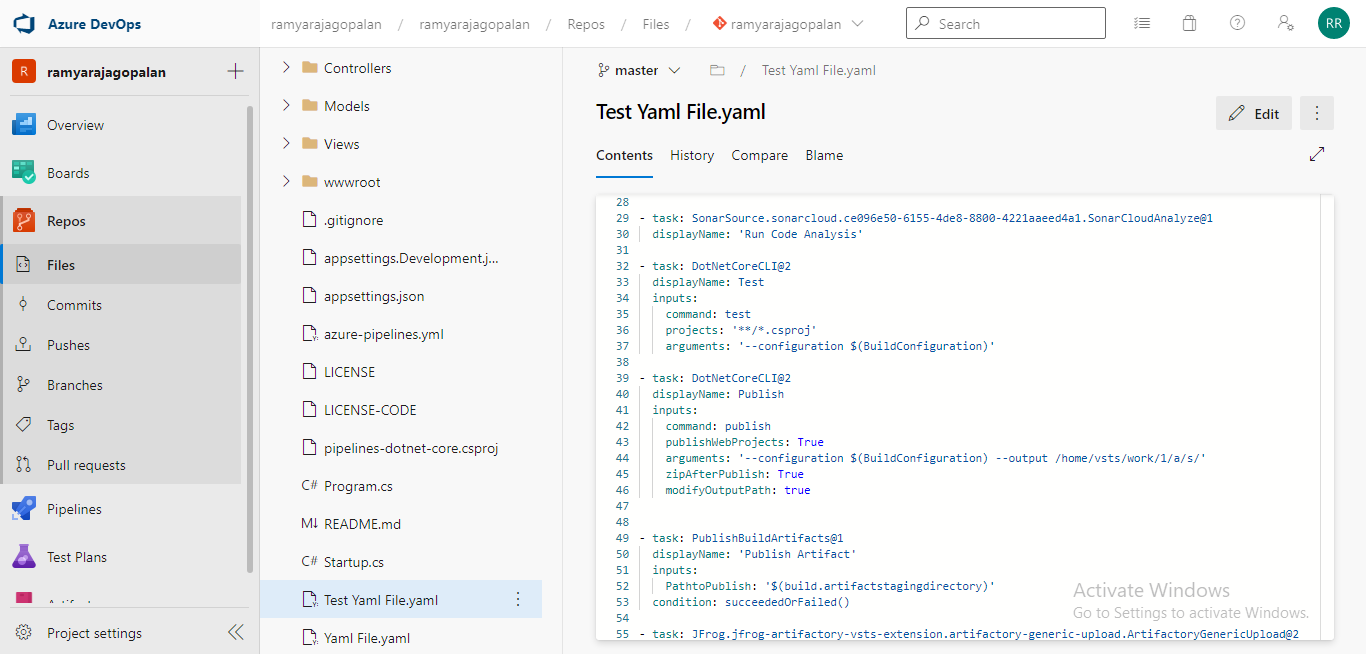
**Note:**

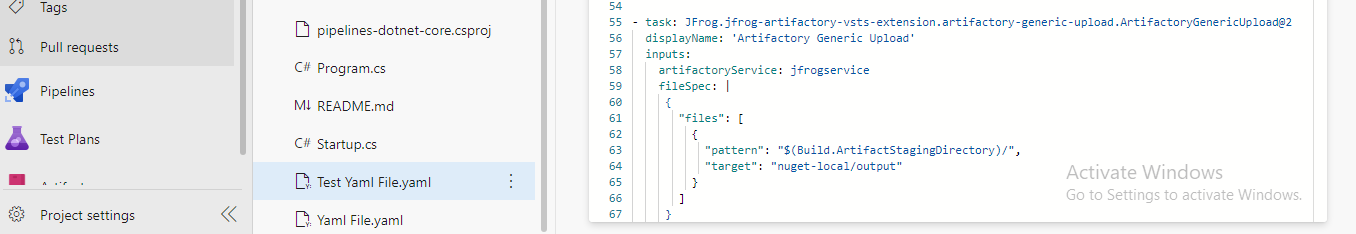
A .NET Core project was used (found in Github repo link) for the test cases. Github and Azure repo is integrated and used while creating pipelines

**Use Case 1: Create YAML based CI-CD pipeline in Azure DevOps**

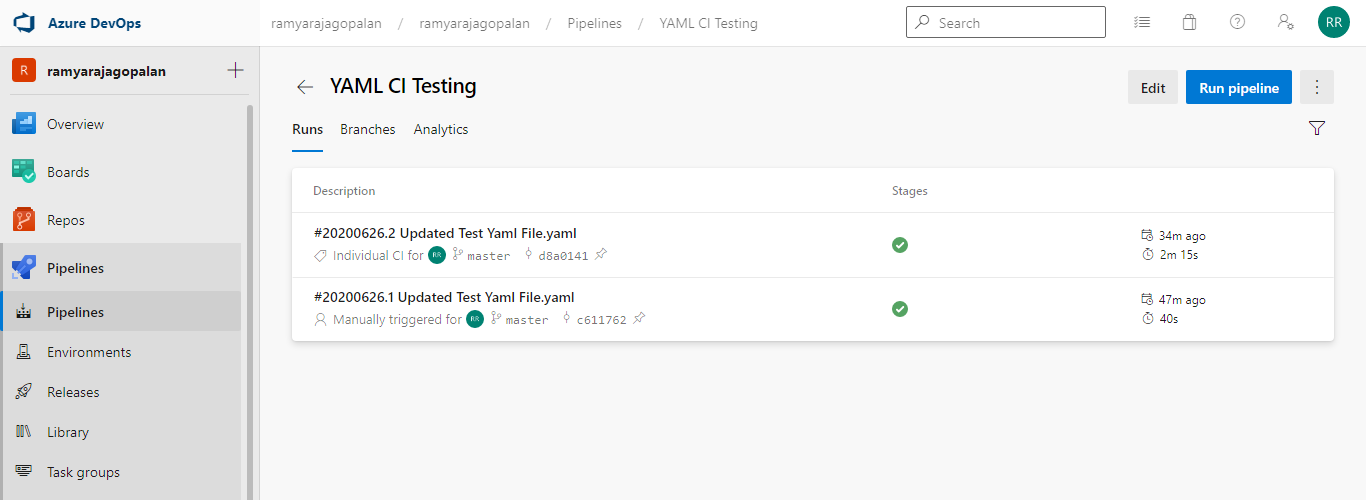
YAML File used – CI (Build Pipeline):



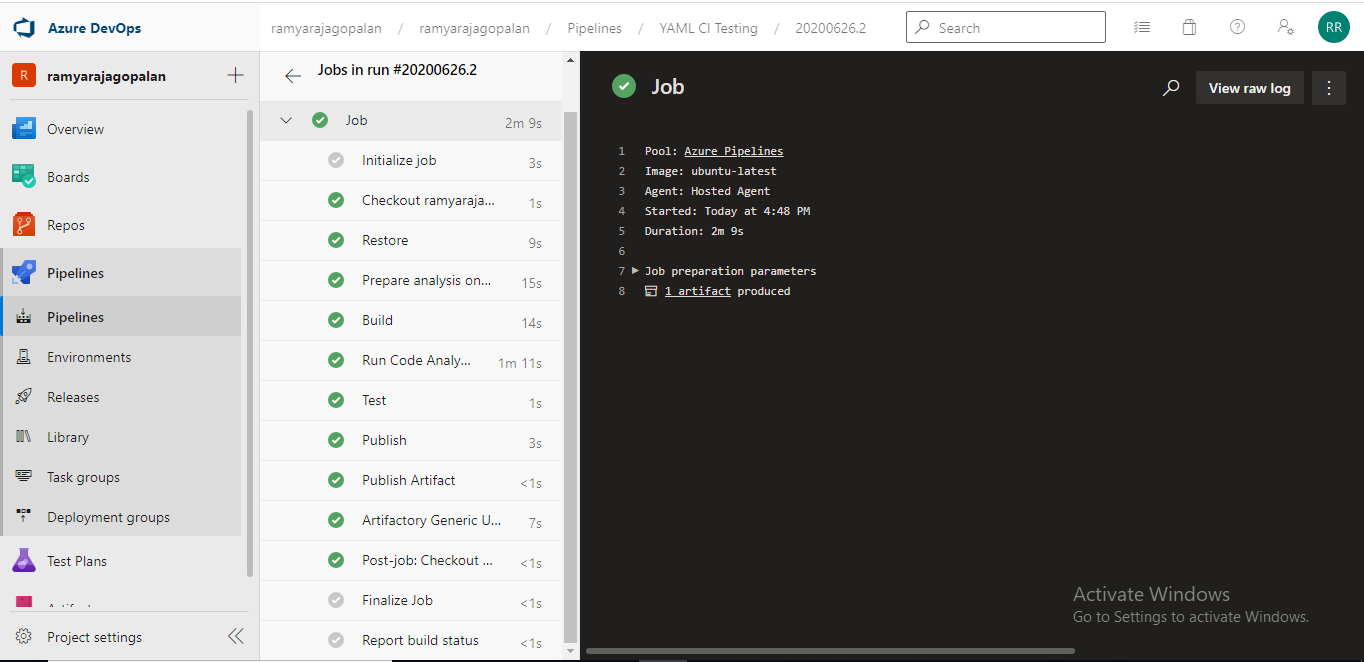




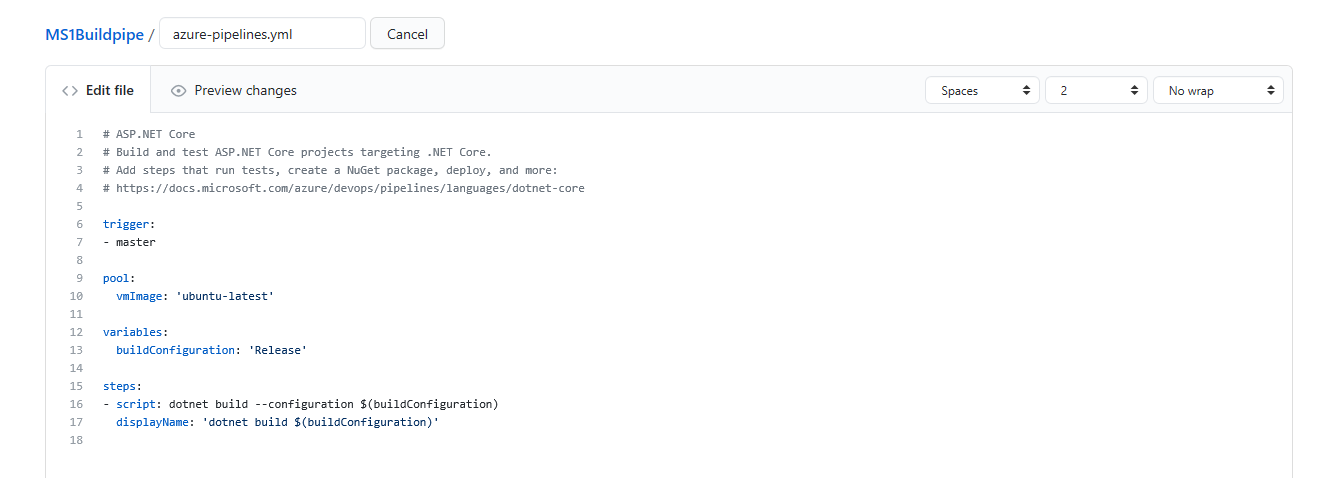
Build Pipeline using YAML file:



Ran successfully:

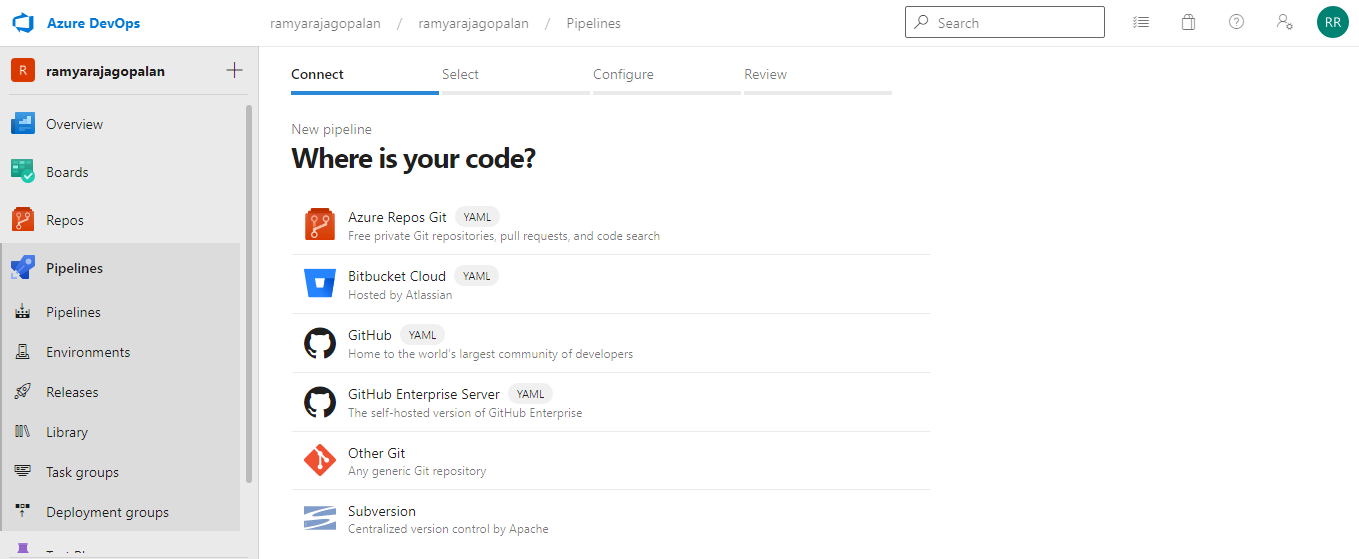


YAML File used – CD(Release Release Pipeline):

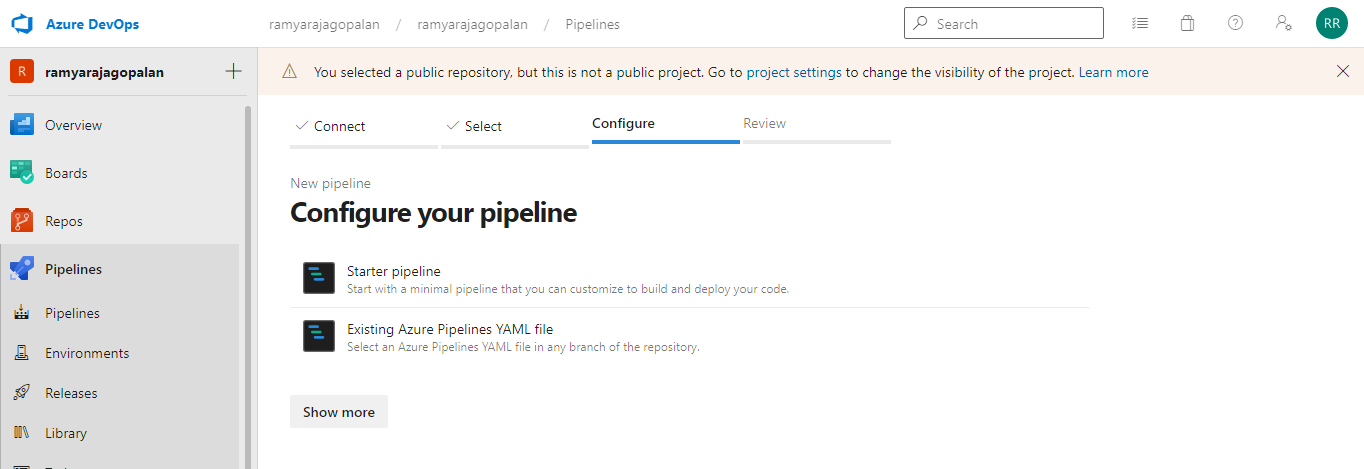


Release Pipeline using YAML file:

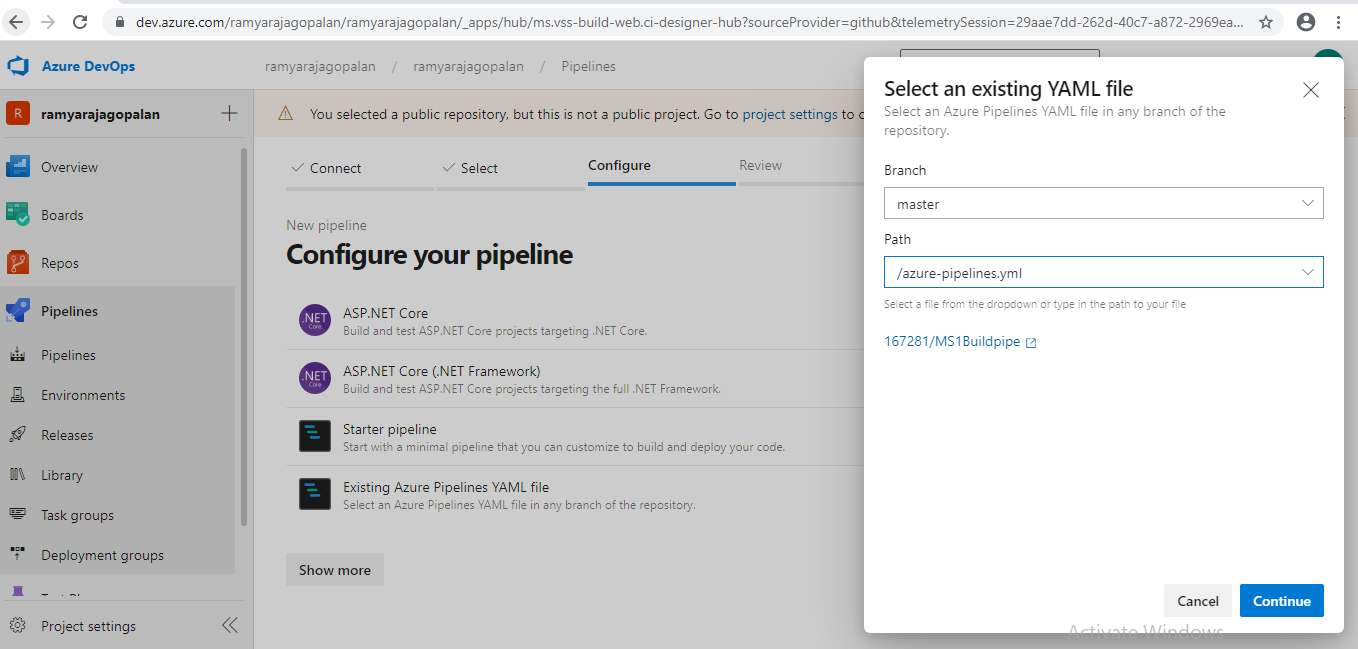
Code in Github:



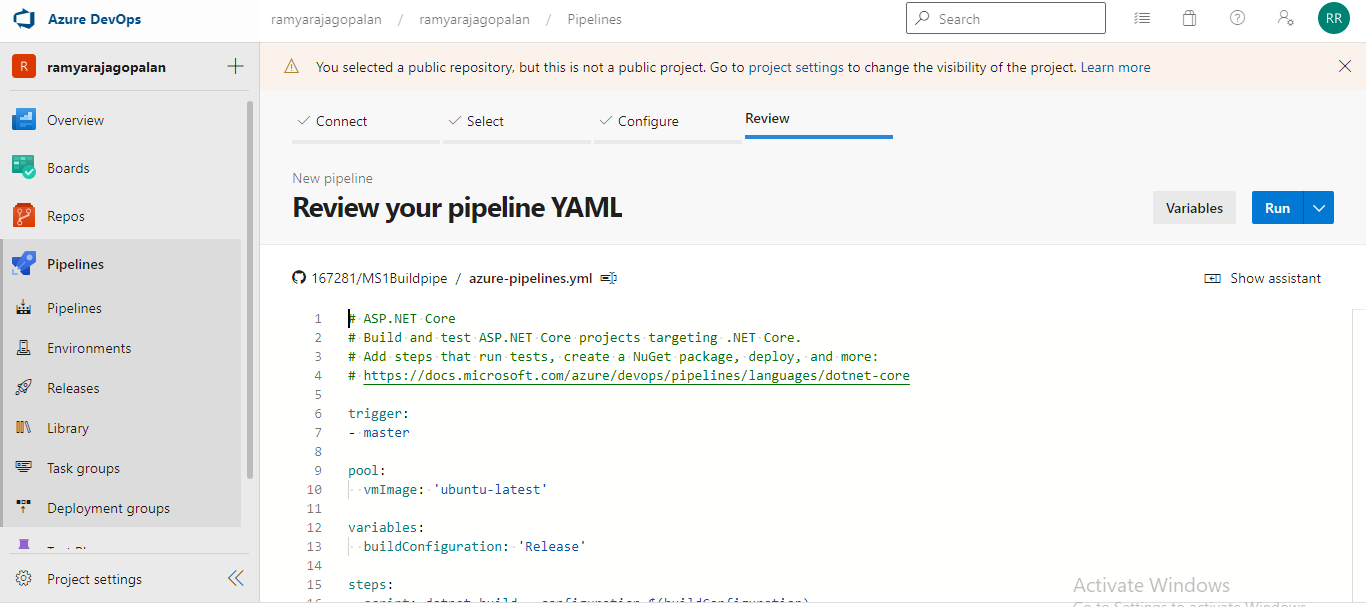
After configuring with Github and choosing relevant repos:



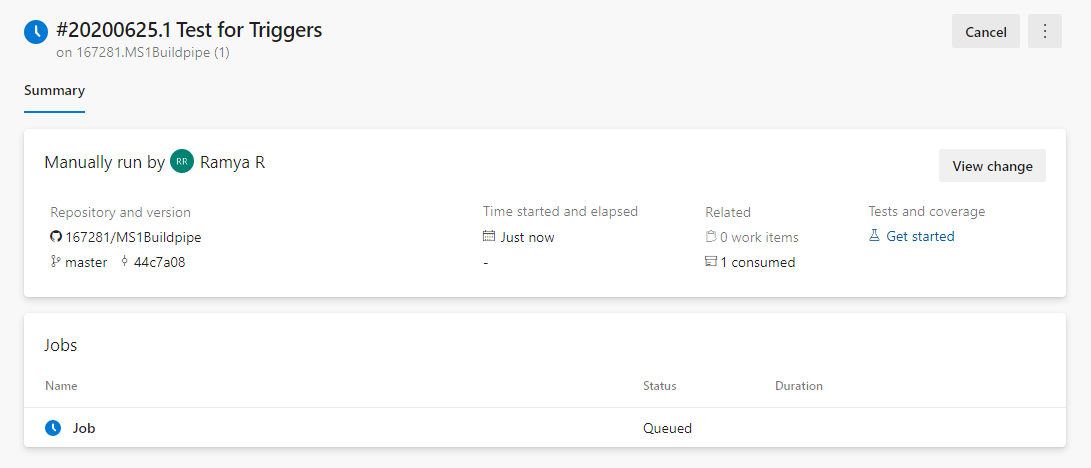
Select YAML file from repo:



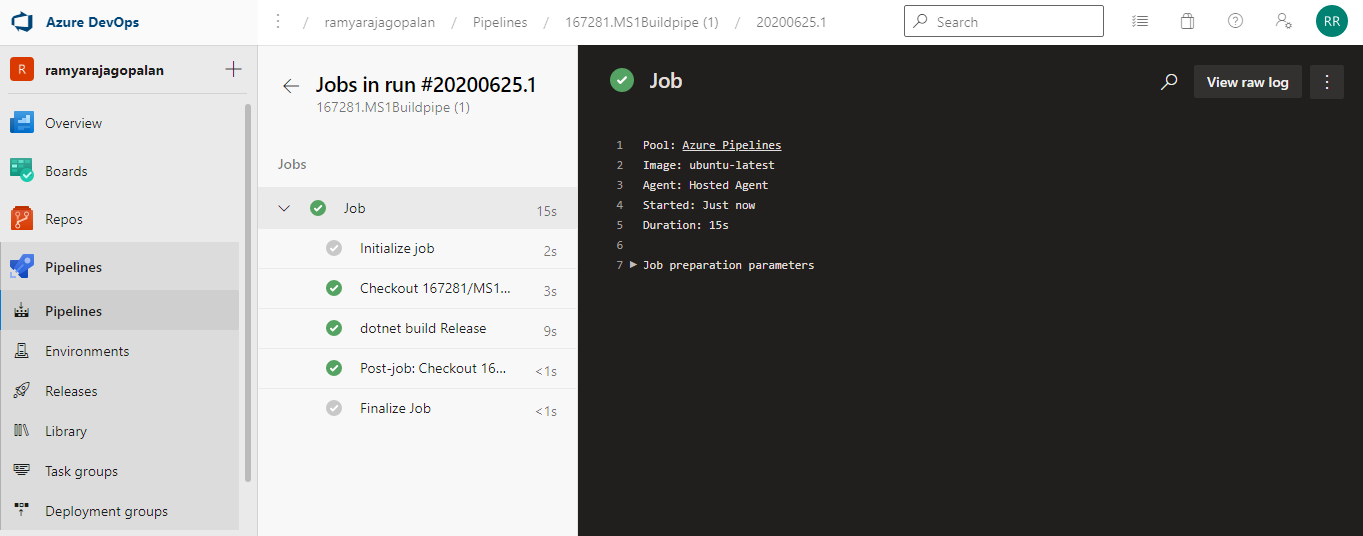
Review and run the pipeline based on our sample YAML file:



Job triggered by YAML file:



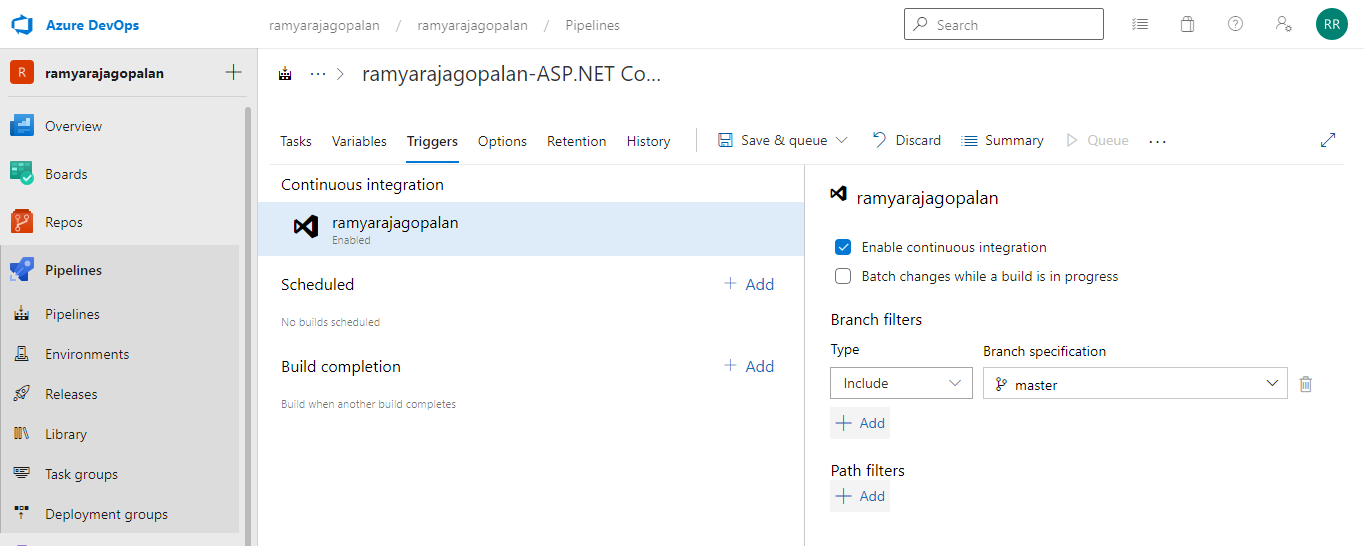
Job completed successfully:



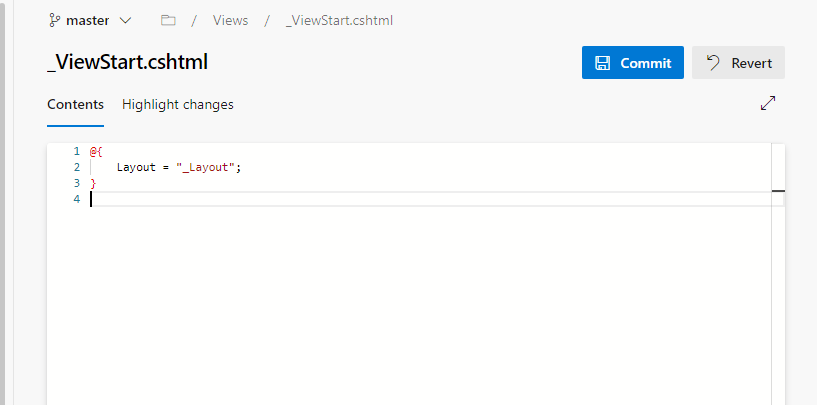
**Use Case 2: Trigger CI-CD pipeline on every code commit to repository and deploy app to Dev region**

**Trigger build pipeline based on code commit in Git:**

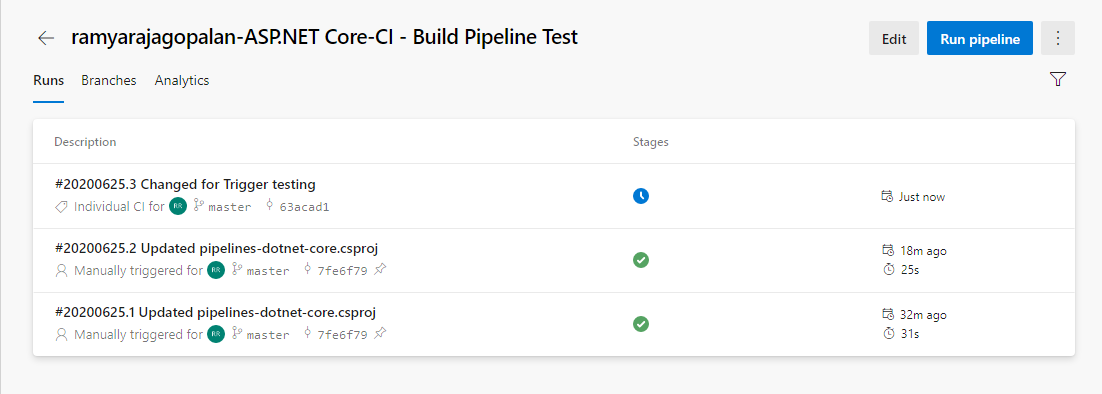
Added trigger to Build pipeline by enabling Continuous integration check box:



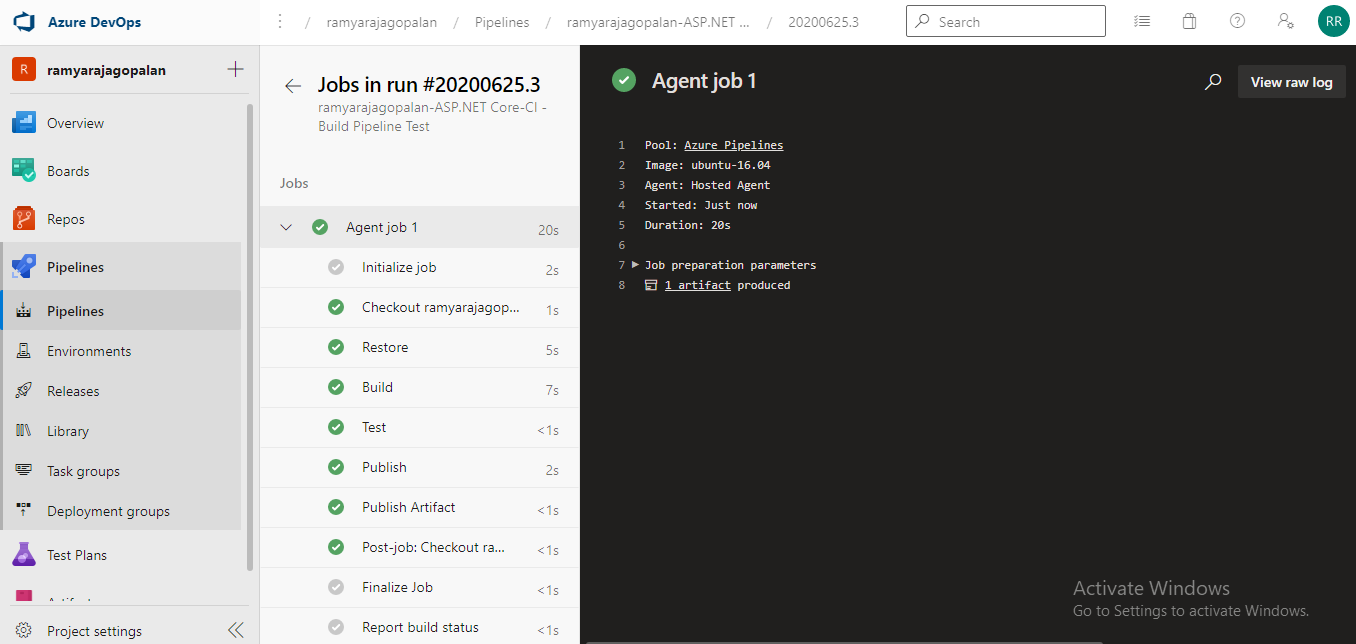
Edit and commit Changes:



Build pipeline triggered automatically:

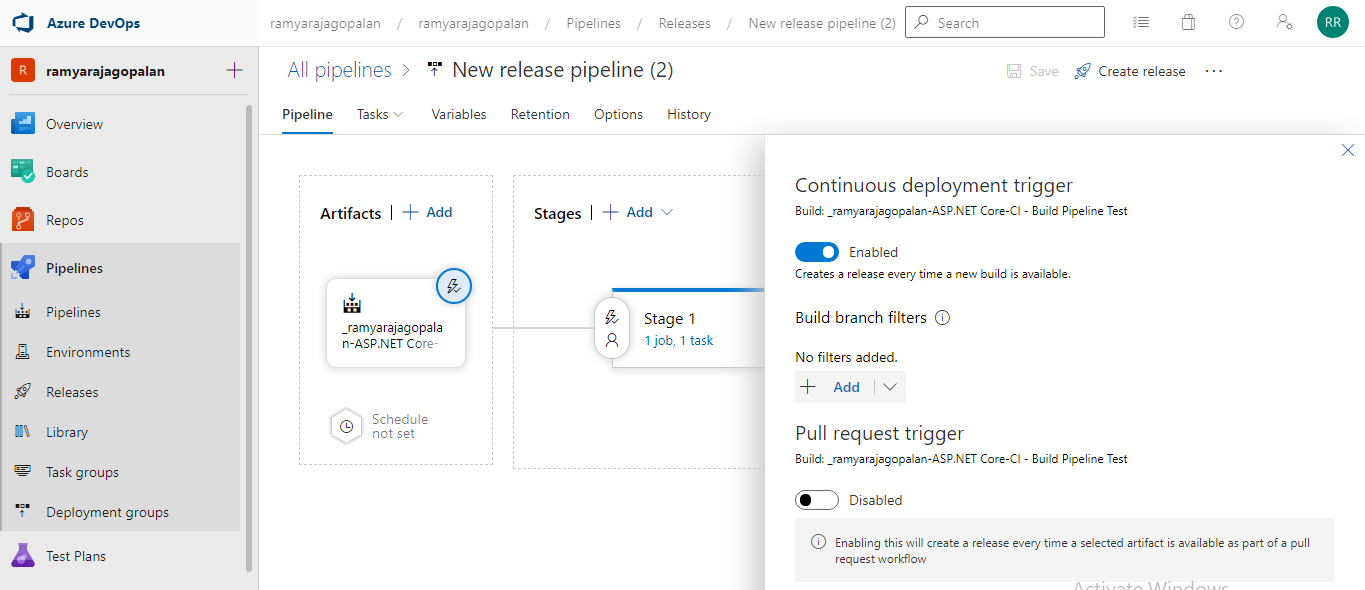


Job Completed successfully and artifact produced:

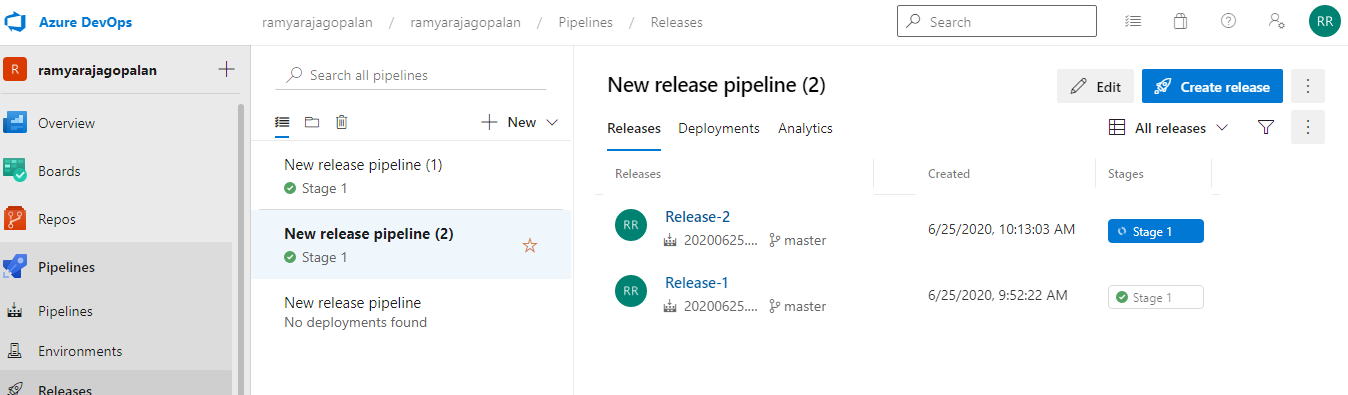


**Trigger release pipeline based on producing build artifact:**

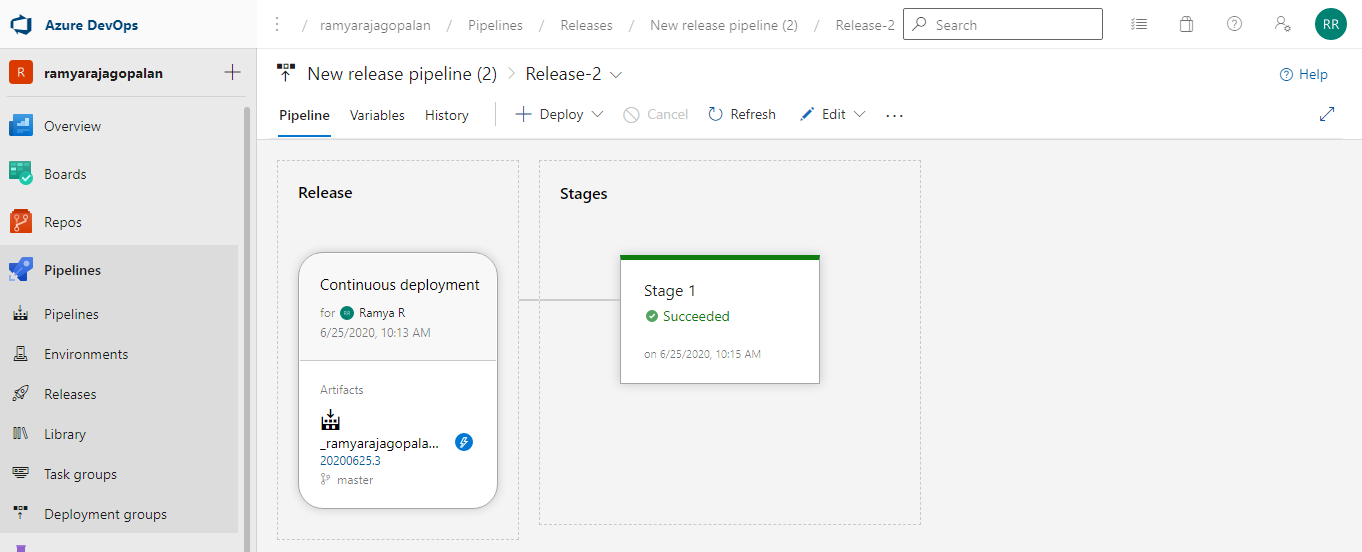
Added trigger to Release pipeline by enabling Continuous deployment trigger as below:



Release pipeline triggered automatically as soon as build was triggered in previous step:

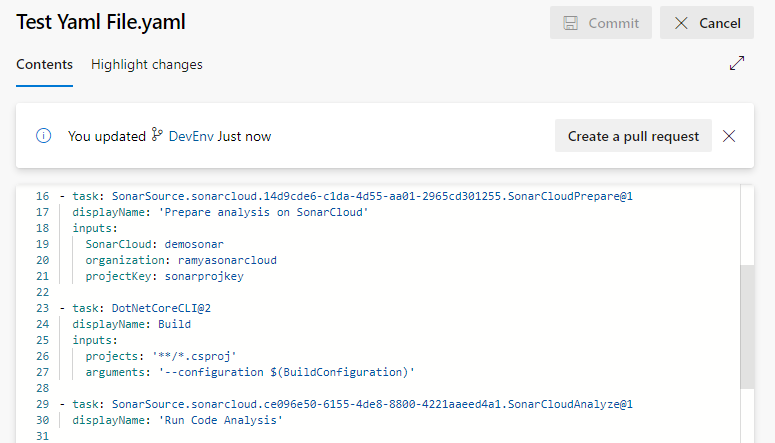


Job completed successfully:

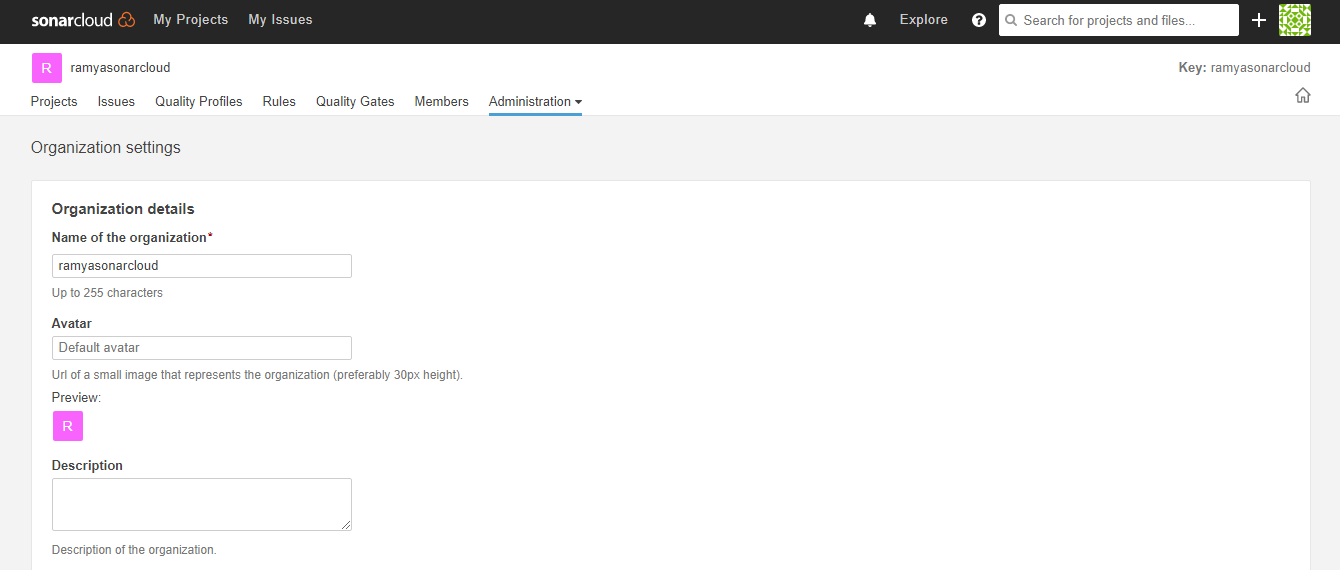


**Use Case 3: Integrate sonarqube with CI pipeline**

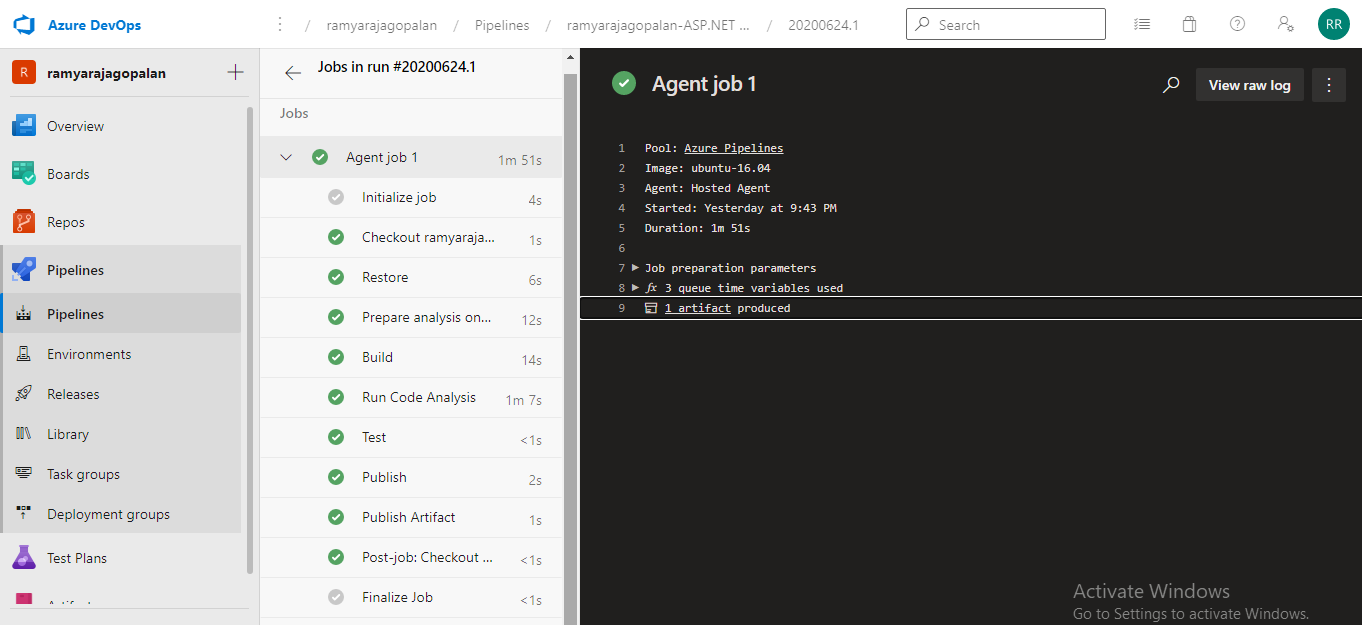
YAML Code:



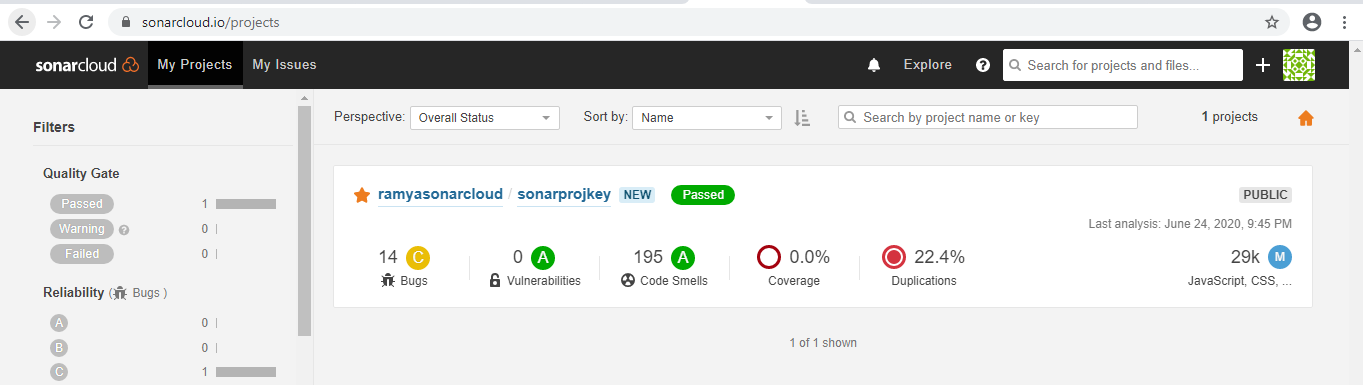
Create Organization and Project key in <https://sonarcloud.io/>



Sonar integrated pipeline is run successfully with Preparation and Run analysis steps:

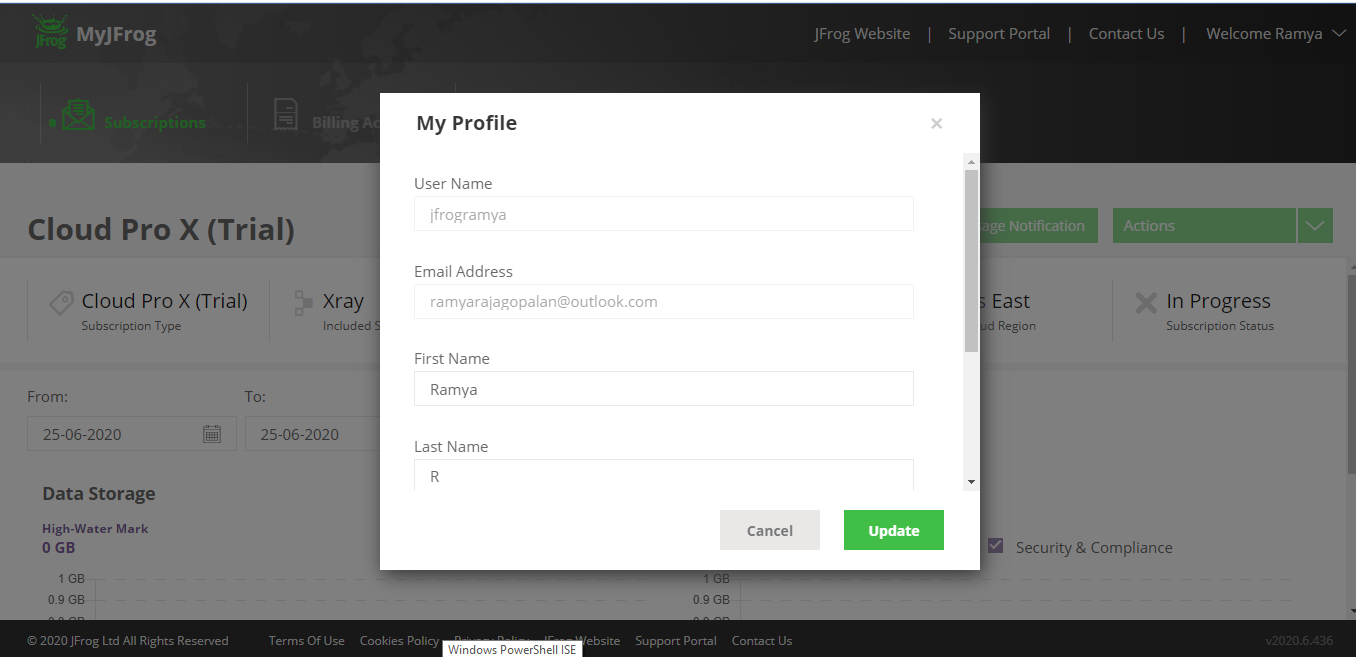


Project Analysis uploaded in Sonarcloud after pipeline completion:

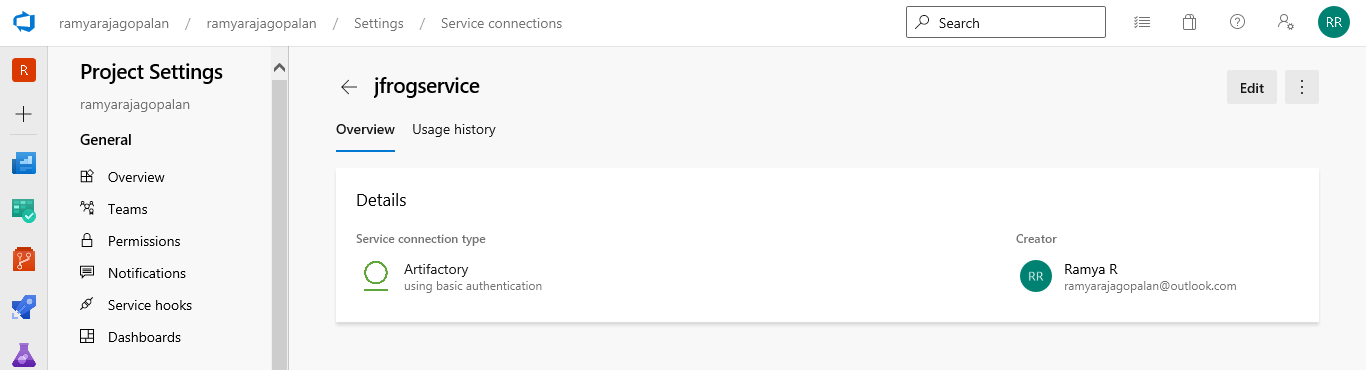


**Use Case 5: Publish application package to Azure/jFrog repository**

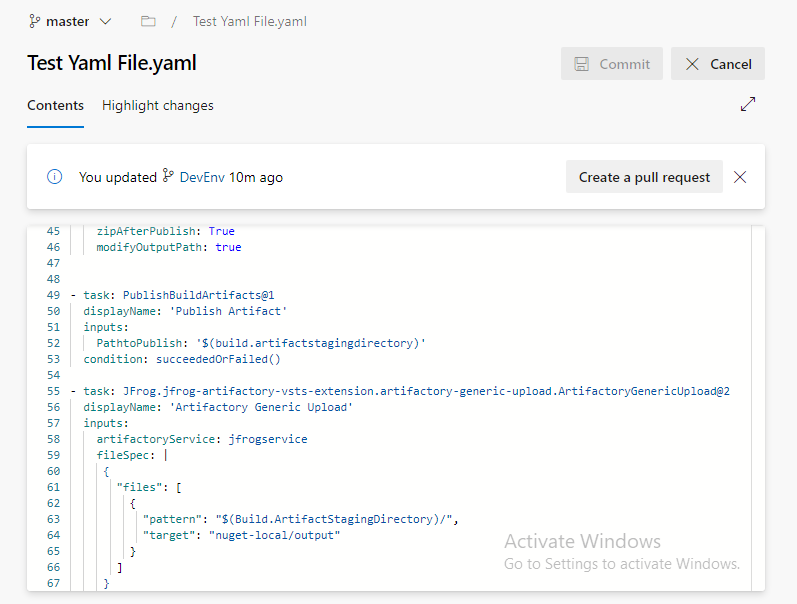
A new Jfrog account is created:



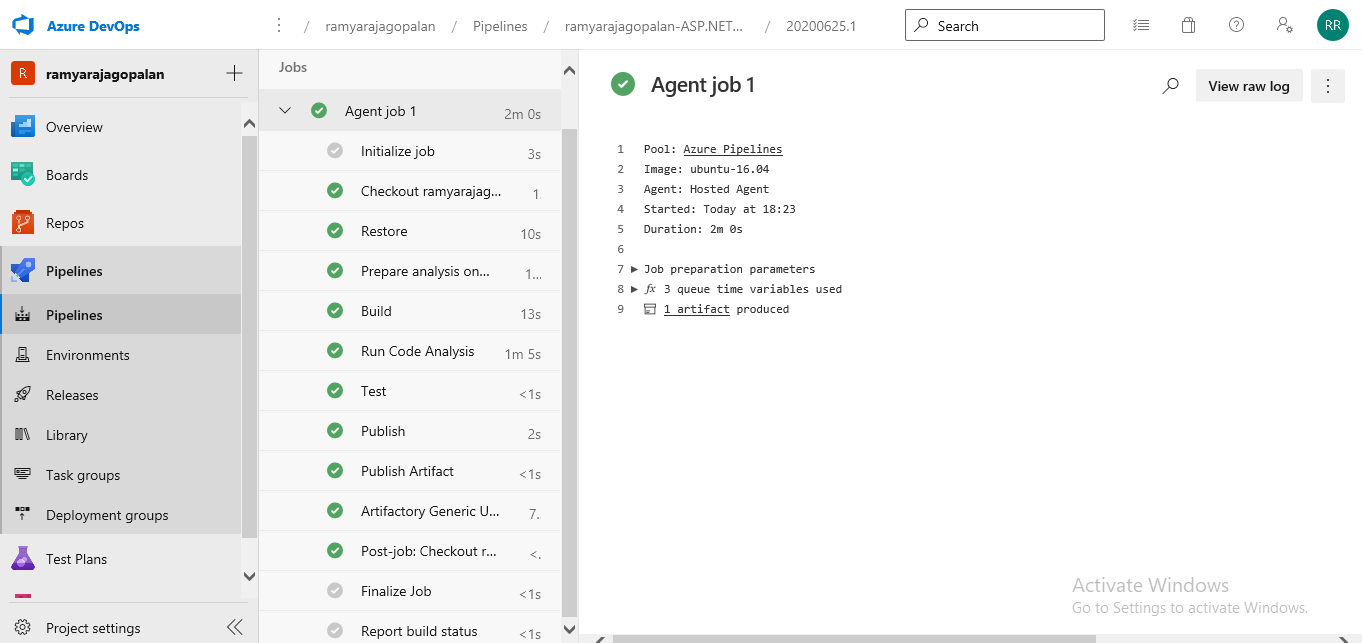
A new service connection is created



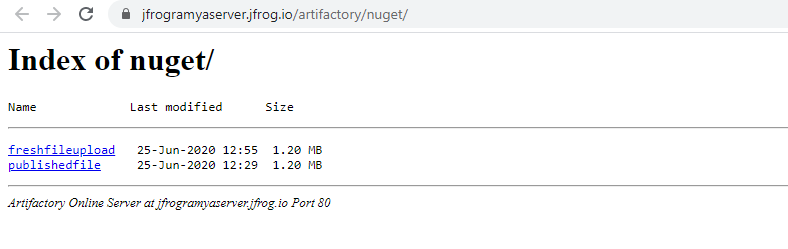
YAML code:



Pipeline completed successfully:



Artifacts(Name - fresh file upload) published in jFrog portal:

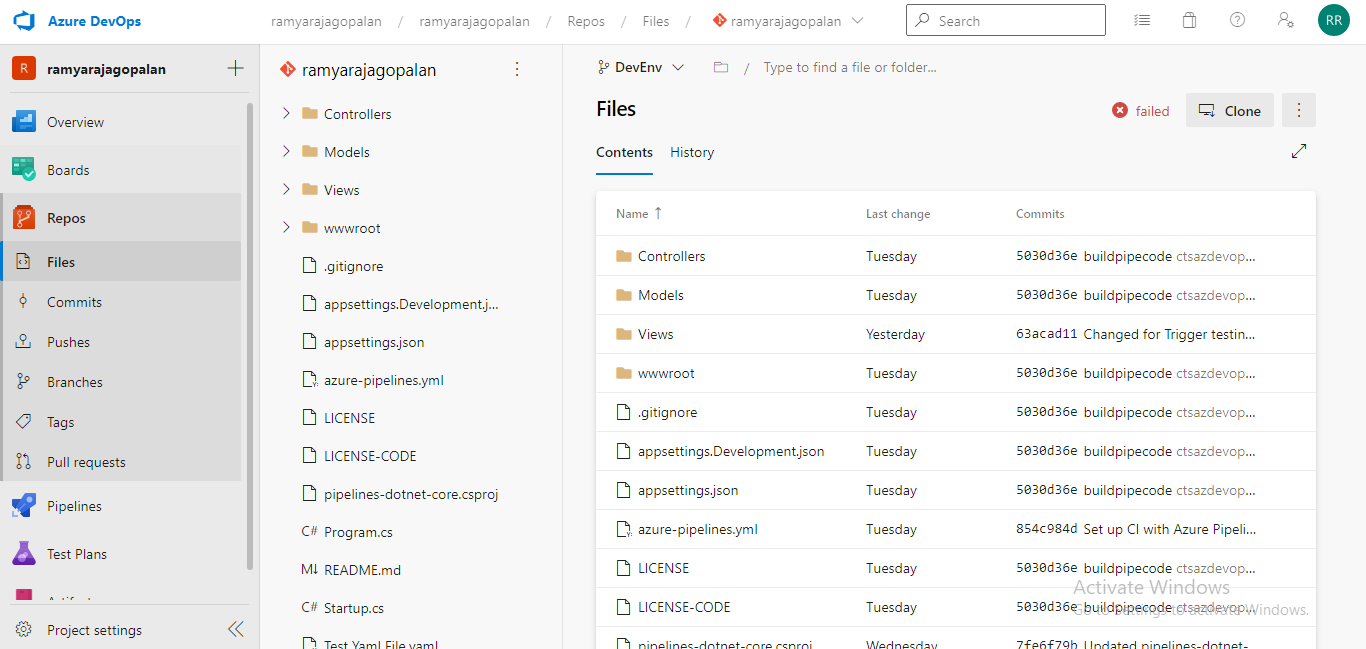


Artifacts published in Azure portal:

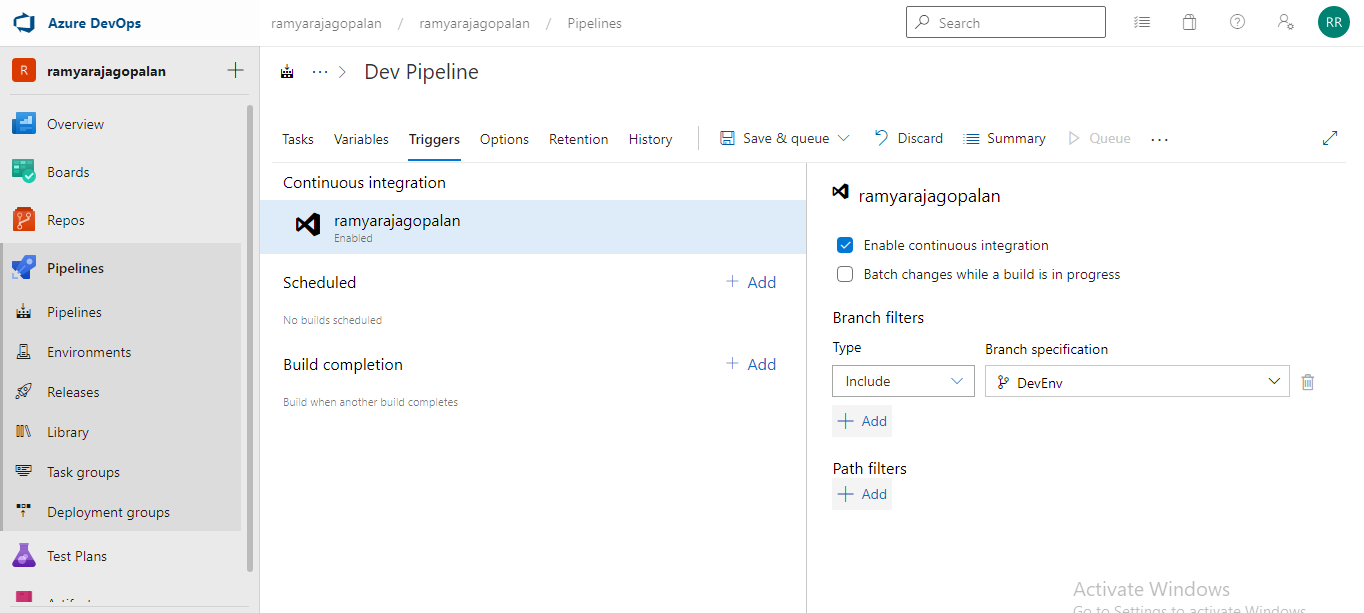


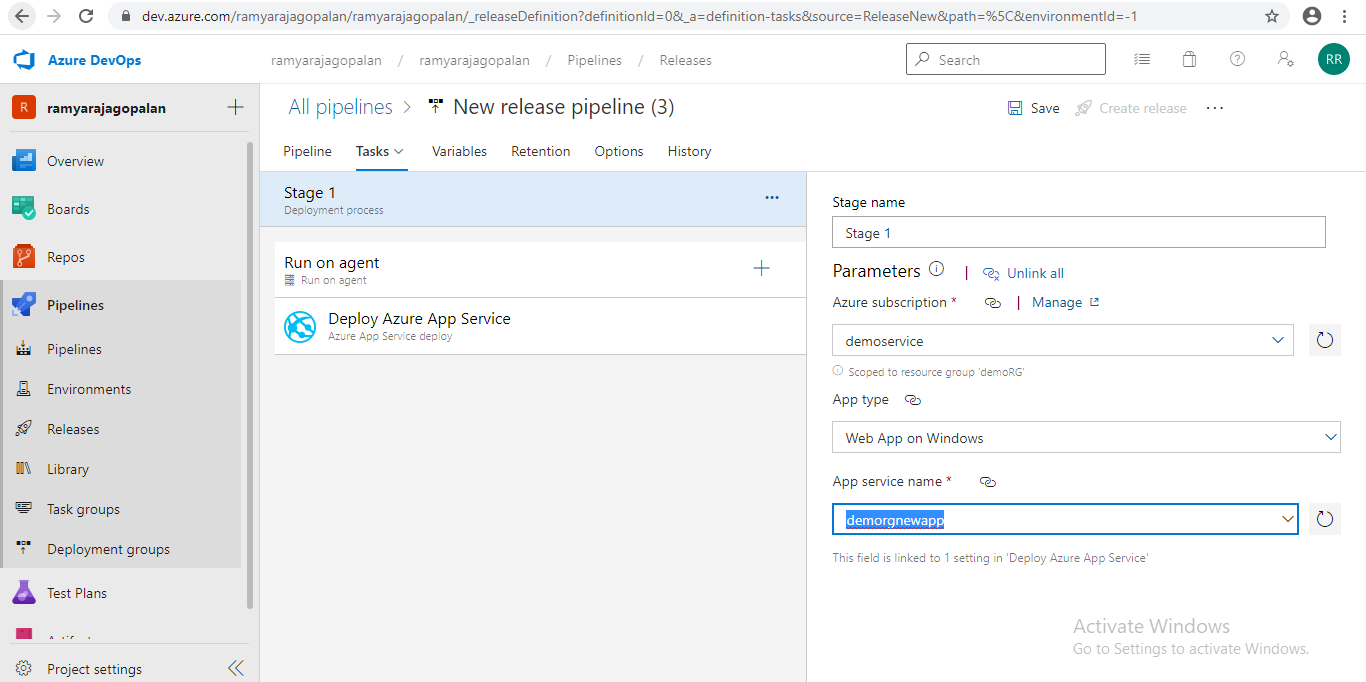
**Use Case 6: Trigger CD pipeline to push package to Dev region**

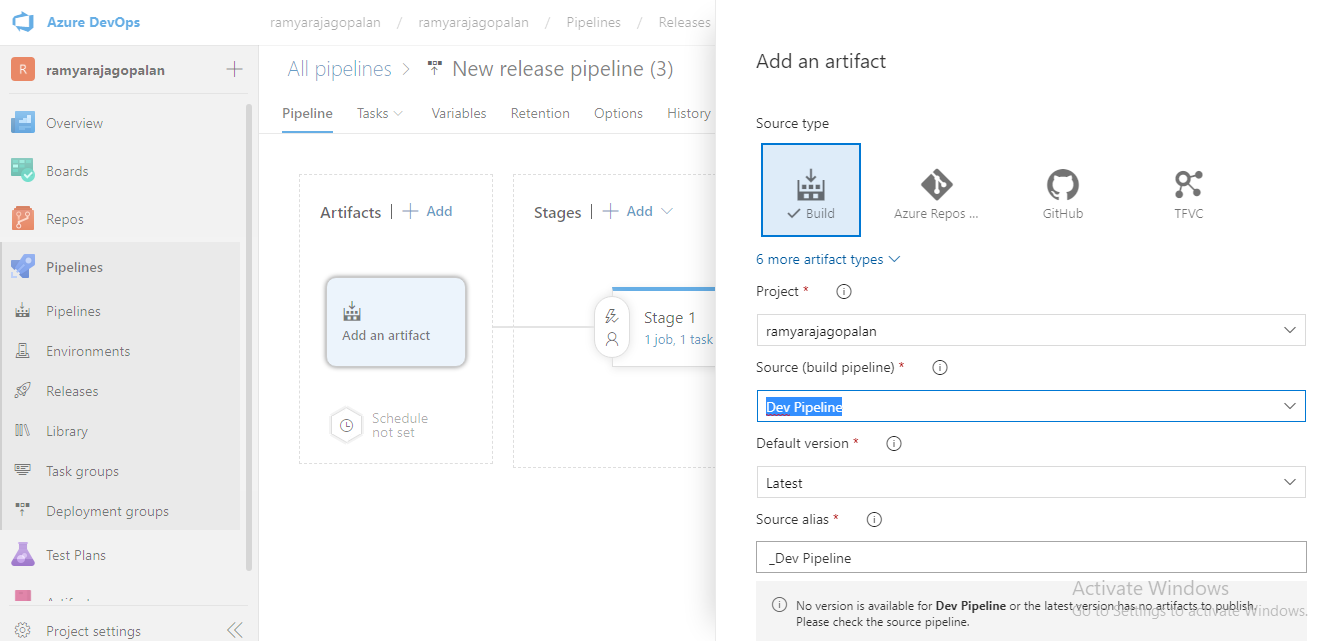
Created a Dev branch:

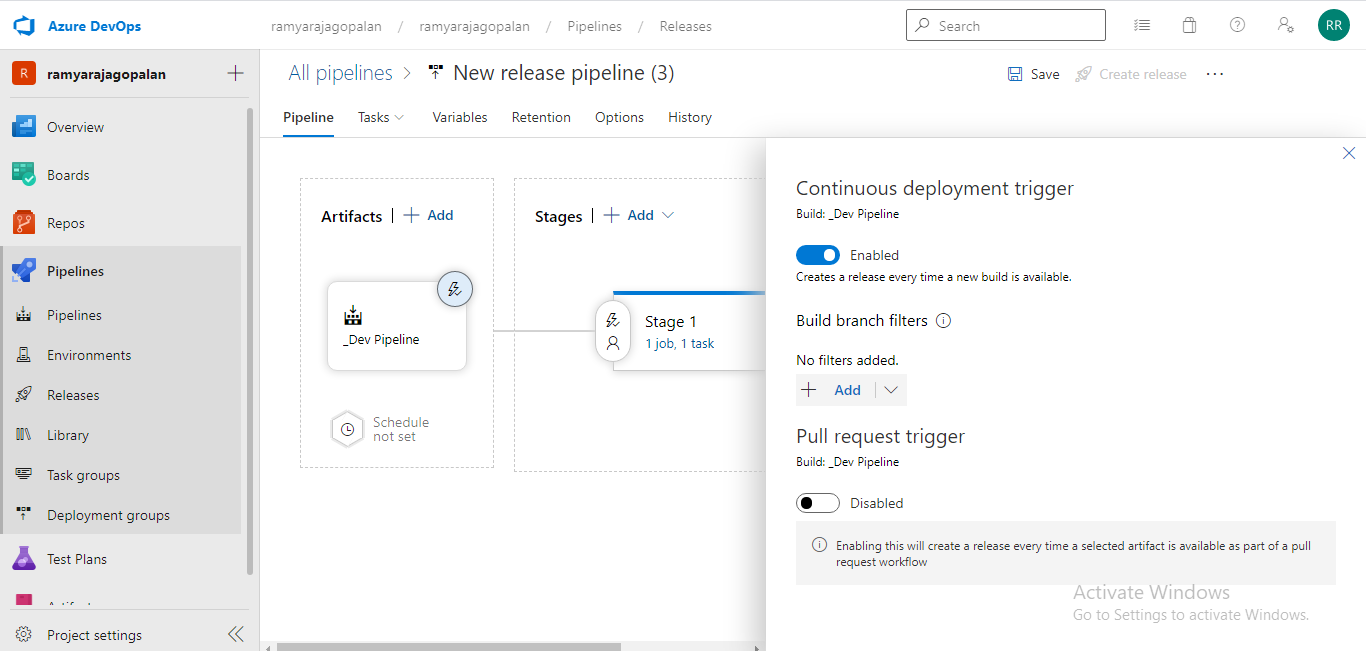


Create separate build and release pipelines for Dev regions with triggers enabled:

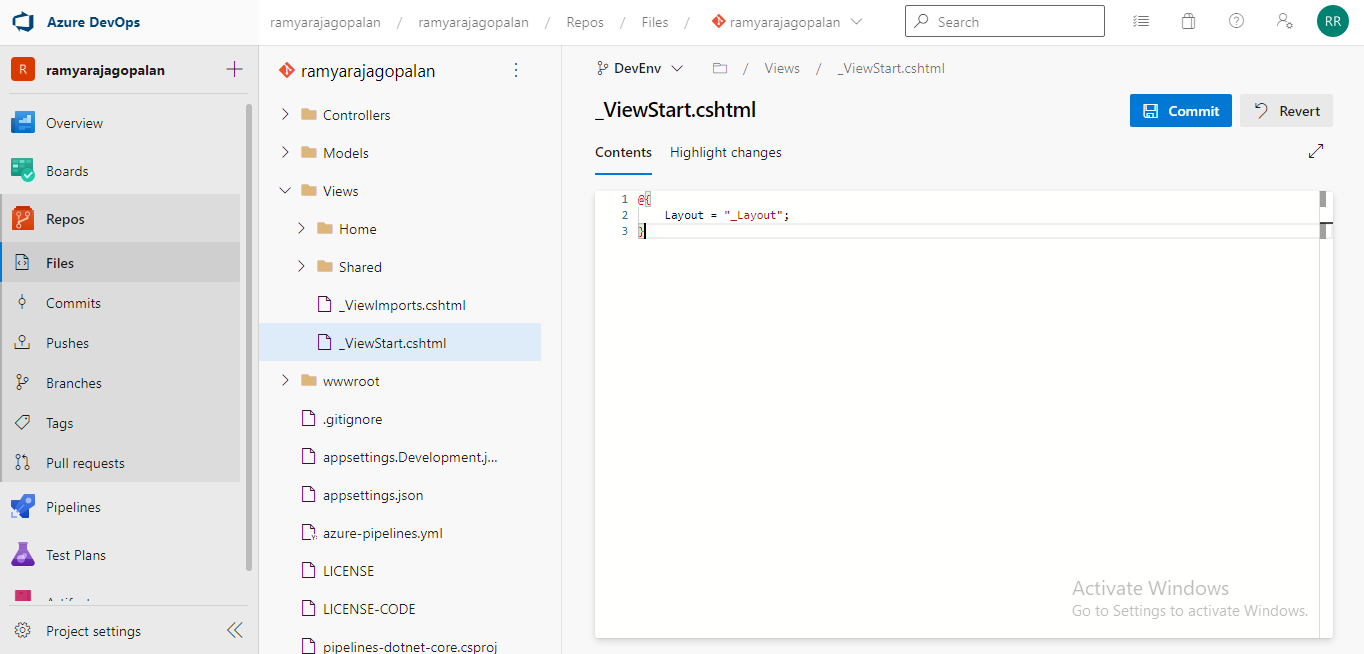


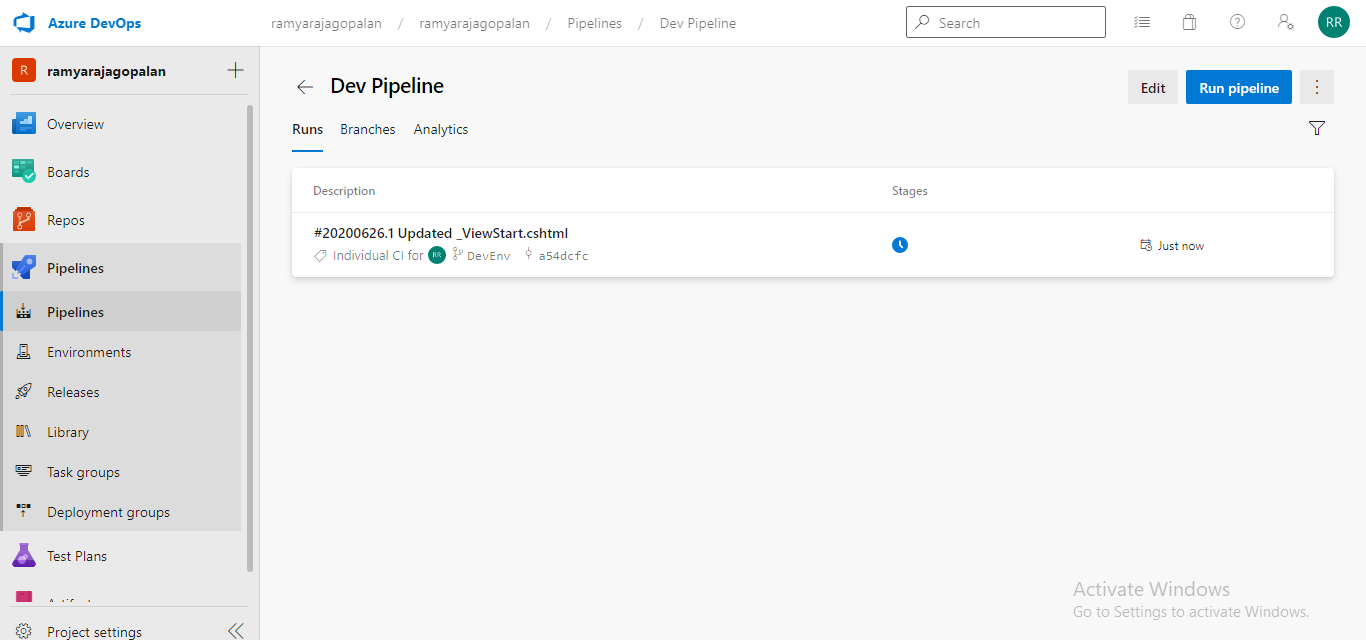


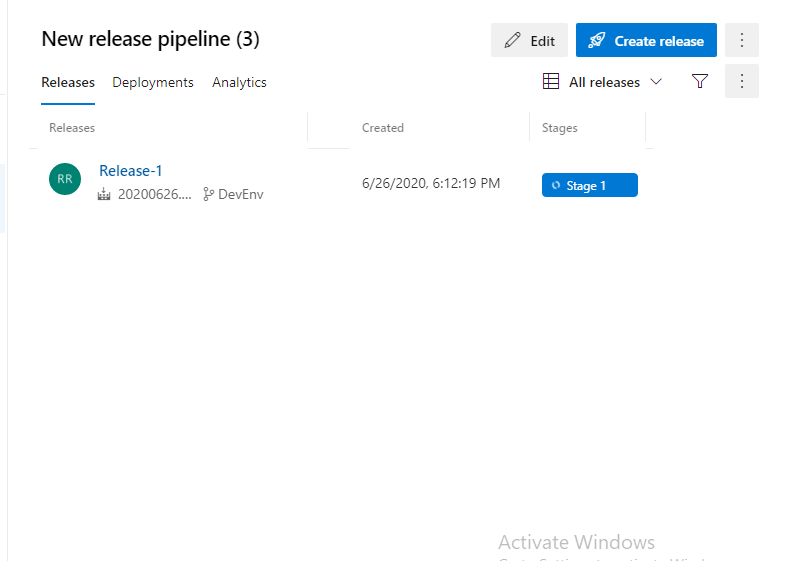


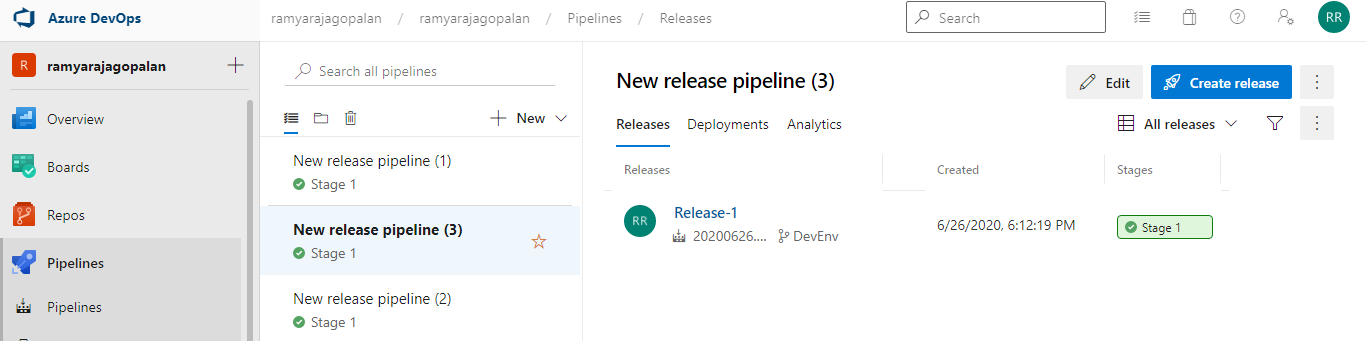


Commit code changes in dev region and it triggers both dev build pipeline and dev release pipeline to host service in Dev region:

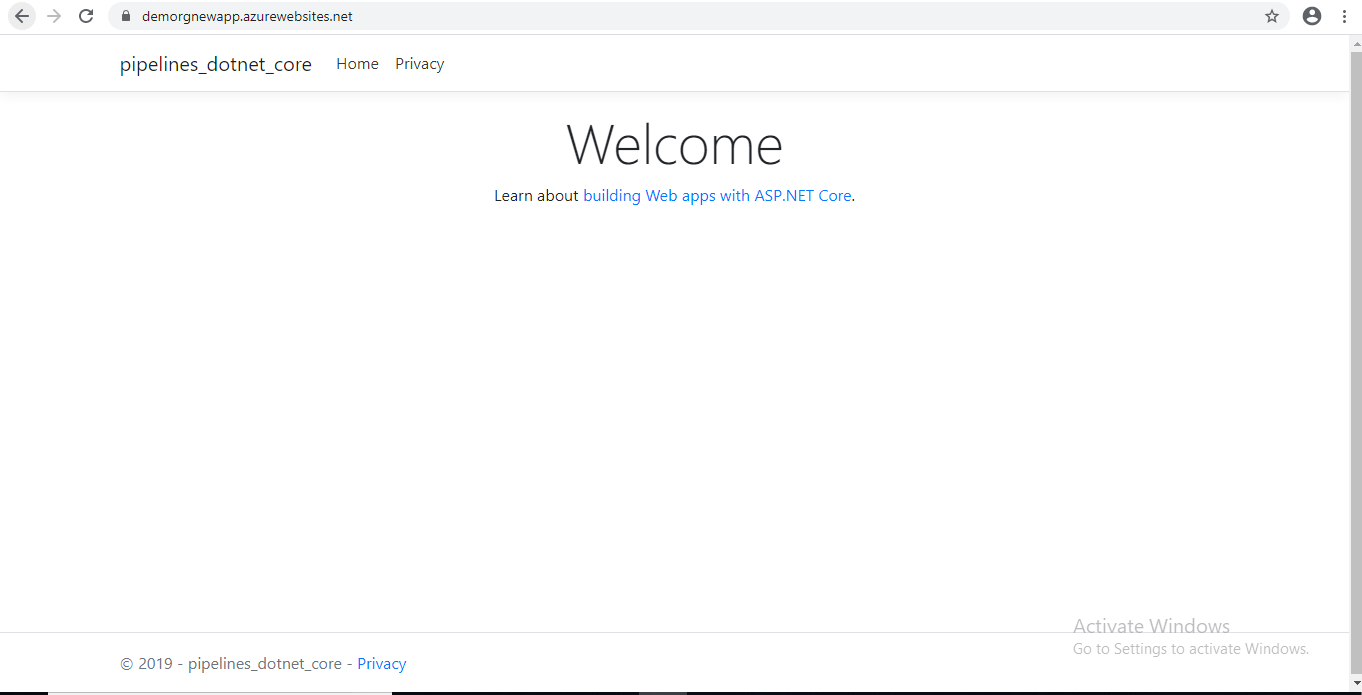








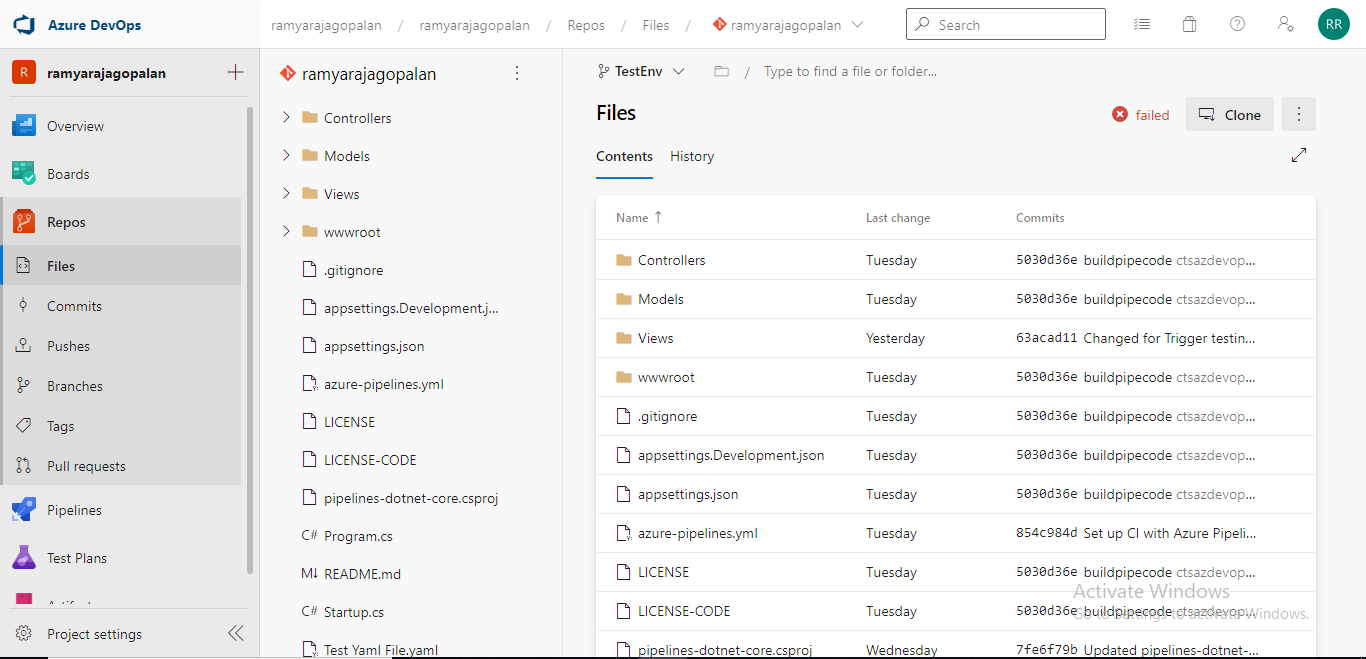
App Service hosted:



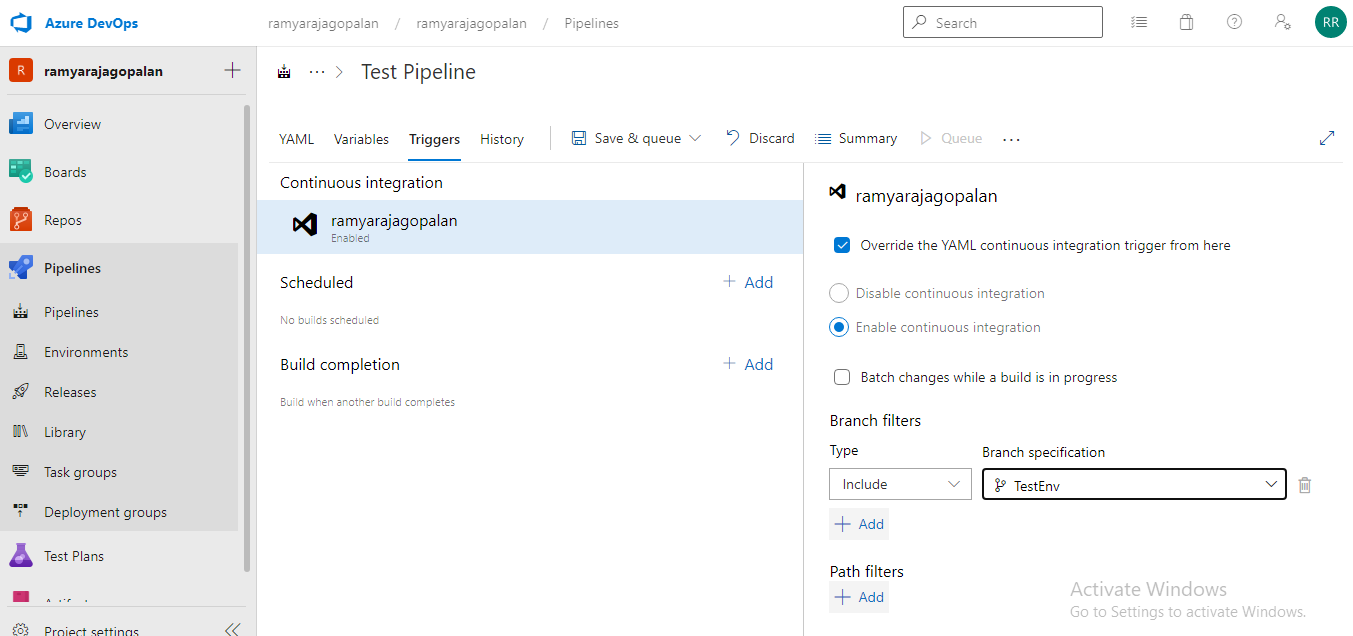
Link: - <https://demorgnewapp.azurewebsites.net>

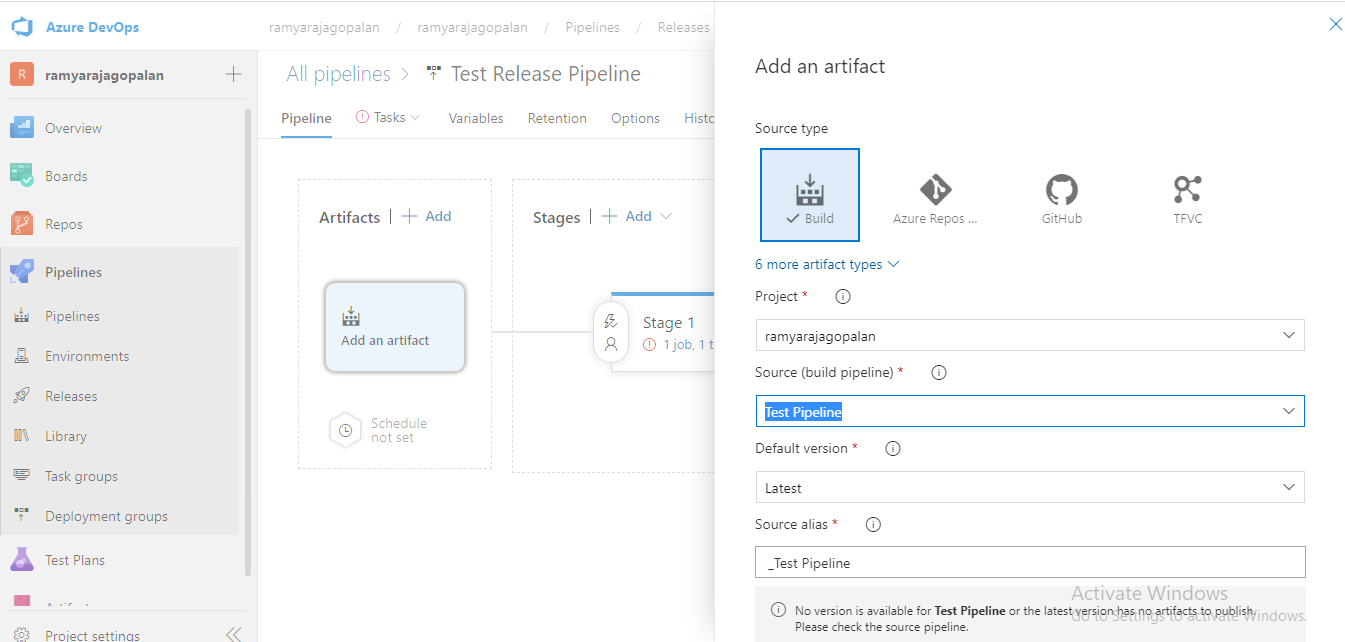
**Use Case 8: Create and trigger release pipeline to release application to test environment**

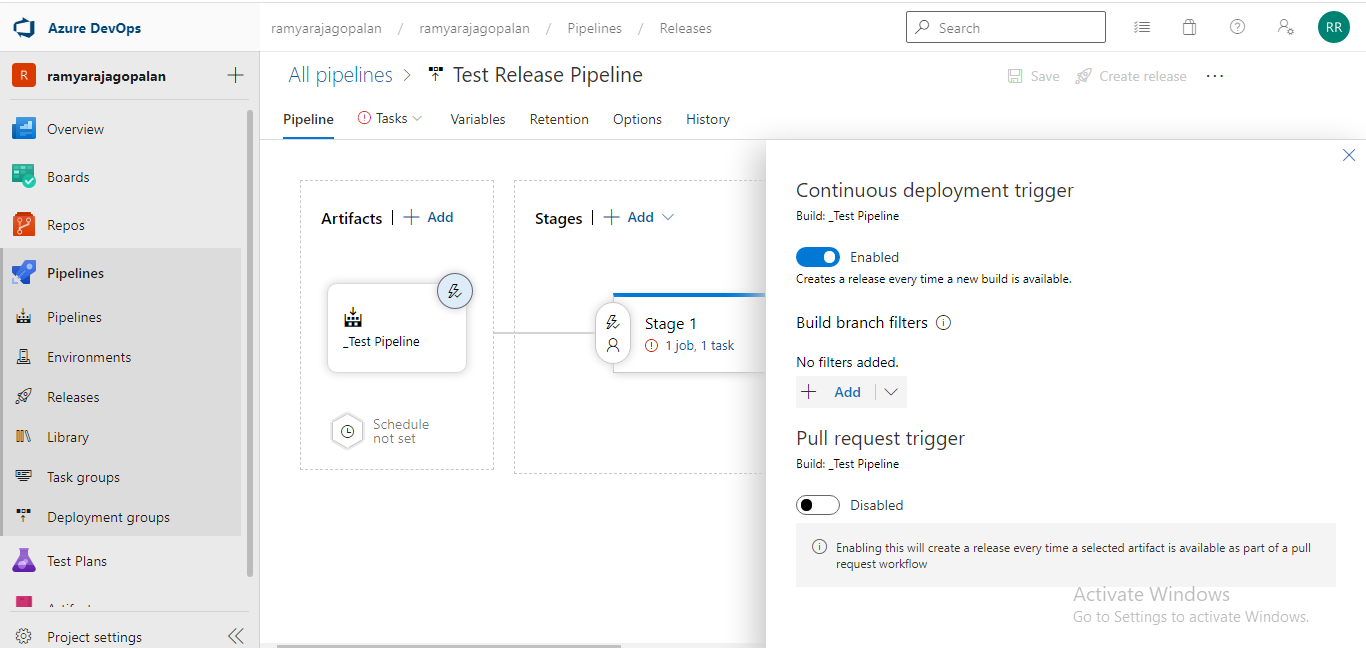
Created a Test branch:



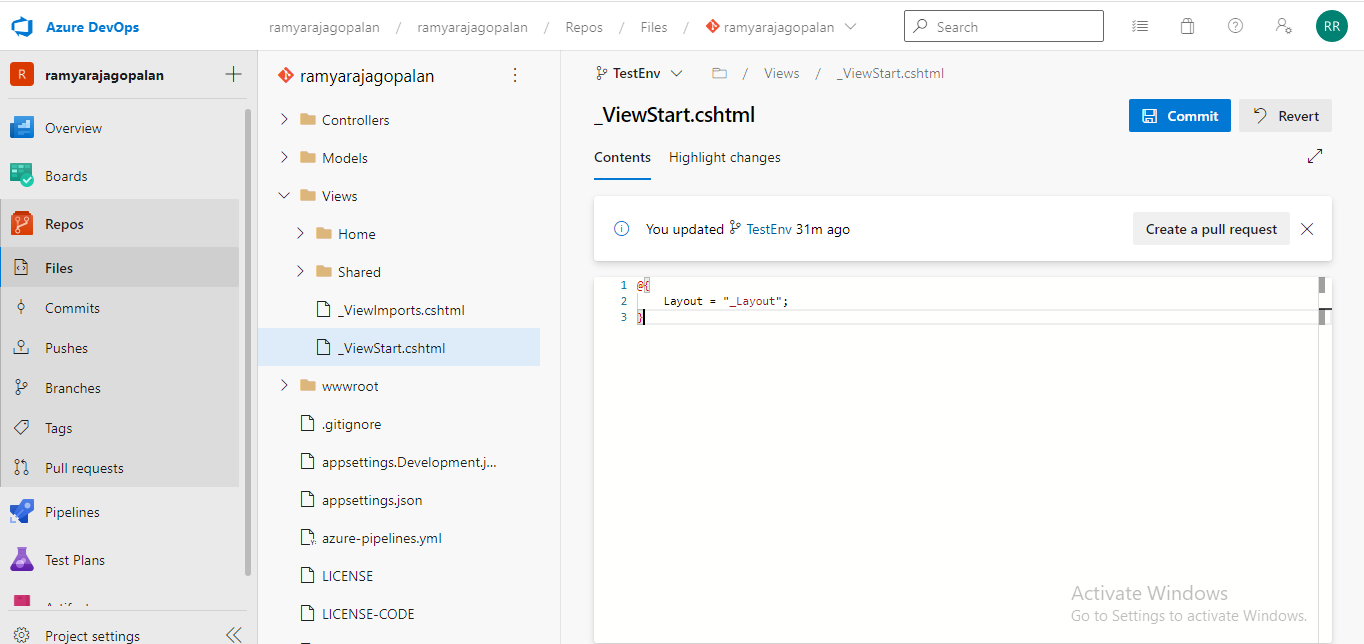
Create separate build and release pipelines for Test regions with triggers enabled:

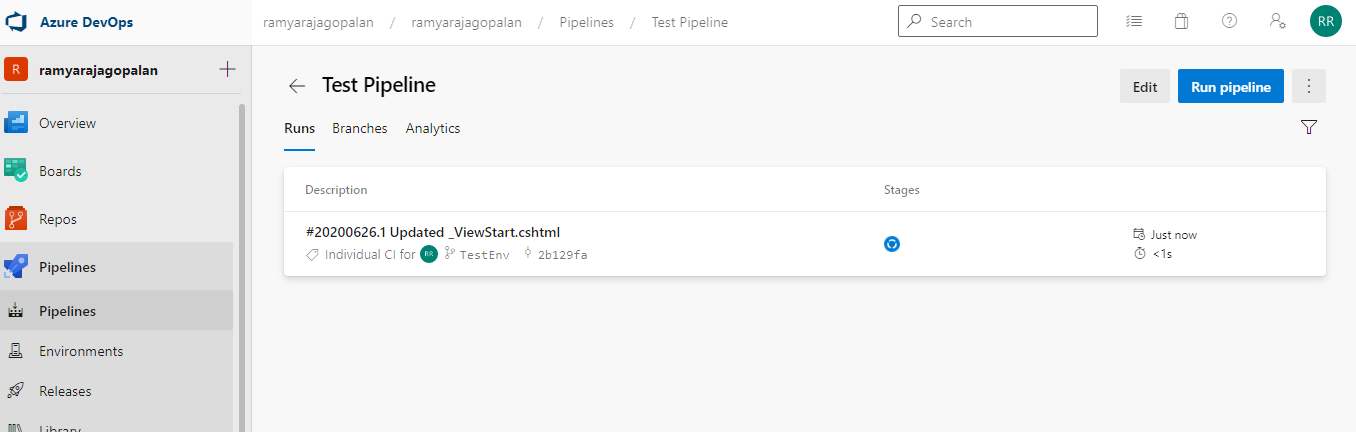


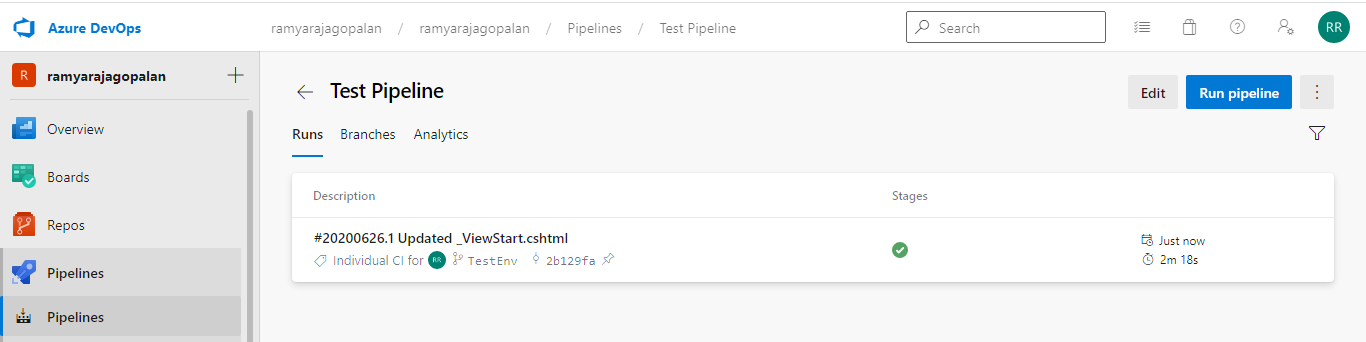


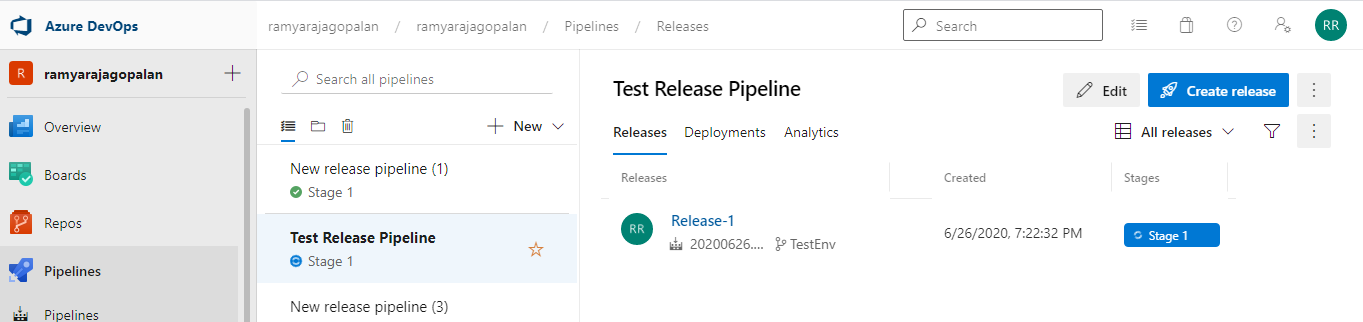


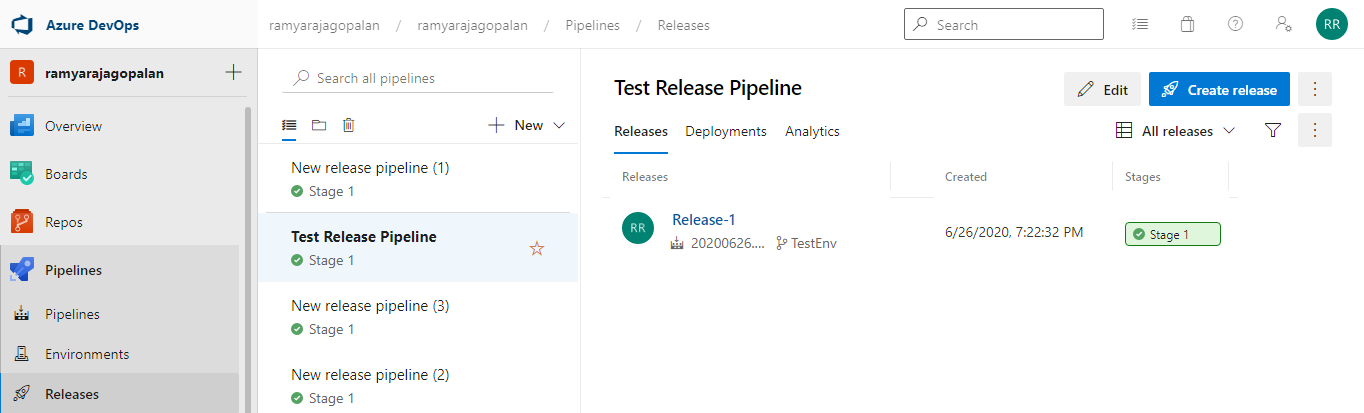
Commit code changes in test region and it triggers both test build pipeline and test release pipeline to host service in test region:



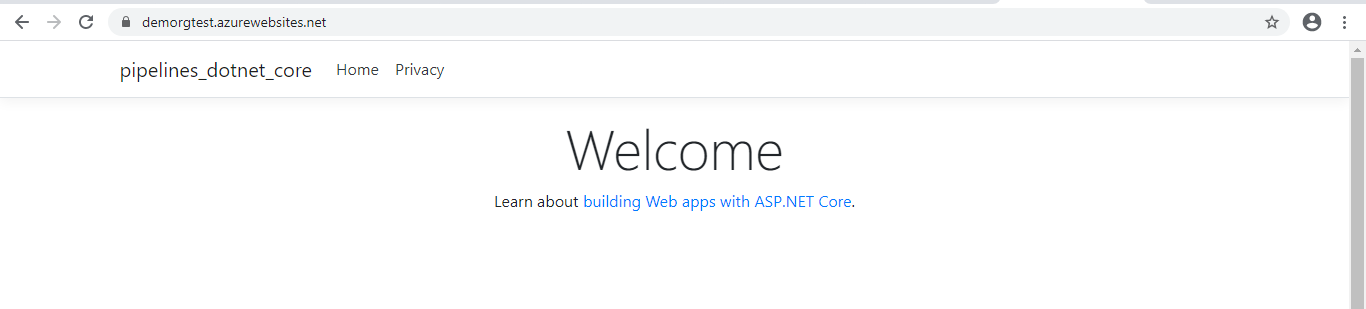








App Service hosted:



Link: <https://demorgtest.azurewebsites.net>