Activate Azure with DevOps

Module 06: End-to-End DevOps - Lab 2(a) - YAML Pipelines

Student Lab Manual

Parts of this lab has been taken from https://azuredevopslabs.com/labs/azuredevops/yaml/. View additional publicly available labs at https://azuredevopslabs.com/.

Contents

Lab 2(a): End-to-End DevOps: Develop and Test using YAML

Introduction

In this lab, you will create a pipeline using YAML to build your code and run your unit tests.

You'll learn:

- Understand the basic features of YAML Pipelines.
- Understand the value of Pipelines as code.

Prerequisites

- Microsoft Visual Studio Code
- Task 1 from https://azuredevopslabs.com/labs/azuredevops/prereq/
- Note: You may want to disable any other CI pipelines before starting the lab to prevent triggering multiple builds at once.
- If you've completed the previous Pull Requests lab, you may also need to temporarily disable any branch policies on master.
- Multi-stage pipelines preview feature enabled

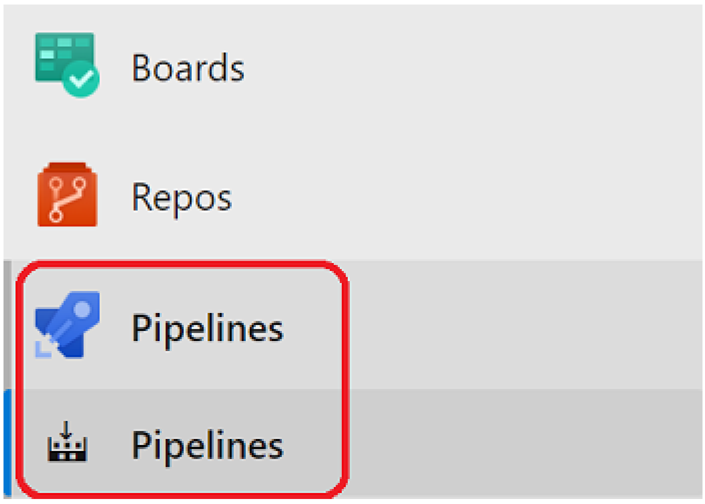
Estimated Time To Complete This Lab
45 minutes

Exercise 1: Introduction to YAML Pipelines

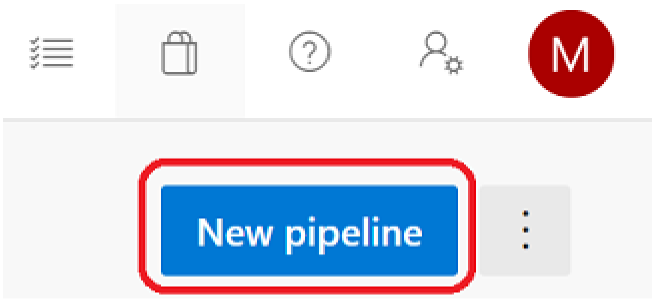
Task 1: Create a YAML pipeline from a template

1. Navigate to your project in Azure DevOps.

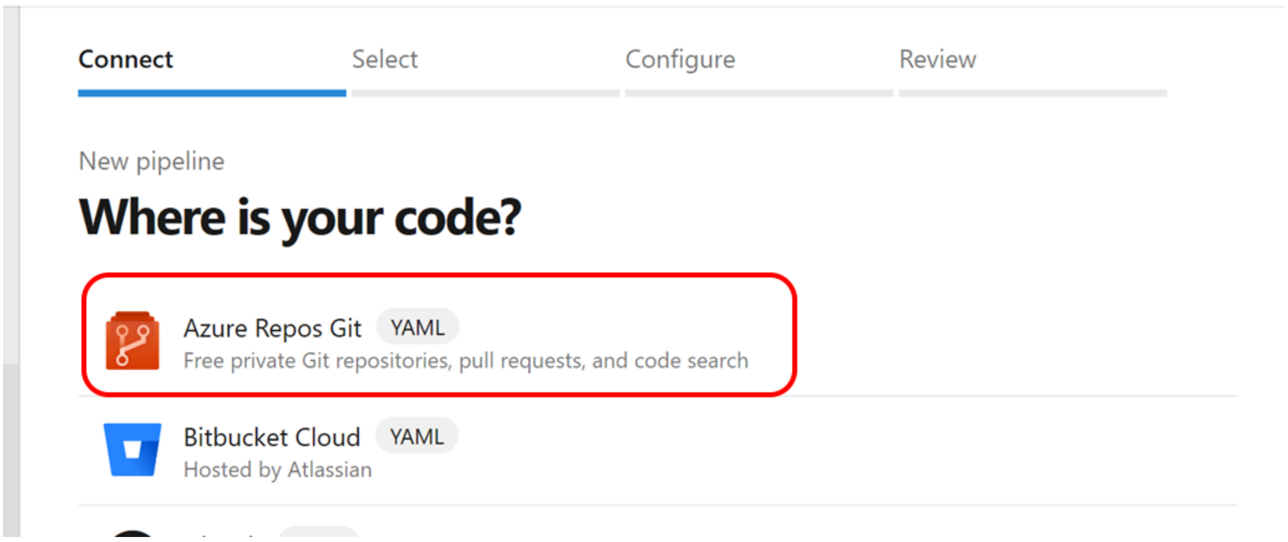
2. Navigate to **Pipelines**, then **Pipelines**.



3. Select **New pipeline** to create a new build pipeline.



4. Select **Azure Repos Git** as the source.

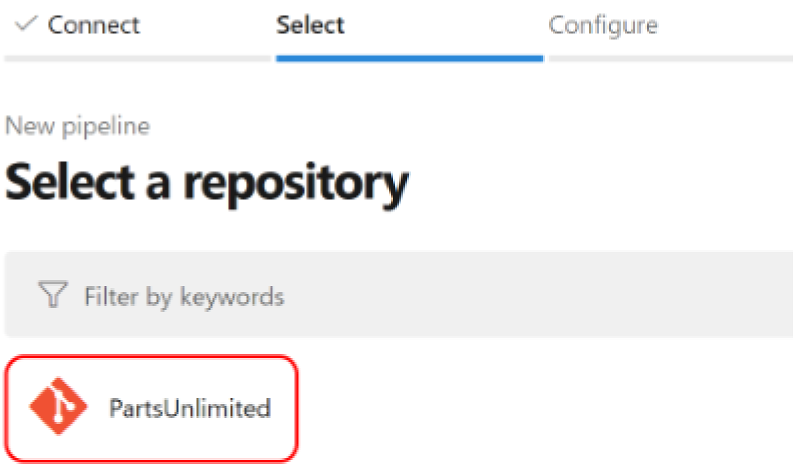


Where is your code?

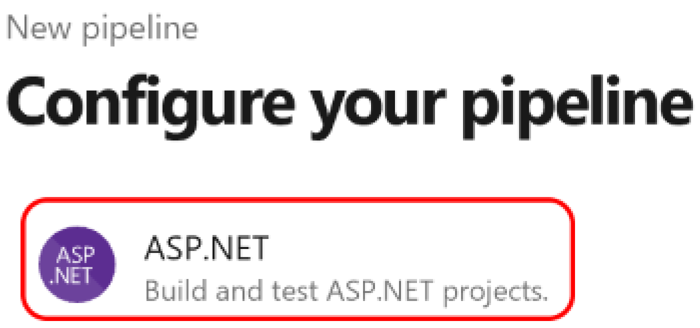
Azure Repos Git YAML
Free private Git repositories, pull requests, and code search

Bitbucket Cloud YAML
Hosted by Atlassian

5. Next, select the **PartsUnlimited** repo.



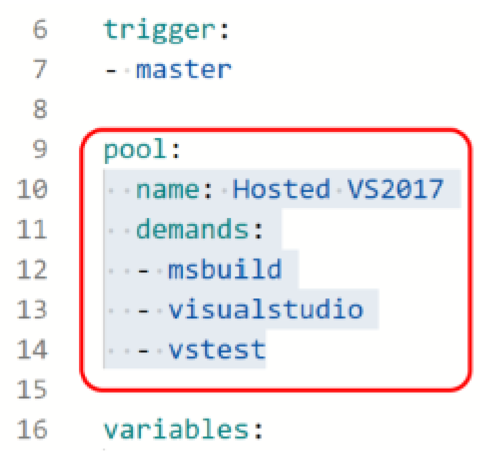
6. In the next screen, pick the ASP.NET template (we could start from an empty template, but the template has some tasks that will be needed anyway and simplifies the initial setup.)



7. Looking at the YAML build pipeline, we can see that it looks like a markup file and contains code for some predefined tasks. By default, the pipeline will be saved as a new file called "azure-pipelines.yml" in the root of the repository and contains everything needed to build and test a typical ASP.NET solution. You can also customize the build as needed.

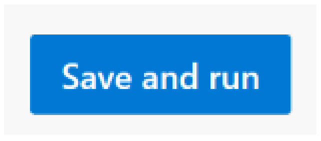
8. In this case, update the pool to specify that the build should use a Visual Studio 2017 build VM and add some demands. Be sure to keep it at the same two-space indentation.

```
name: Hosted VS2017
demands:
- msbuild
- visualstudio
- vstest
```

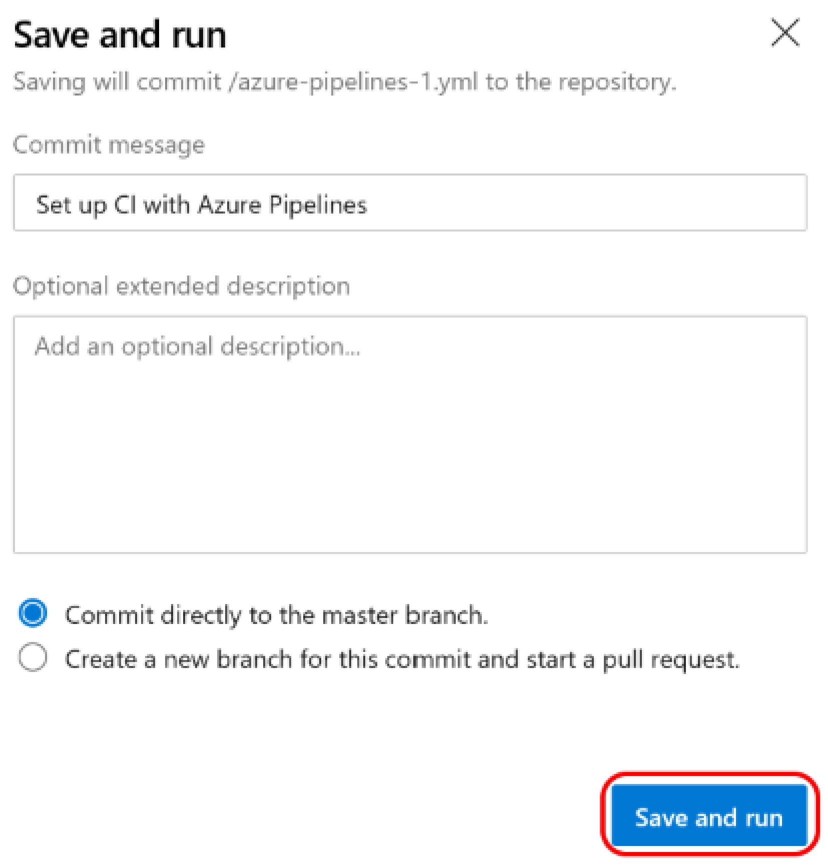


```
trigger:
- master

pool:
  name: Hosted VS2017
  demands:
  - msbuild
  - visualstudio
  - vstest

variables:
```

9. Let's test the build pipeline as is by queueing a new build. Click **"Save and run"** near the top.



10. Update the commit message if you'd like. Leave the **Commit directly to the master branch** selected and click **"Save and run"** again to confirm the commit.



Save and run

Saving will commit /azure-pipelines-1.yml to the repository.

Commit message

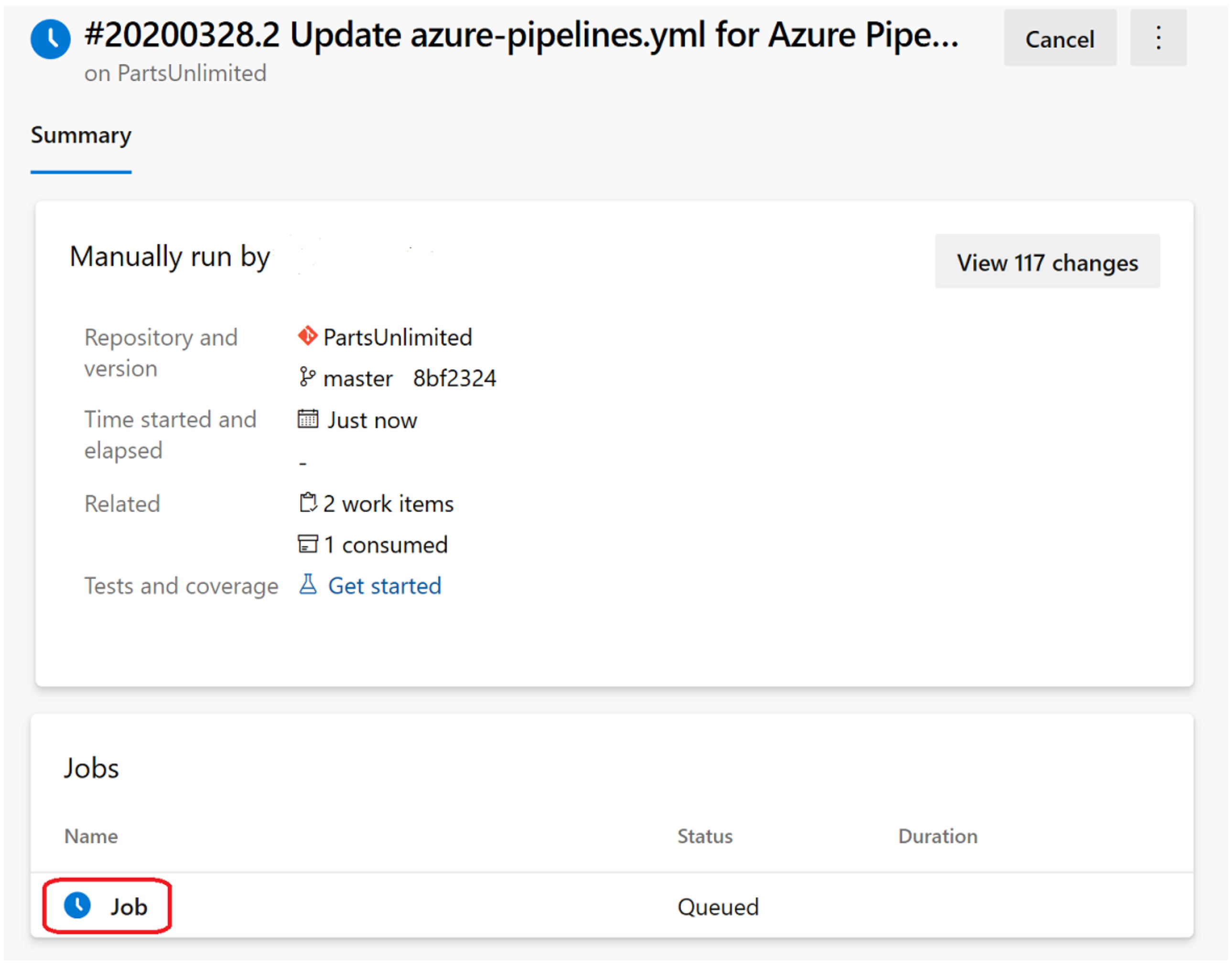
Set up CI with Azure Pipelines

Optional extended description

Add an optional description...

[✓ Commit directly to the master branch.]
Create a new branch for this commit and start a pull request.

11. The page should update to a view of the build running. By clicking **"Job"** at the bottom, we can view the build log output as our project is being built on a hosted agent. If you want to review an earlier task, you can click to review its logs. Verify that the build is successful.



#20200328.2 Update azure-pipelines.yml for Azure Pipe...

on PartsUnlimited

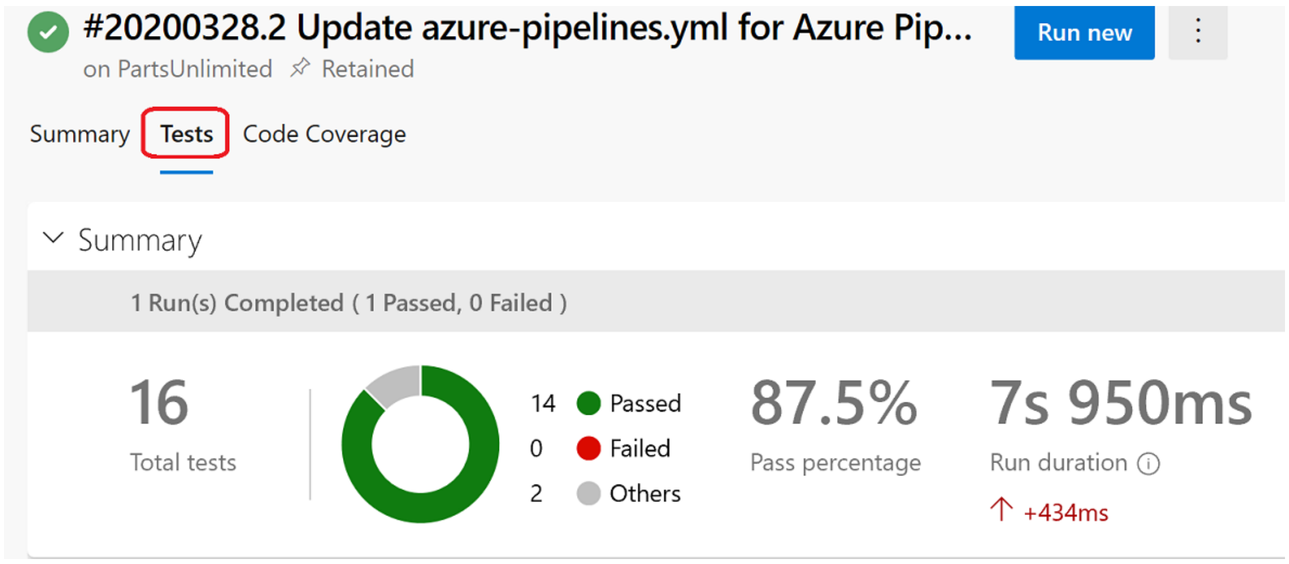
Manually run by

Repository and version	PartsUnlimited master 8bf2324
Time started and elapsed	Just now -
Related	2 work items 1 consumed
Tests and coverage	Get started

Jobs

Name	Status	Duration
Job	Queued	

12. Select the **Tests** tab to review test results for this build.



#20200328.2 Update azure-pipelines.yml for Azure Pip...

on PartsUnlimited Retained

1 Run(s) Completed (1 Passed, 0 Failed)

16 Total tests

14	Passed
0	Failed
2	Others

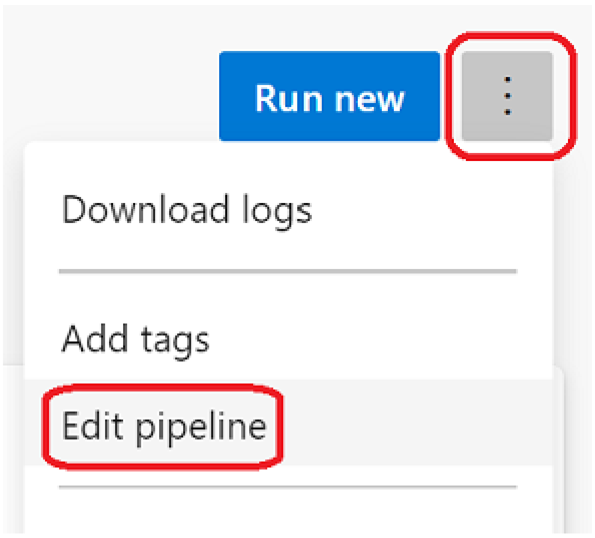
87.5% Pass percentage

7s 950ms Run duration

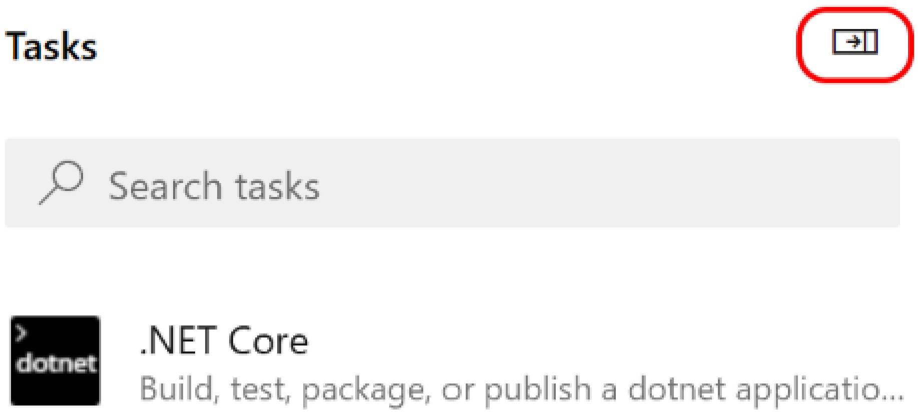
+434ms

Task 2: Adding Tasks to YAML Pipelines

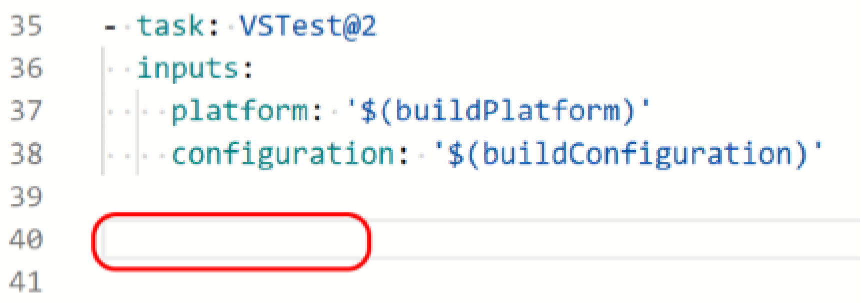
1. Now that our intial build and test processes are successful, we can make additional customizations. From the options(...) dropdown, select **Edit pipeline**.



2. This should return you to the YAML editor view. Notice a task library that appears to the right. The library can be used to add tasks to your YAML file. This task library contains the same tasks as the Classic pipeline library. You can toggle this view by clicking the hide button as shown below.

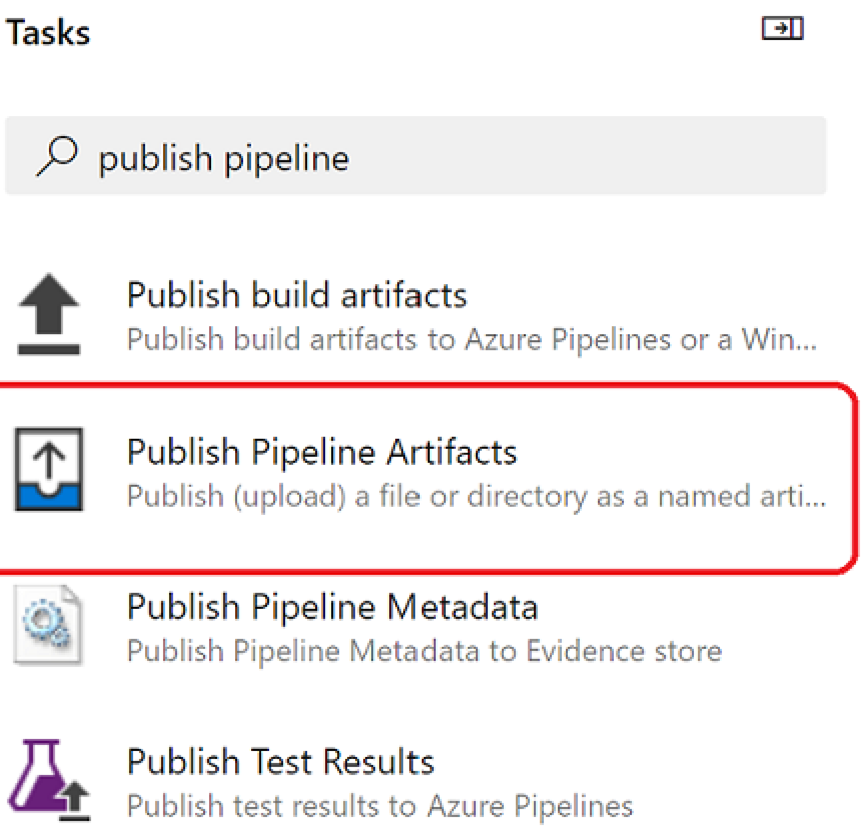


3. In order to be able to deploy the output of our build later, we need to add a task to publish the build output. This will save the output to Azure DevOps. Place the cursor on a blank line after the test step. Try to leave an extra blank line in between tasks to make the pipeline easier to read.



```
- task: VSTest@2
  inputs:
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'


```

4. Search the tasks for **"publish pipeline"** and select the **Publish Pipeline Artifacts** task.


Tasks

publish pipeline

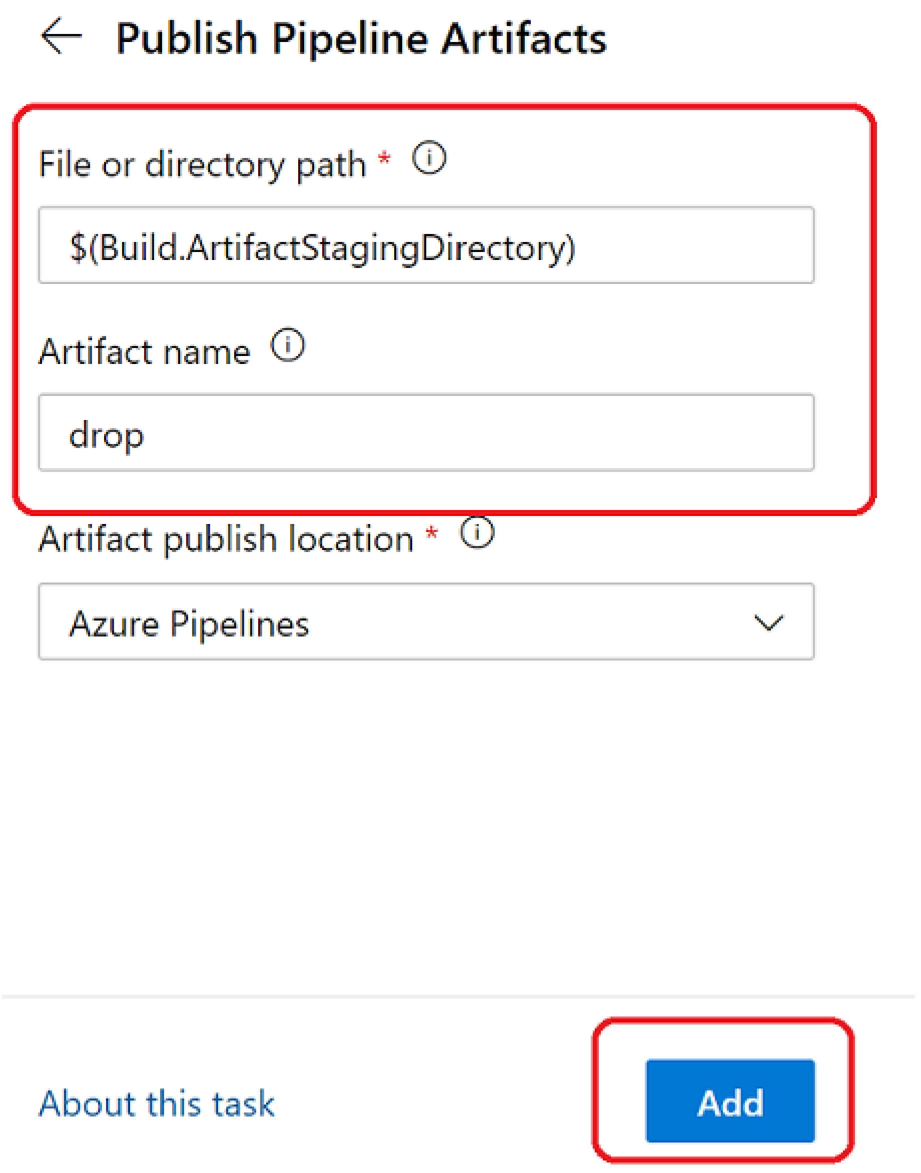
Publish build artifacts
Publish build artifacts to Azure Pipelines or a Win...

Publish Pipeline Artifacts
Publish (upload) a file or directory as a named arti...

Publish Pipeline Metadata
Publish Pipeline Metadata to Evidence store

Publish Test Results
Publish test results to Azure Pipelines

5. Change the **File or directory path** to publish to $(Build.ArtifactStagingDirectory). Enter the **Artifact name** as 'drop' and click **Add**. This task will publish the pipeline artifacts to a location in Azure DevOps that will be downloadable under the alias **drop**.

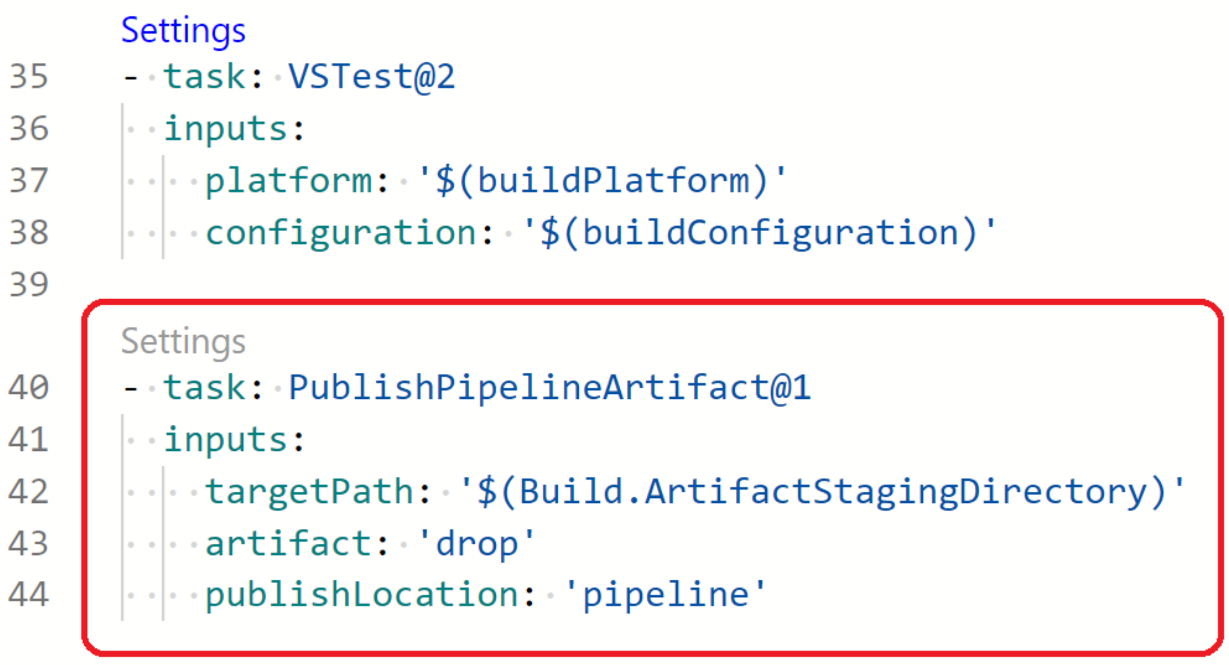

Publish Pipeline Artifacts

File or directory path *: $(Build.ArtifactStagingDirectory)

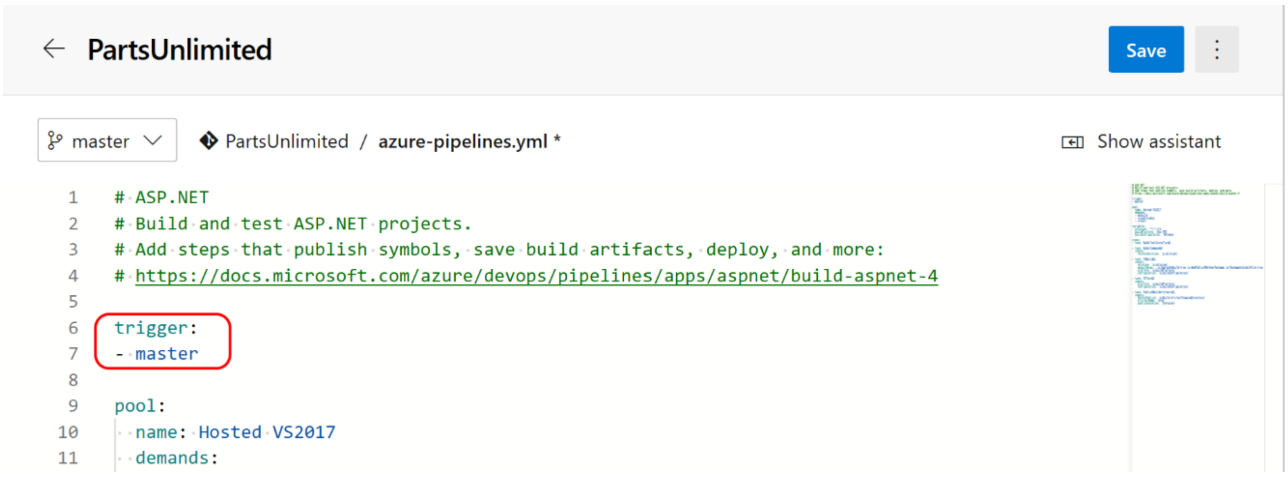
Artifact name: drop

Artifact publish location *: Azure Pipelines

6. Review the code that was added where the cursor was.


```yaml
Settings
- task: VSTest@2
  inputs:
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'

Settings
- task: PublishPipelineArtifact@1
  inputs:
    targetPath: '$(Build.ArtifactStagingDirectory)'
    artifact: 'drop'
    publishLocation: 'pipeline'
```

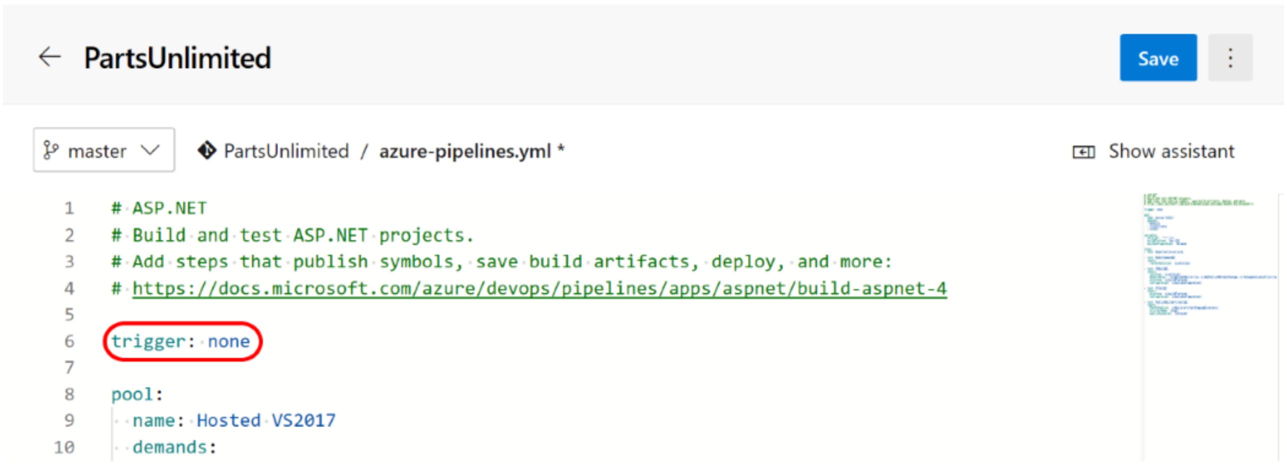
7. Before we save the changes, let's review what triggers this build. Near the top of the YAML file, notice the trigger is set to the master branch. This was set as part of the template we chose in Task 1. Since this YAML file is stored in the master branch, whenever we save a change to the file, a build will automatically be triggered.



```
# ASP.NET
# Build and test ASP.NET projects.
# Add steps that publish symbols, save build artifacts, deploy, and more:
# https://docs.microsoft.com/azure/devops/pipelines/apps/aspnet/build-aspnet-4

trigger:
- master

pool:
  name: Hosted VS2017
  demands:
```

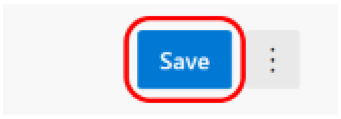
8. To disable the trigger, modify the statement as shown below.

```
trigger: none
```

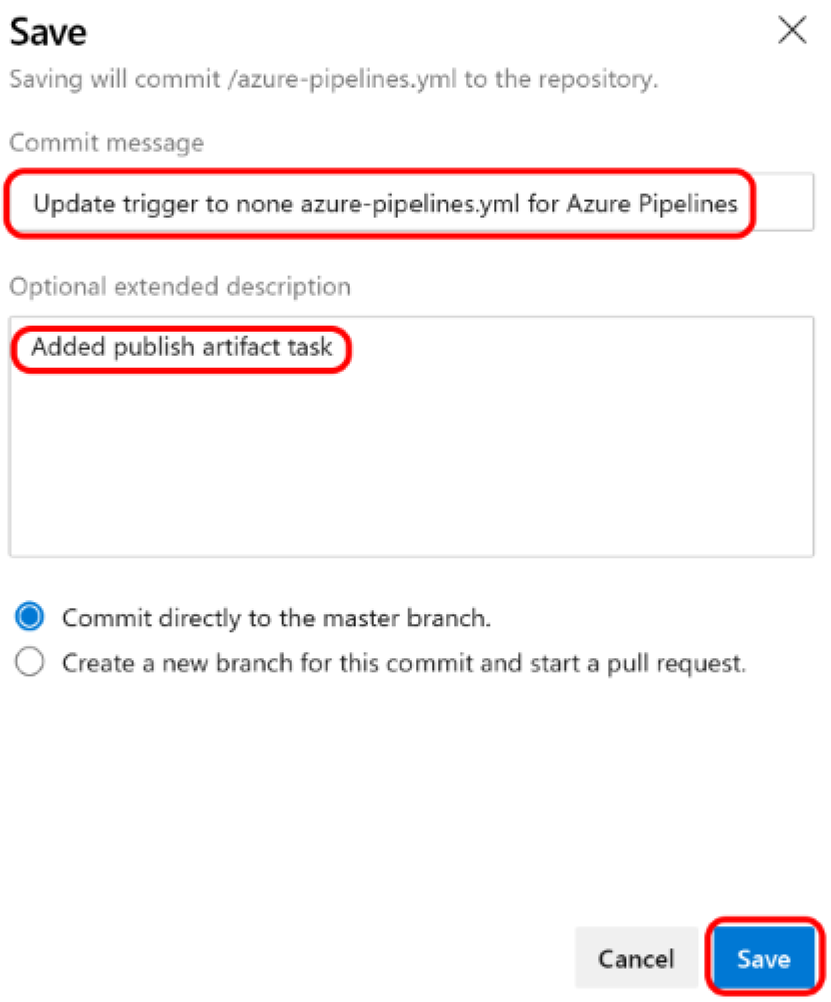


```
# ASP.NET
# Build and test ASP.NET projects.
# Add steps that publish symbols, save build artifacts, deploy, and more:
# https://docs.microsoft.com/azure/devops/pipelines/apps/aspnet/build-aspnet-4

trigger: none

pool:
  name: Hosted VS2017
  demands:
```

9. Click **Save** in the top right corner.



10. Update the commit message and click **Save** again to commit the changes. Notice no build has been triggered.


Save

Saving will commit /azure-pipelines.yml to the repository.

Commit message

Update trigger to none azure-pipelines.yml for Azure Pipelines

Optional extended description

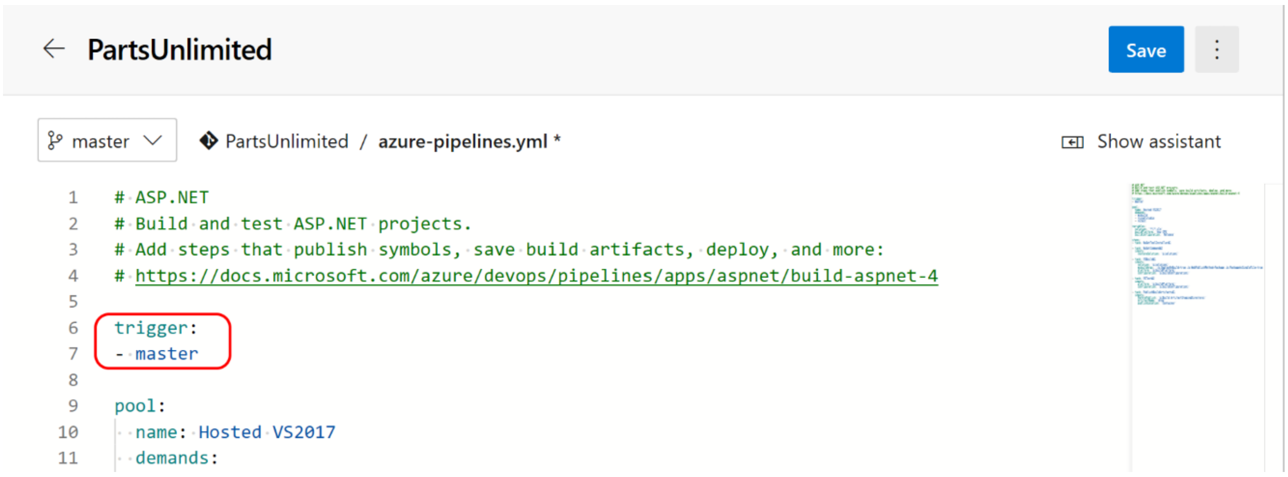
Added publish artifact task

[✓ Commit directly to the master branch.]

Create a new branch for this commit and start a pull request.

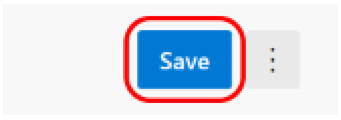
11. Because we will need the continuous integration trigger in later sections, let's re-enable it. Change the statement back to include master.

```
trigger:
- master
```

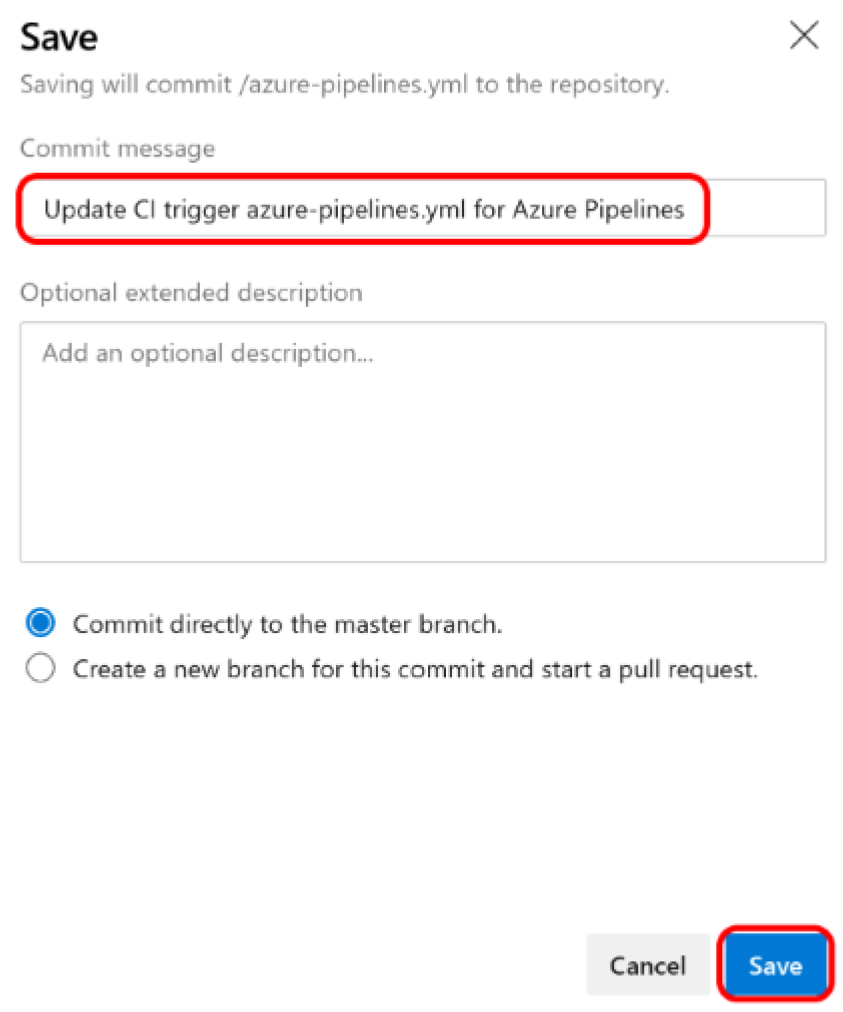


```
# ASP.NET
# Build and test ASP.NET projects.
# Add steps that publish symbols, save build artifacts, deploy, and more:
# https://docs.microsoft.com/azure/devops/pipelines/apps/aspnet/build-aspnet-4

trigger:
- master

pool:
  name: Hosted VS2017
  demands:
```

12. When you are ready to test the changes, click **Save**. Saving will commit the changes and now also initiate a new build due to the trigger that we just modified.



13. Click **Save** again to save the changes and trigger a new build.



Save

Saving will commit /azure-pipelines.yml to the repository.

Commit message

Update CI trigger azure-pipelines.yml for Azure Pipelines

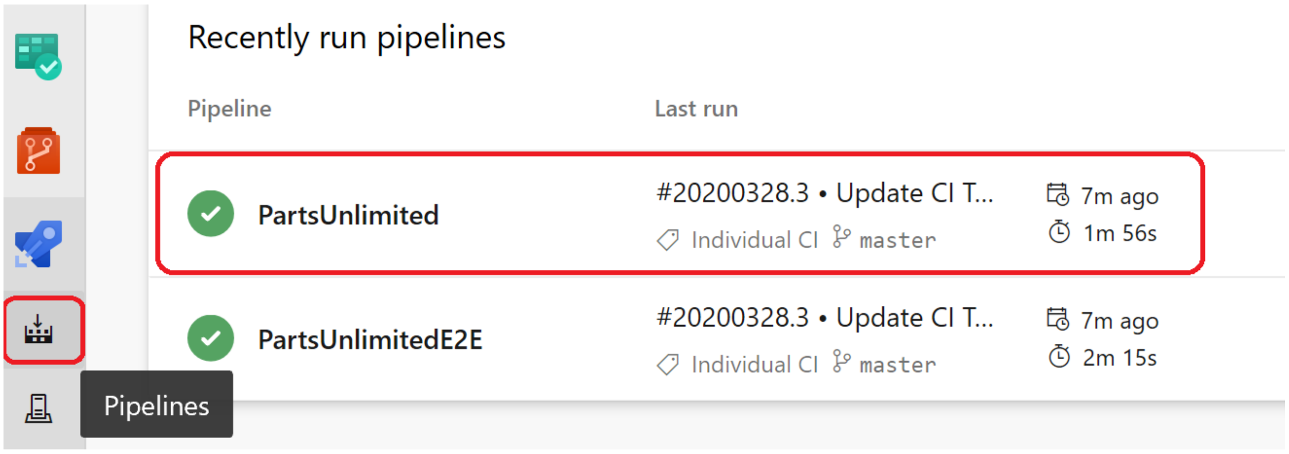
Optional extended description

Add an optional description...

[✓ Commit directly to the master branch.]

Create a new branch for this commit and start a pull request.

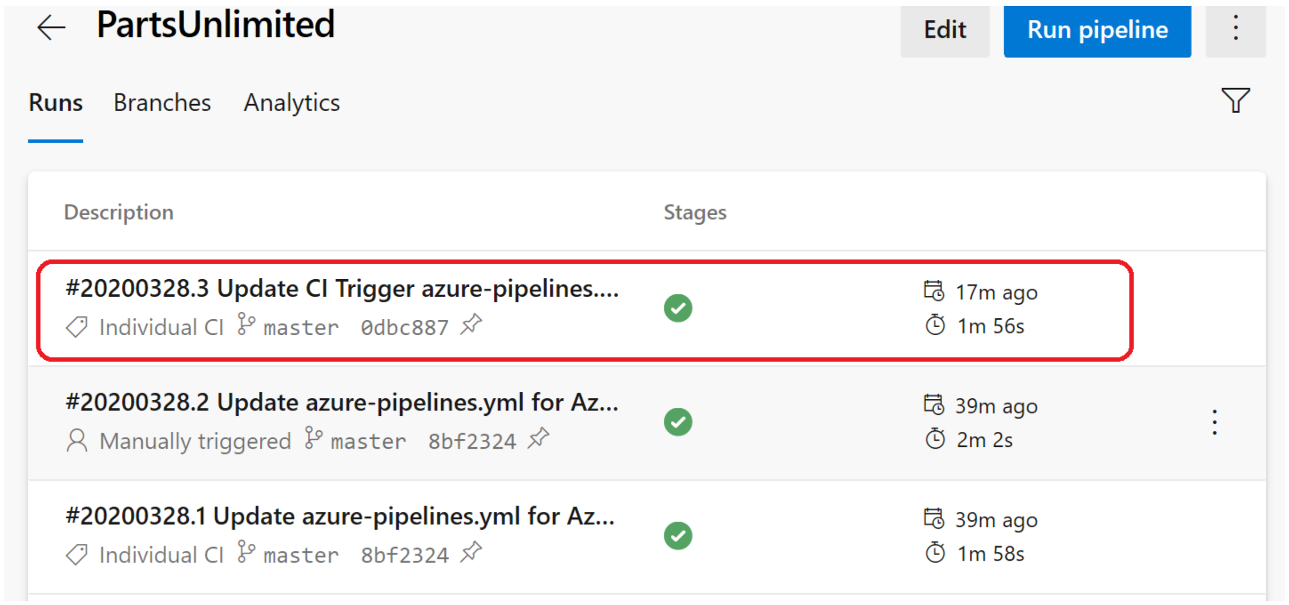
14. Let's check for our new build. Return to the **Build Pipelines** view.



Recently run pipelines

Pipeline	Last run	
PartsUnlimited	#20200328.3 • Update CI T... Individual CI master	7m ago 1m 56s
PartsUnlimitedE2E	#20200328.3 • Update CI T... Individual CI master	7m ago 2m 15s

15. Ensure your YAML pipeline is selected. In the build history, there should be a new build with our commit message. Click the build run to open it.

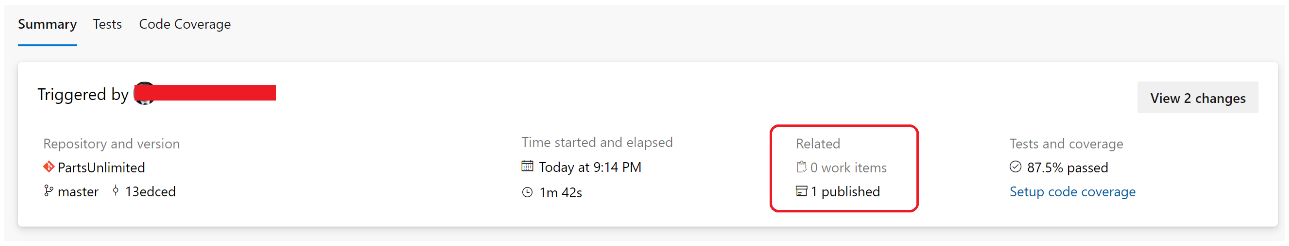


#20200328.3 Update CI Trigger azure-pipelines.... Individual CI master 0dbc887 — 17m ago, 1m 56s

#20200328.2 Update azure-pipelines.yml for Az... Manually triggered master 8bf2324 — 39m ago, 2m 2s

#20200328.1 Update azure-pipelines.yml for Az... Individual CI master 8bf2324 — 39m ago, 1m 58s

16. Again, review the build logs and wait for the build to complete. Once the build succeeds, view the newly created artifacts by clicking on **1 Published** next to the section that says **related**.



Triggered by [redacted]

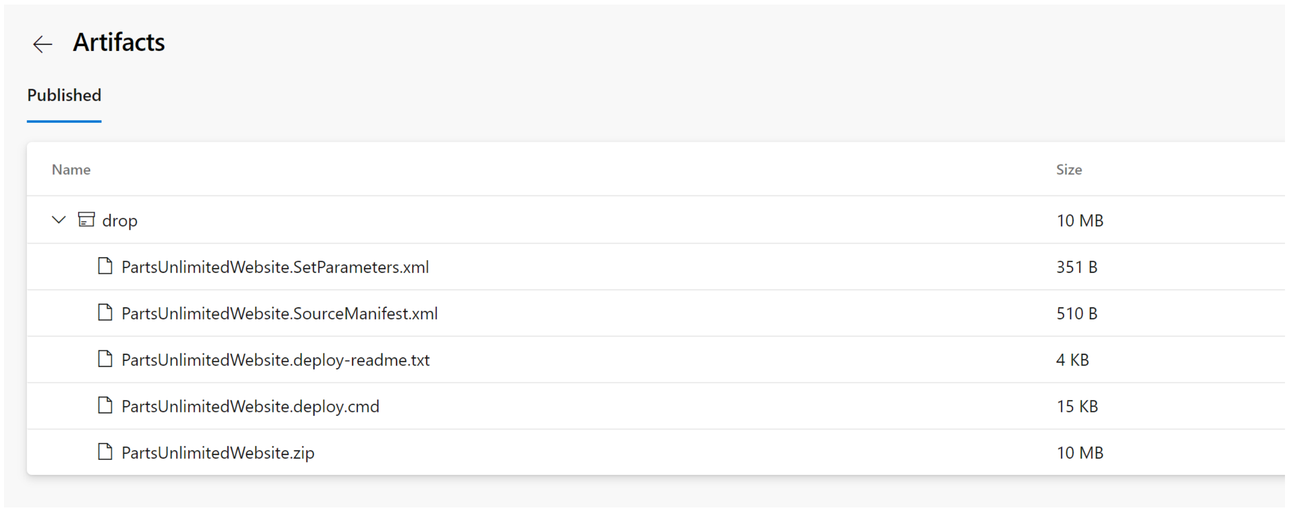
Repository and version: PartsUnlimited, master, 13edced

Time started and elapsed: Today at 9:14 PM, 1m 42s

Related: 0 work items, 1 published

Tests and coverage: 87.5% passed, Setup code coverage

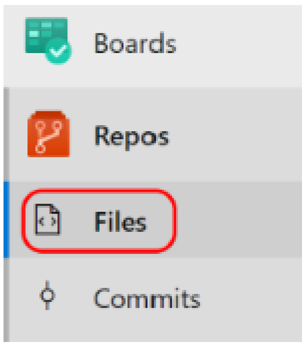
17. Expand the drop folder to view the contents.



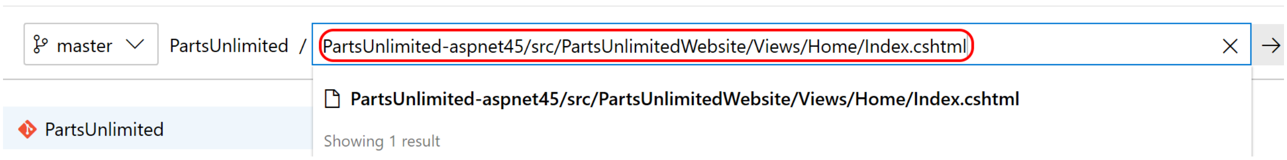
Name	Size
drop	10 MB
PartsUnlimitedWebsite.SetParameters.xml	351 B
PartsUnlimitedWebsite.SourceManifest.xml	510 B
PartsUnlimitedWebsite.deploy-readme.txt	4 KB
PartsUnlimitedWebsite.deploy.cmd	15 KB
PartsUnlimitedWebsite.zip	10 MB

Task 3: Invoking a Continuous Integration Build

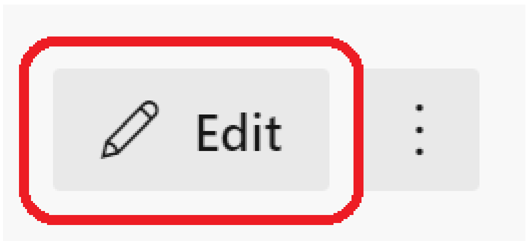
1. As you saw in the previous task, the pipeline was configured to support continuous integration as part of the template. In this section, we will trigger a build as part of a code change. Navigate to the code for this project using **Repos | Files**.

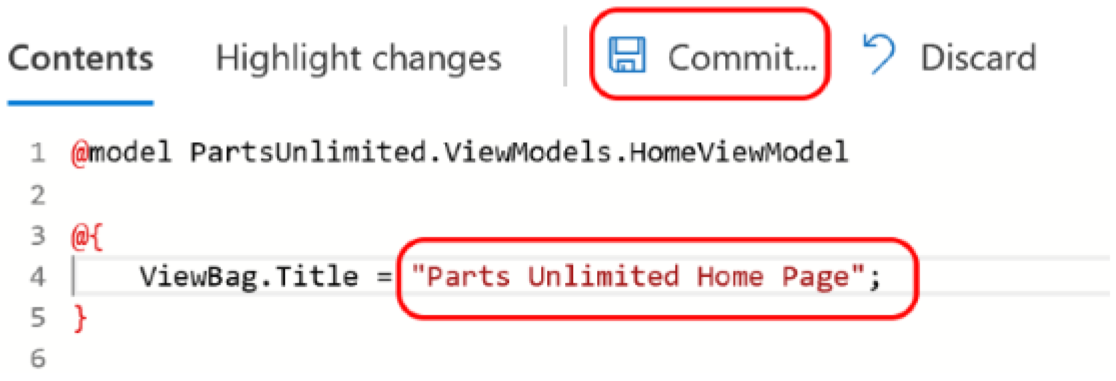
2. Open the file at **PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml**

   
   master
   PartsUnlimited /
   PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml
   PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml
   PartsUnlimited
   Showing 1 result

3. Click **Edit**.

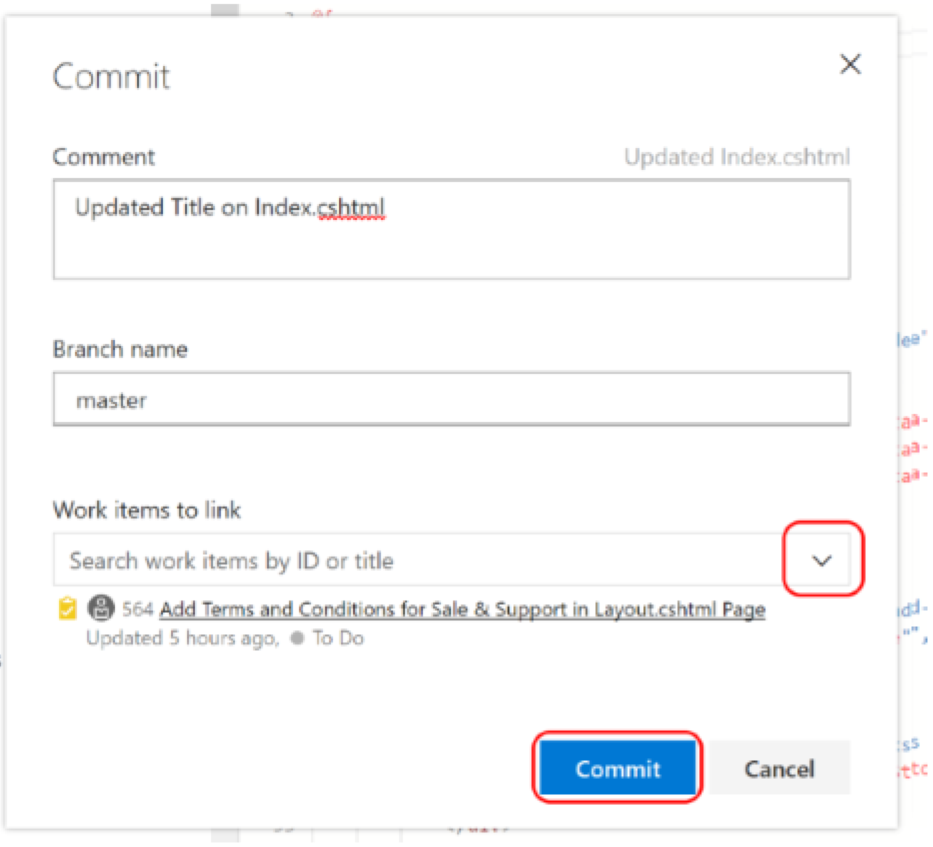
   

4. Make a minor cosmetic change, such as by tweaking the title of the document. Click **Commit**.

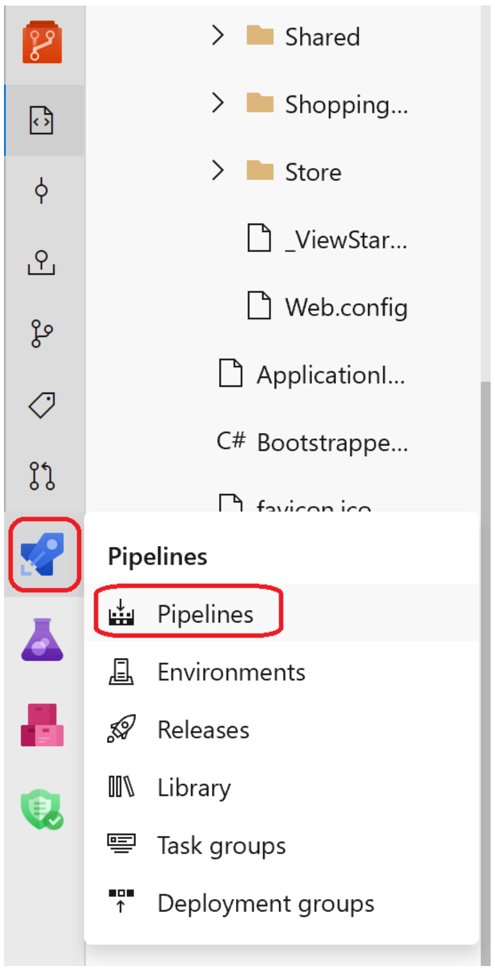
   

   ```
   @model PartsUnlimited.ViewModels.HomeViewModel

   @{
       ViewBag.Title = "Parts Unlimited Home Page";
   }

   ```

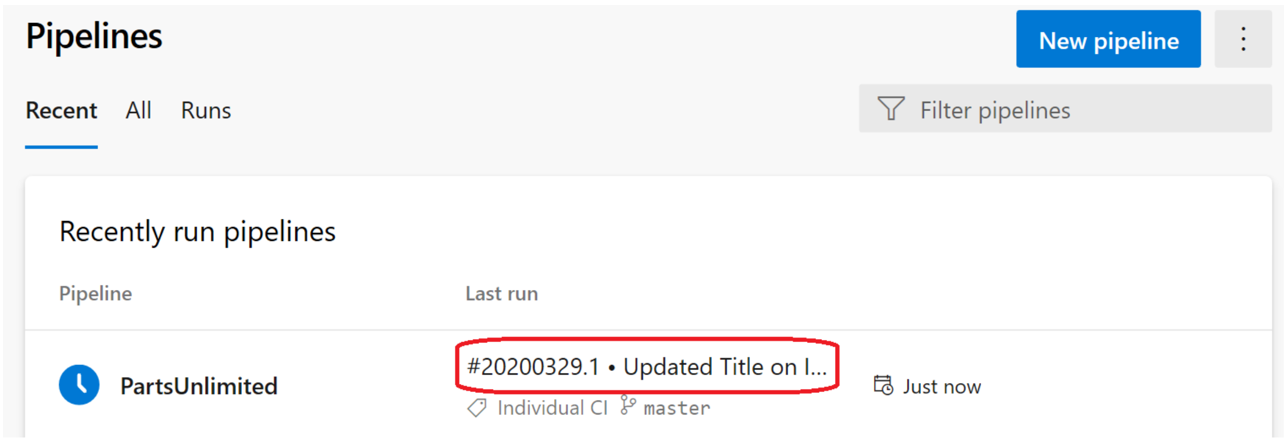
5. Add a **comment** and **link a work item** by expanding the drop down. For the purpose of the demo, you can choose any work item. We want to link a work item to show traceability in the later sections. Click **Commit**.


Commit
Comment
Updated Index.cshtml
Updated Title on Index.cshtml
Branch name
master
Work items to link
Search work items by ID or title
564 Add Terms and Conditions for Sale & Support in Layout.cshtml Page
Updated 5 hours ago, To Do
Commit
Cancel

6. A build should be triggered shortly. Select **Pipelines | Pipelines** to see if it's in progress.



7. You should now see that a new build is in progress and that it was triggered by your change. Click the build to track it.


Pipelines

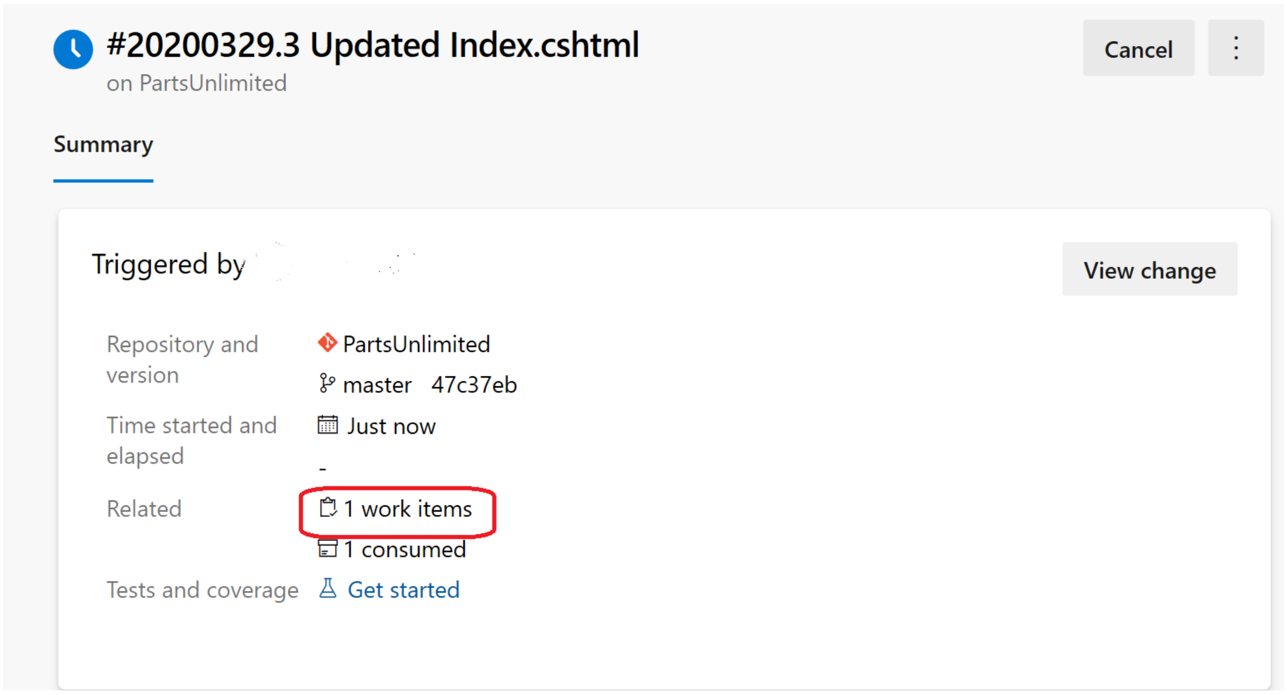
Recent All Runs

Filter pipelines

Recently run pipelines

Pipeline	Last run	
PartsUnlimited	#20200329.1 • Updated Title on I... Individual CI master	Just now

8. This build should run and succeed just like the previous build. Under the **Related** section we will see the linked work item.


#20200329.3 Updated Index.cshtml
on PartsUnlimited

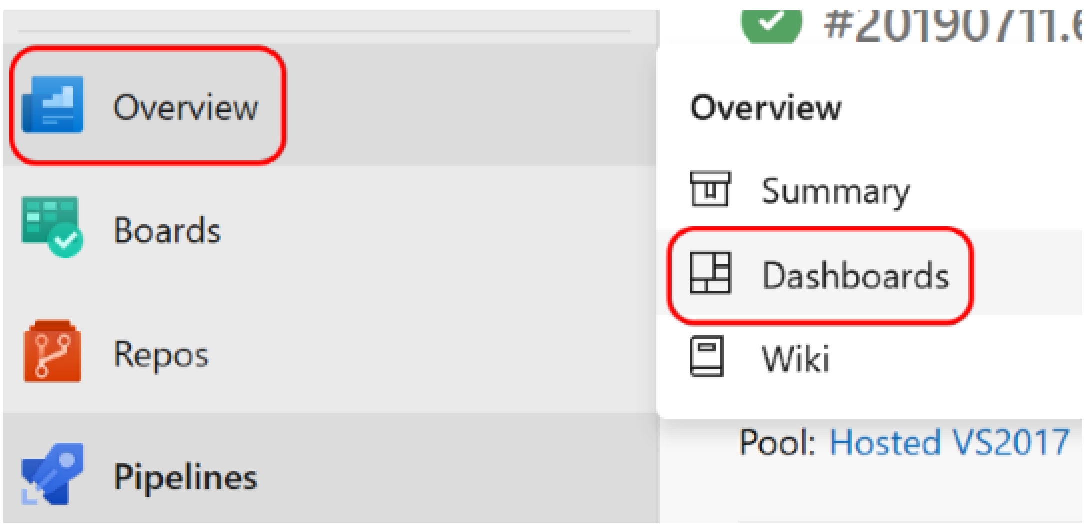
Summary

Triggered by

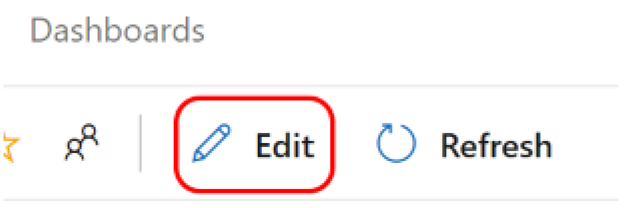
Repository and version	PartsUnlimited master 47c37eb
Time started and elapsed	Just now -
Related	1 work items 1 consumed
Tests and coverage	Get started

Task 4: Monitor Code and Build KPIs

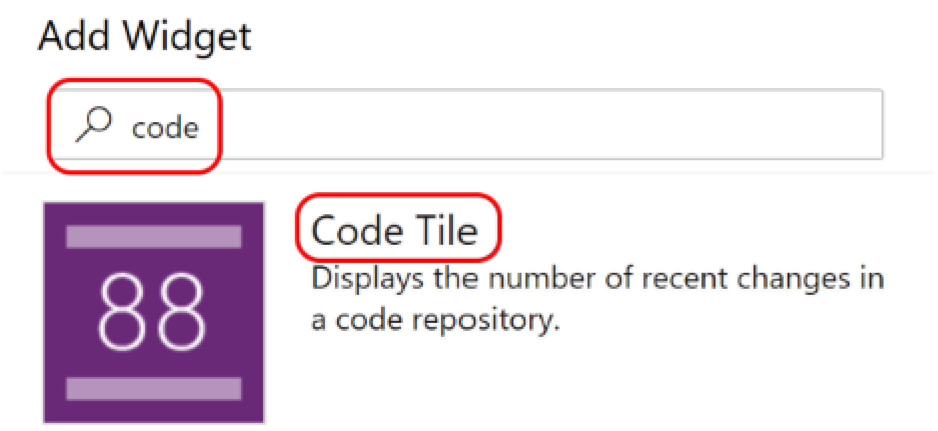
1. Next, navigate to the Team Overview dashboard by clicking **Overview | Dashboards** from the left hand menu.



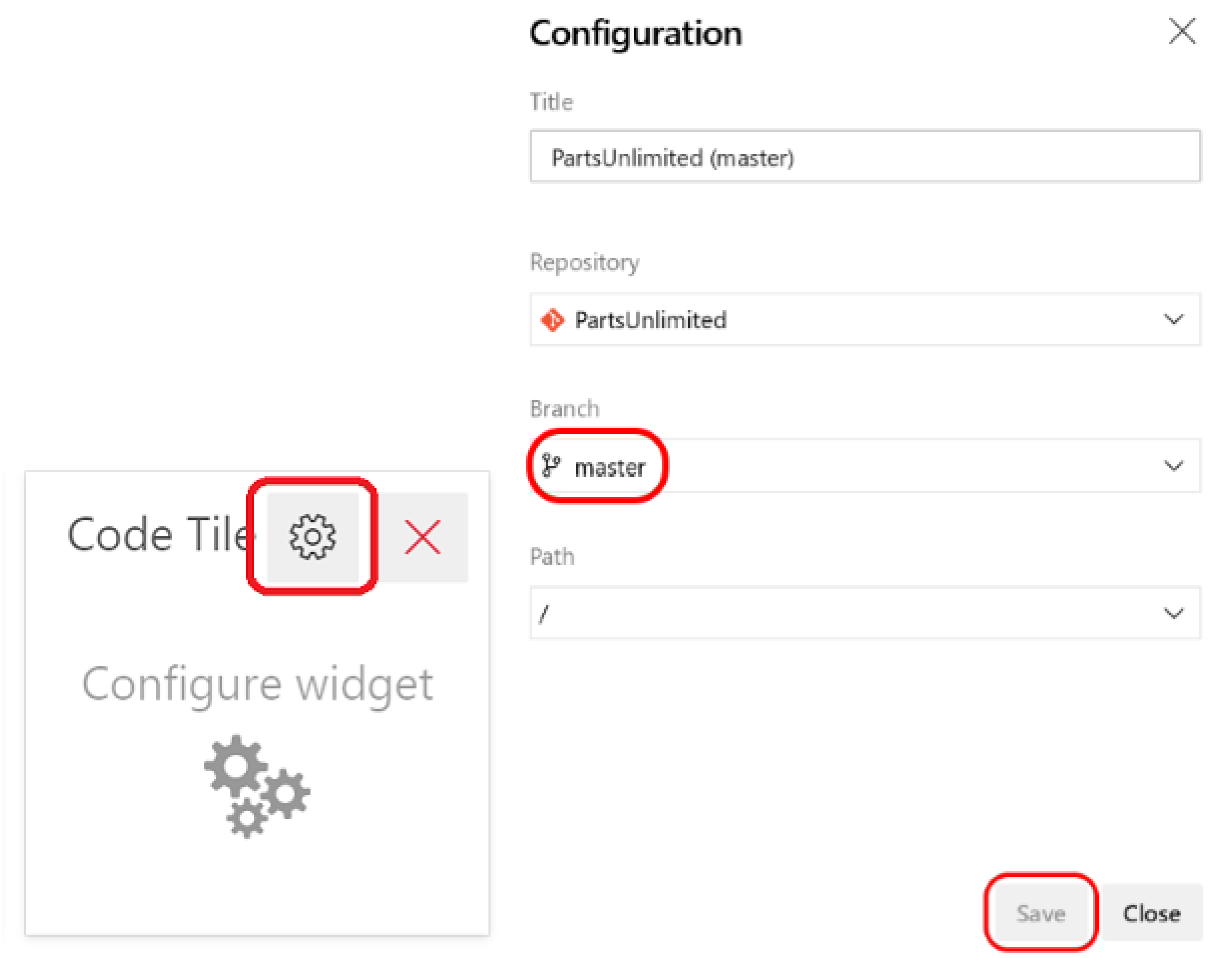
2. Click Edit. Note: If it's your first time navigating to the Dashboard area, you may have to additionally select the Overview Dashboard before you can click Edit.



3. Add the **Code Tile** widget.



4. Configure the widget and point it to the master branch. Click **Save**.



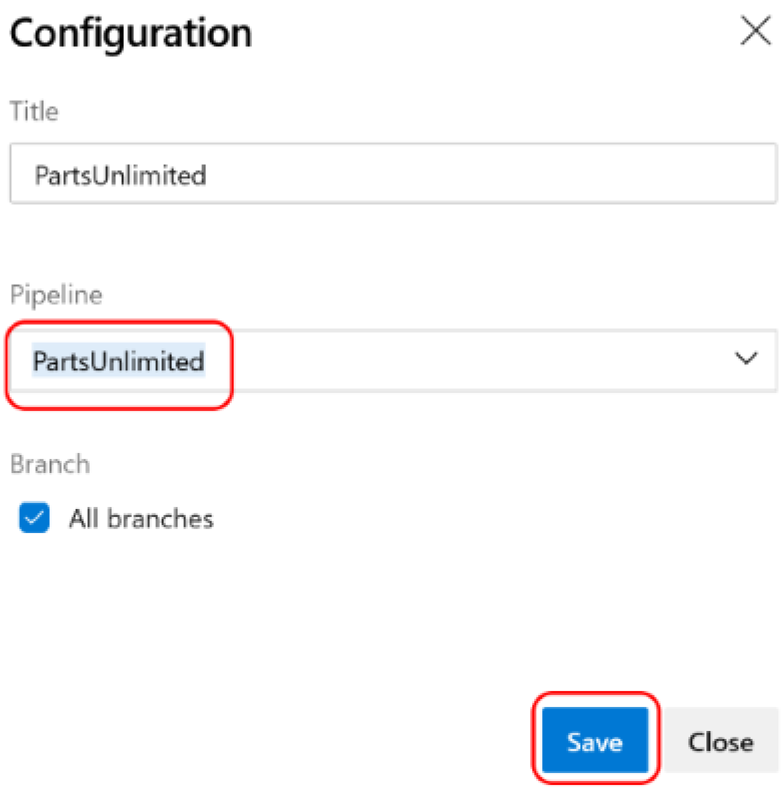
Title: PartsUnlimited (master)

Repository: PartsUnlimited

Branch: master

Path: /

5. Next, add the **Build History** widget and point it to the build pipeline you created. Click **Save**.

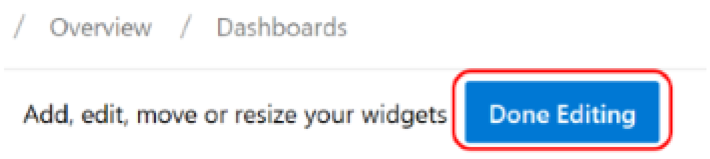


Title: PartsUnlimited

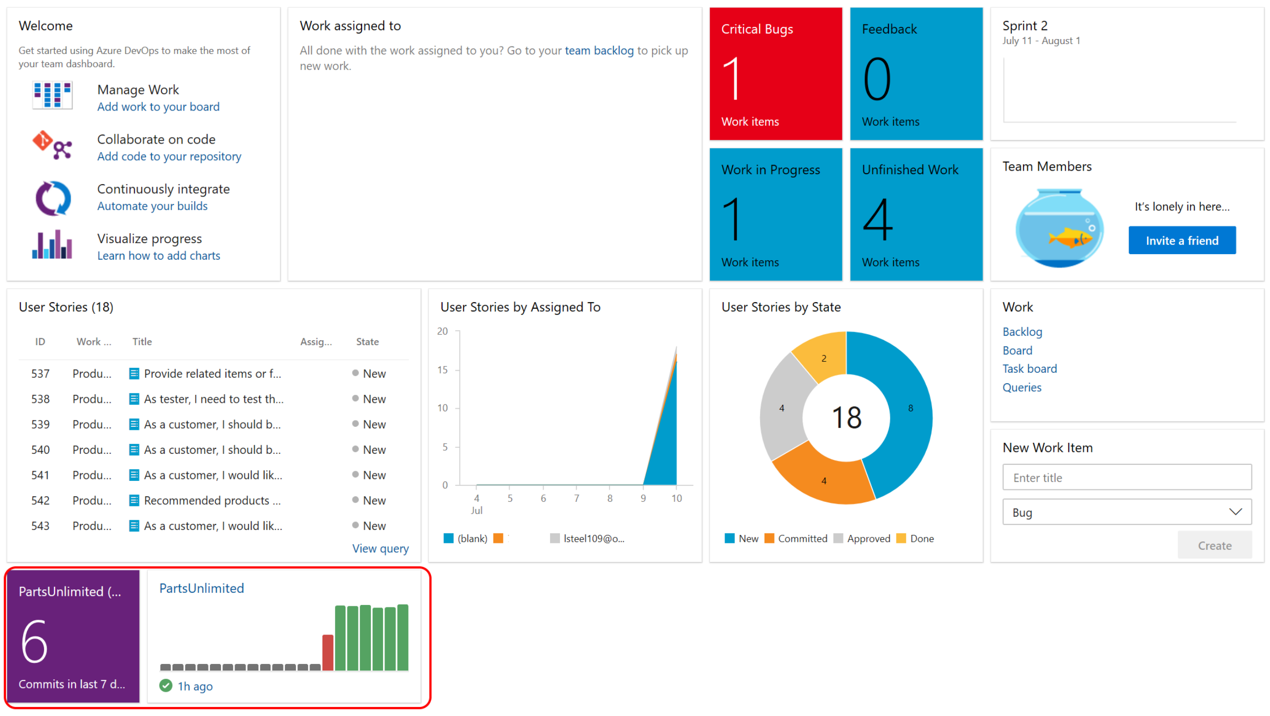
Pipeline: PartsUnlimited

Branch: ✓ All branches

6. Click **Done Editing** near the top.



7. The team dashboard should now look similar to the below:


Welcome

Get started using Azure DevOps to make the most of your team dashboard.

Manage Work
Add work to your board

Collaborate on code
Add code to your repository

Continuously integrate
Automate your builds

Visualize progress
Learn how to add charts

Work assigned to

All done with the work assigned to you? Go to your team backlog to pick up new work.

Critical Bugs 1 Work items

Feedback 0 Work items

Work in Progress 1 Work items

Unfinished Work 4 Work items

Sprint 2
July 11 - August 1

Team Members
It's lonely in here...
Invite a friend

User Stories (18)

ID	Work ...	Title	Assig...	State
537	Produ...	Provide related items or f...		New
538	Produ...	As tester, I need to test th...		New
539	Produ...	As a customer, I should b...		New
540	Produ...	As a customer, I should b...		New
541	Produ...	As a customer, I would lik...		New
542	Produ...	Recommended products ...		New
543	Produ...	As a customer, I would lik...		New

View query

Work
Backlog
Board
Task board
Queries

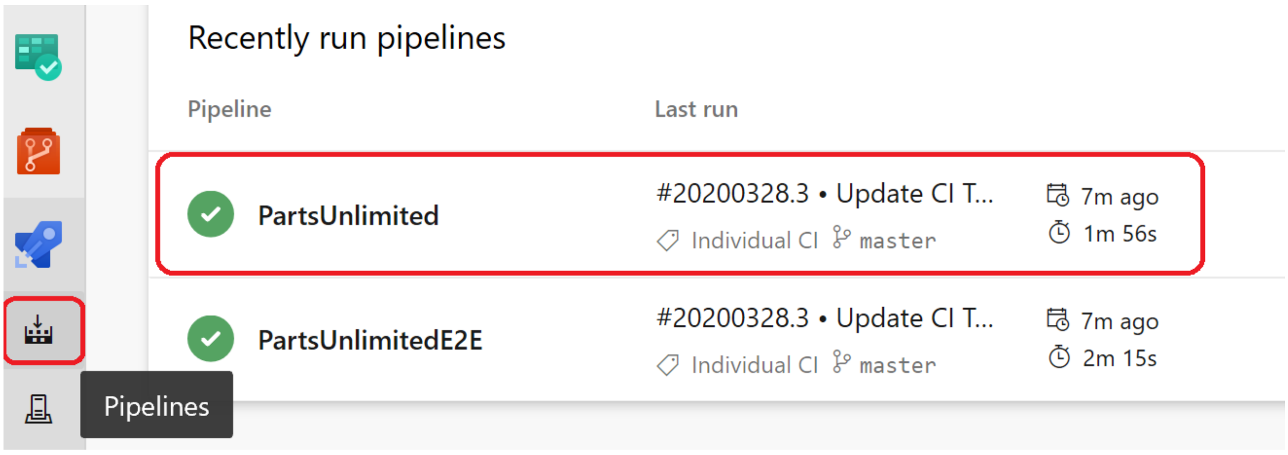
New Work Item
Enter title
Bug
Create

PartsUnlimited (...
6
Commits in last 7 d...

PartsUnlimited
1h ago

Task 5: Publish Pipeline Artifacts

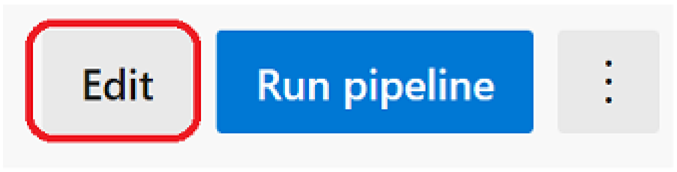
1. In this task, we will copy the ARM templates from the Azure Repo to our pipeline artifacts drop location. Navigate back to the YAML pipeline by clicking **Pipelines | Pipelines** and selecting the YAML pipeline we have been working with.



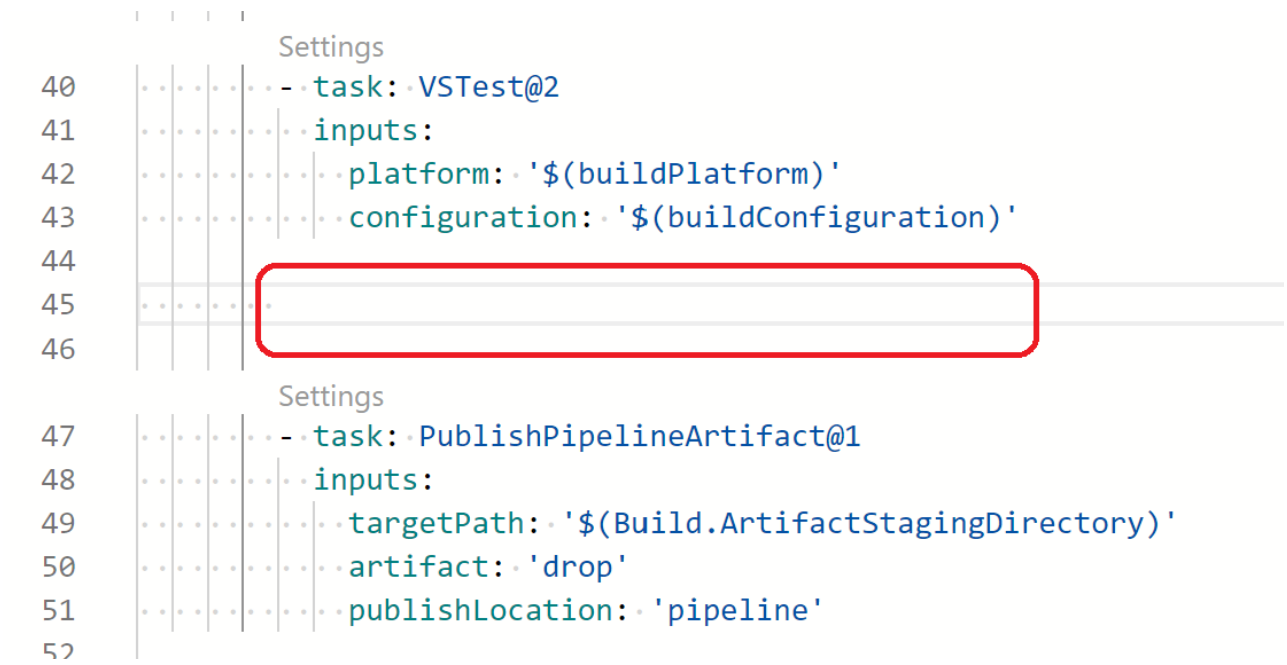
Recently run pipelines

Pipeline	Last run	
PartsUnlimited	#20200328.3 • Update CI T... Individual CI master	7m ago 1m 56s
PartsUnlimitedE2E	#20200328.3 • Update CI T... Individual CI master	7m ago 2m 15s

2. Next, click **Edit**.

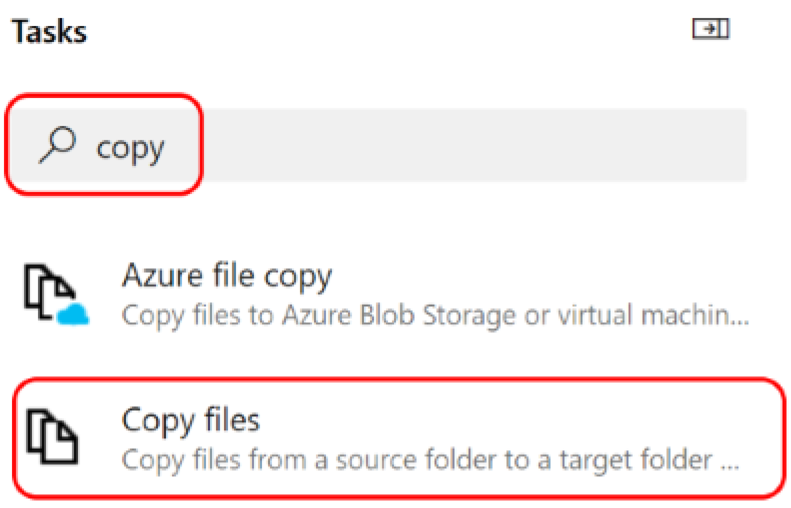


3. Between the **VS Test** and **Publish Pipeline Artifact** tasks, add two new blank lines and place your cursor in the middle. This will help make the tasks more readable as we add the next task from the library.



```
                Settings
                - task: VSTest@2
                  inputs:
                    platform: '$(buildPlatform)'
                    configuration: '$(buildConfiguration)'



                Settings
                - task: PublishPipelineArtifact@1
                  inputs:
                    targetPath: '$(Build.ArtifactStagingDirectory)'
                    artifact: 'drop'
                    publishLocation: 'pipeline'
```

4. From the tasks library on the right, search for **copy** and select the **Copy Files** task.


Tasks

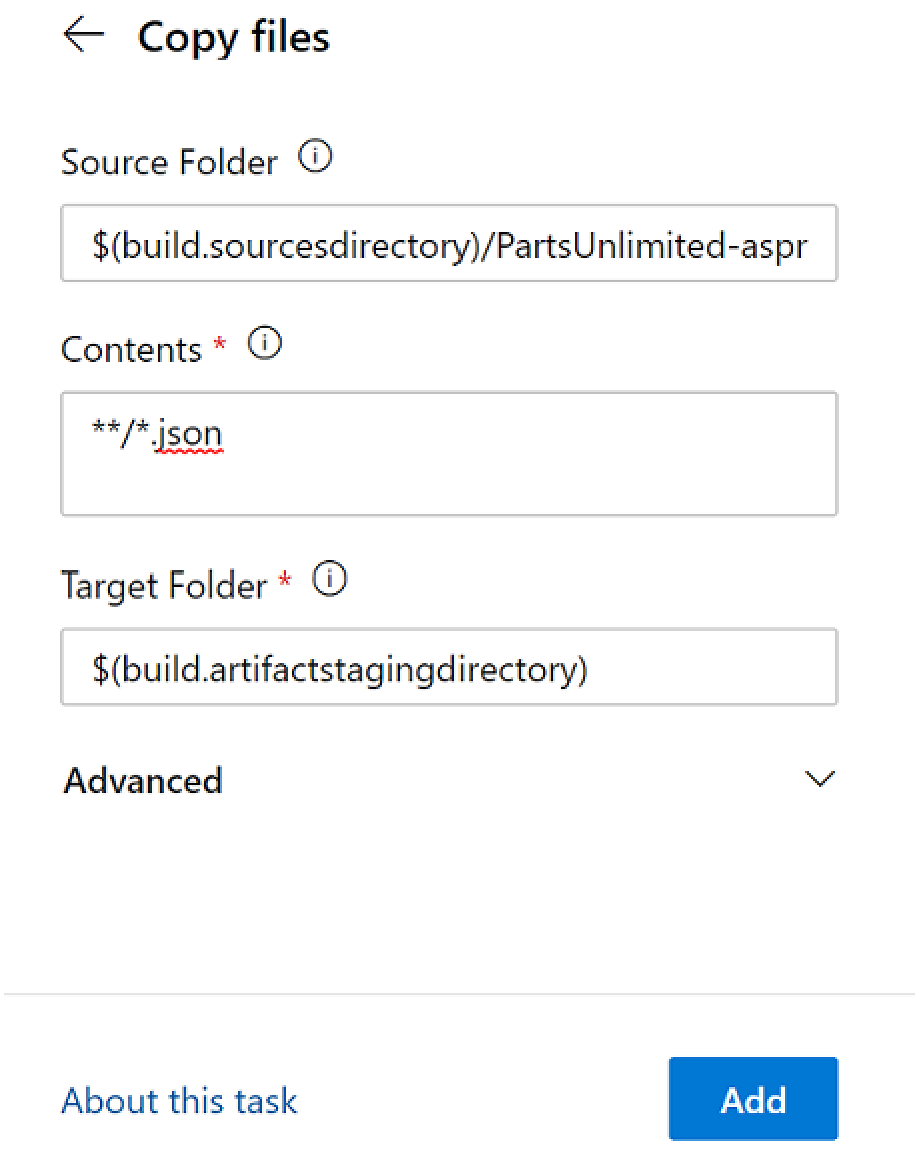
copy

Azure file copy
Copy files to Azure Blob Storage or virtual machin...

Copy files
Copy files from a source folder to a target folder ...

5. Add the following information. The contents filter will select all json files in the env folder and its subdirectories and copy them to the build artifacts folder. Click **Add**.

   SourceFolder: $(build.sourcesdirectory)/PartsUnlimited-aspnet45/env/
   Contents: **/*.json
   TargetFolder: $(build.artifactstagingdirectory)


Copy files

Source Folder: $(build.sourcesdirectory)/PartsUnlimited-aspr

Contents *: **/*.json

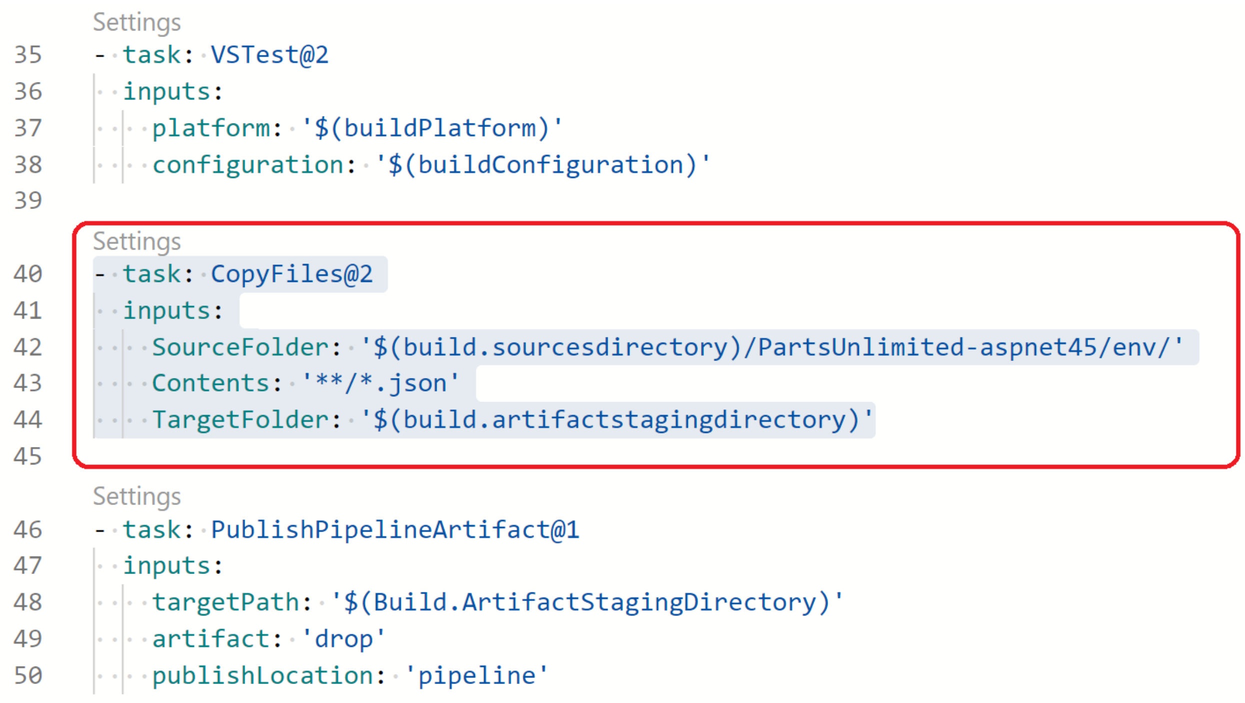
Target Folder *: $(build.artifactstagingdirectory)

Advanced

About this task

Add

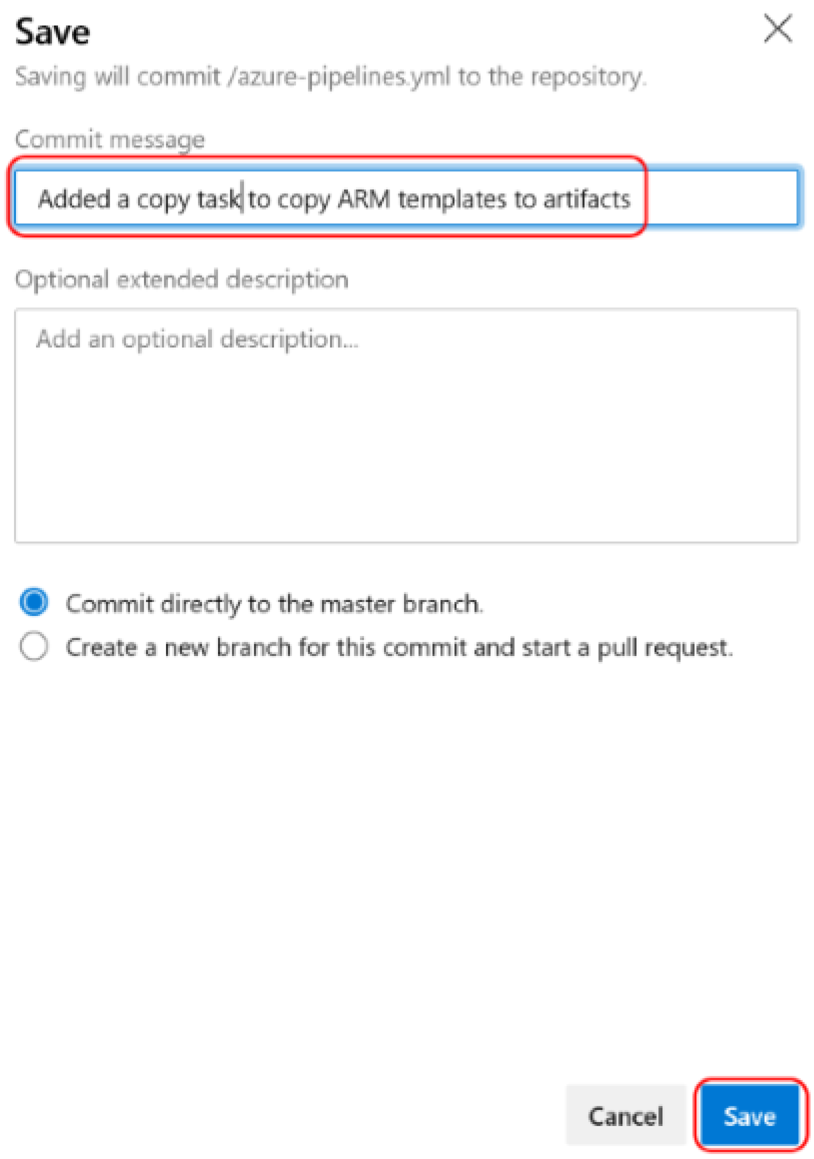
6. The YAML pipeline should now contain a copy files task as shown below.



```
Settings
- task: VSTest@2
  inputs:
    platform: '$(buildPlatform)'
    configuration: '$(buildConfiguration)'

Settings
- task: CopyFiles@2
  inputs:
    SourceFolder: '$(build.sourcesdirectory)/PartsUnlimited-aspnet45/env/'
    Contents: '**/*.json'
    TargetFolder: '$(build.artifactstagingdirectory)'

Settings
- task: PublishPipelineArtifact@1
  inputs:
    targetPath: '$(Build.ArtifactStagingDirectory)'
    artifact: 'drop'
    publishLocation: 'pipeline'
```

7. When you are ready to test the changes, click **Save**.

8. Enter a commit message and click **Save** again.


Save

Saving will commit /azure-pipelines.yml to the repository.

Commit message

Added a copy task to copy ARM templates to artifacts

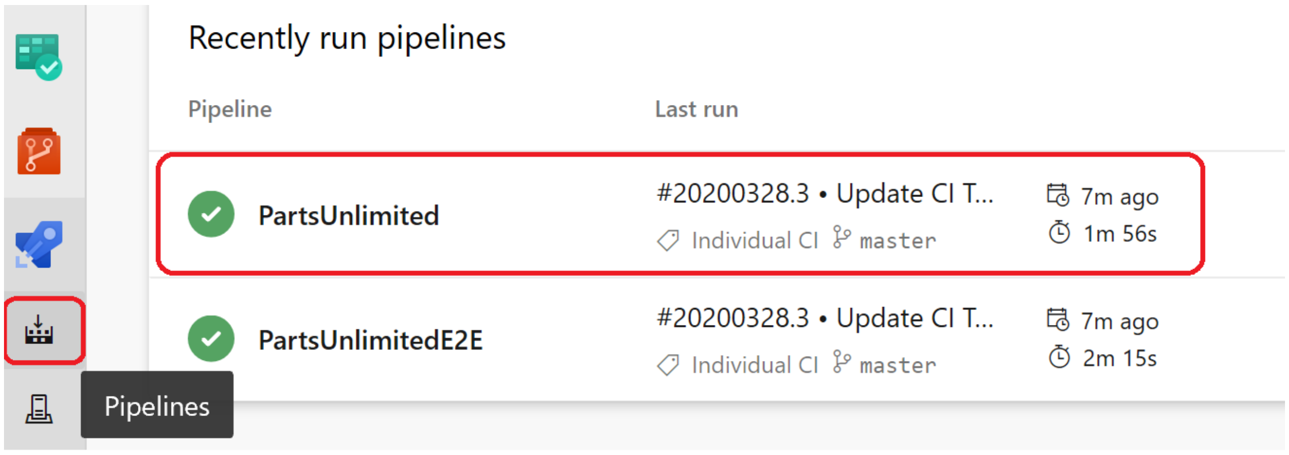
Optional extended description

Add an optional description...

[✓ Commit directly to the master branch.]

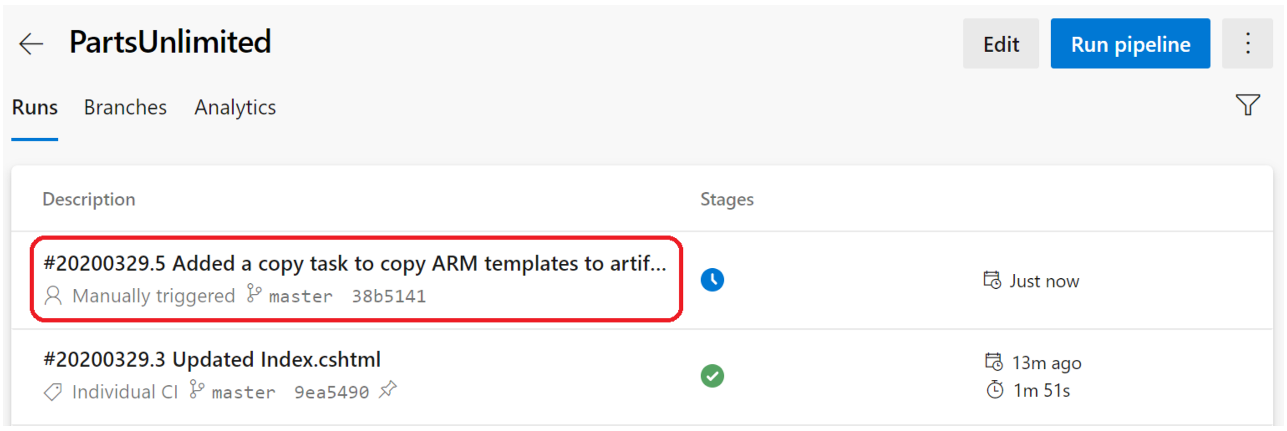
Create a new branch for this commit and start a pull request.

9. A new build should be triggered. Return to the **Build Pipelines** view to check.


Recently run pipelines

Pipeline	Last run	
PartsUnlimited	#20200328.3 • Update CI T... Individual CI master	7m ago 1m 56s
PartsUnlimitedE2E	#20200328.3 • Update CI T... Individual CI master	7m ago 2m 15s

10. Ensure your YAML pipeline is selected. In the build history, there should be a new build with our commit message. Click the build run to open it.



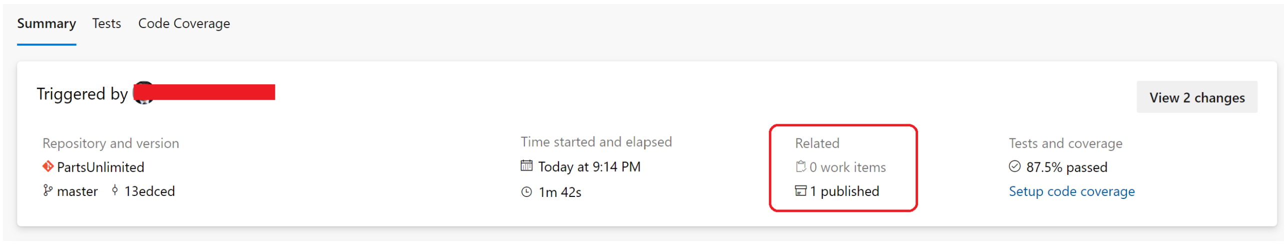
PartsUnlimited

Edit | Run pipeline

Runs | Branches | Analytics

Description	Stages	
#20200329.5 Added a copy task to copy ARM templates to artif... Manually triggered master 38b5141		Just now
#20200329.3 Updated Index.cshtml Individual CI master 9ea5490	✓	13m ago 1m 51s

11. Again, review the build logs and wait for the build to complete. Once the build succeeds, view the newly created artifacts by clicking on **1 published**.



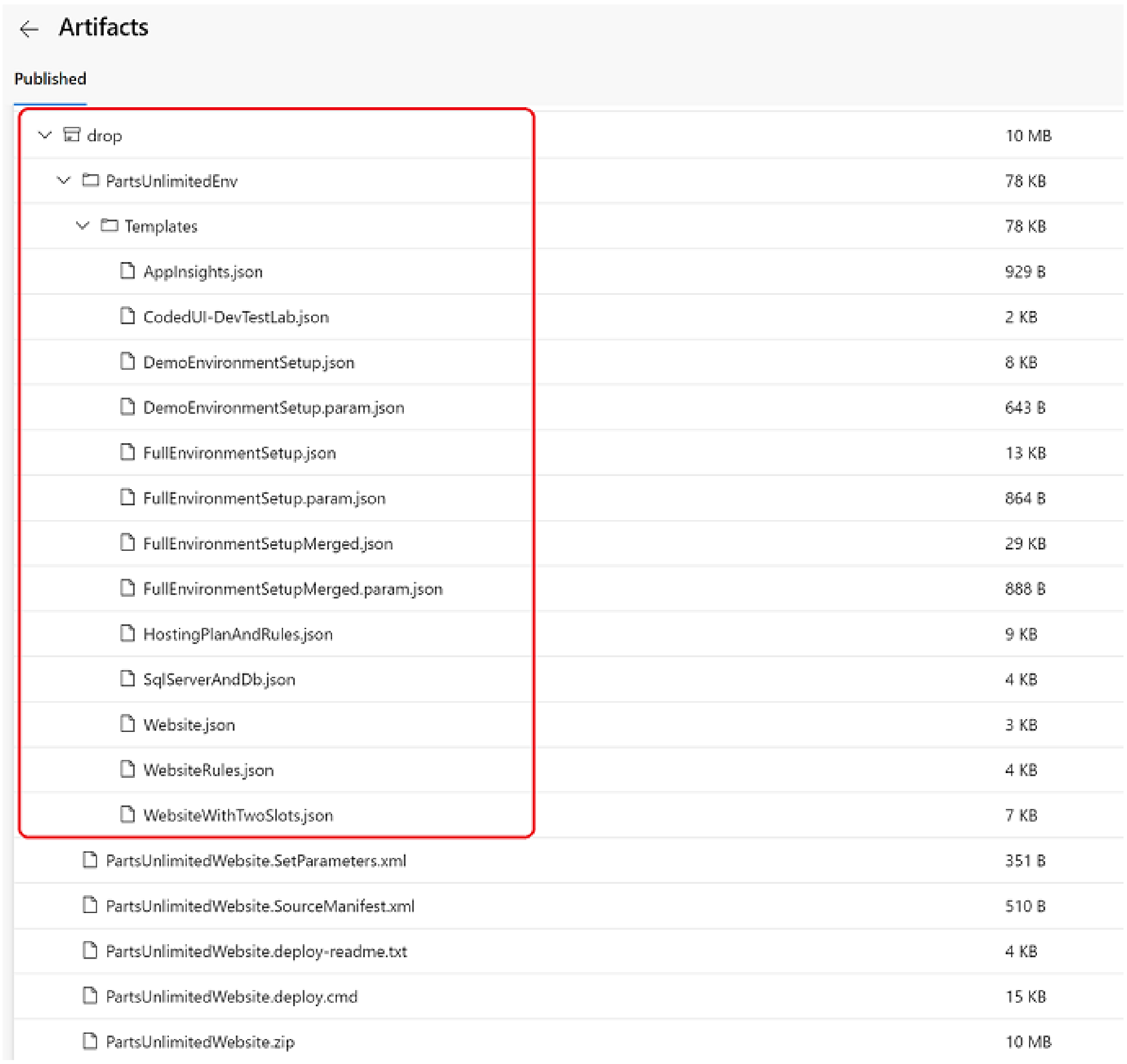
Summary | Tests | Code Coverage

Triggered by [redacted]

View 2 changes

Repository and version	Time started and elapsed	Related	Tests and coverage
PartsUnlimited	Today at 9:14 PM	0 work items	87.5% passed
master 13edced	1m 42s	1 published	Setup code coverage

12. Expand the drop folder to view the contents. We should see a folder with the ARM templates.



Artifacts

Published

drop	10 MB
PartsUnlimitedEnv	78 KB
Templates	78 KB
AppInsights.json	929 B
CodedUI-DevTestLab.json	2 KB
DemoEnvironmentSetup.json	8 KB
DemoEnvironmentSetup.param.json	643 B
FullEnvironmentSetup.json	13 KB
FullEnvironmentSetup.param.json	864 B
FullEnvironmentSetupMerged.json	29 KB
FullEnvironmentSetupMerged.param.json	888 B
HostingPlanAndRules.json	9 KB
SqlServerAndDb.json	4 KB
Website.json	3 KB
WebsiteRules.json	4 KB
WebsiteWithTwoSlots.json	7 KB
PartsUnlimitedWebsite.SetParameters.xml	351 B
PartsUnlimitedWebsite.SourceManifest.xml	510 B
PartsUnlimitedWebsite.deploy-readme.txt	4 KB
PartsUnlimitedWebsite.deploy.cmd	15 KB
PartsUnlimitedWebsite.zip	10 MB