Enabling Continuous Integration with Azure Pipelines

Overview

In this lab, you will learn how to configure continuous integration (CI) and continuous deployment (CD) for your applications using Build and Release in Azure Pipelines. This scriptable CI/CD system is both web-based and cross-platform, while also providing a modern interface for visualizing sophisticated workflows. Although we won't demonstrate all of the cross-platform possibilities in this lab, it is important to point out that you can also build for iOS, Android, Java (using Ant, Maven, or Gradle) and Linux.

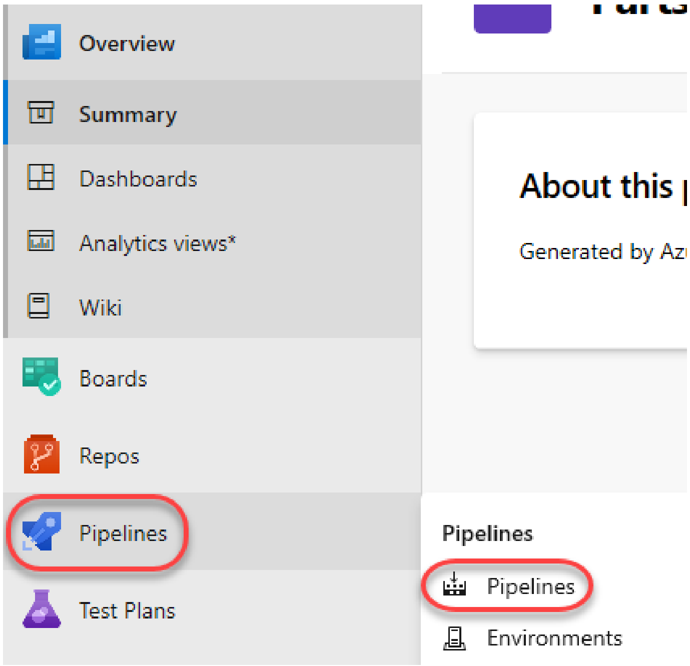
Prerequisites

- This lab requires you to complete task 1 from the prerequisite instructions.

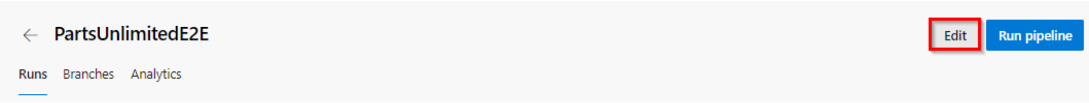
Exercise 1: Introduction to Azure DevOps Build

Task 1: Creating a basic build pipeline from a template

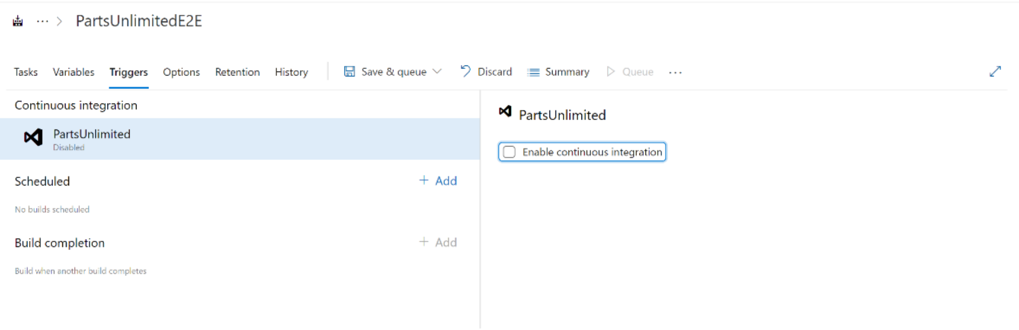
1. Navigate to your team project on Azure DevOps.
2. Navigate to **Pipelines | Pipelines**.



3. Open (click) the existing **PartsUnlimitedE2E** pipeline (which was created by the demo generator tool) and click on **Edit**



4. Not to have two pipelines triggered later in the lab, disable the CI trigger for the template created pipeline (uncheck) and **Save**.


··· > PartsUnlimitedE2E

Tasks Variables Triggers Options Retention History | Save & queue Discard Summary Queue ···

Continuous integration

PartsUnlimited
Disabled

Scheduled + Add

No builds scheduled

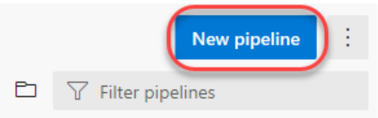
Build completion + Add

Build when another build completes

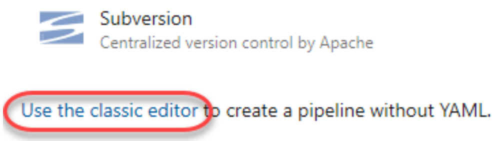
PartsUnlimited

□ Enable continuous integration

5. Navigate back to **Pipelines | Pipelines** and click **New pipeline** to create a new build pipeline.

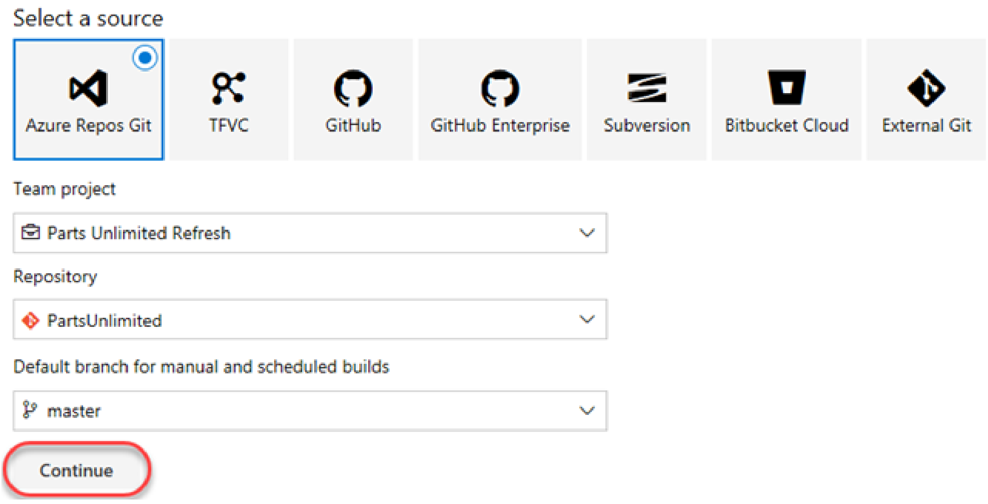


6. The default option for build pipelines involves using YAML to define the process. If you are interested in that, please check out that lab. For this lab, click **use the classic editor**.


Subversion
Centralized version control by Apache

Use the classic editor to create a pipeline without YAML.

7. The first thing you'll need to do is to configure the source repository. Every major platform is available, but the default options are all we need here. This build will use the **master** branch of the **PartsUnlimited** repo. Leave the defaults and click **Continue**.


Select a source

● Azure Repos Git | TFVC | GitHub | GitHub Enterprise | Subversion | Bitbucket Cloud | External Git

Team project

Parts Unlimited Refresh

Repository

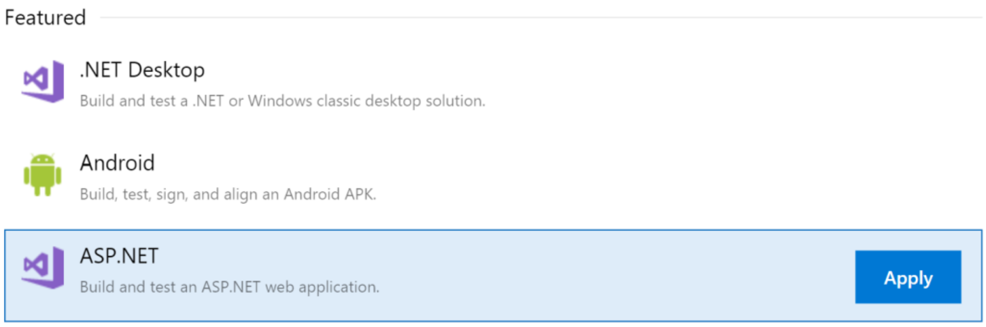
PartsUnlimited

Default branch for manual and scheduled builds

master

Continue

8. Locate the **ASP.NET** template and click **Apply** to apply this template to the build definition. Note that there are many options that should cover all of our mainstream scenarios. For our purposes here, we'll just build the project using the baseline ASP.NET template.



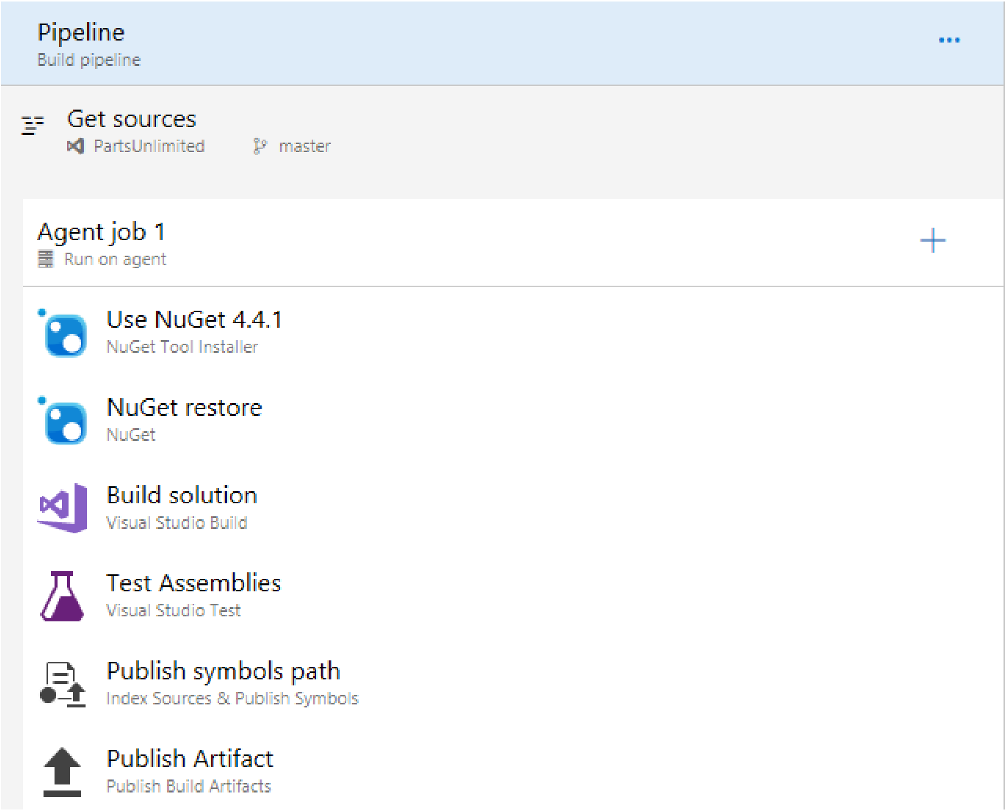
Featured

.NET Desktop
Build and test a .NET or Windows classic desktop solution.

Android
Build, test, sign, and align an Android APK.

ASP.NET
Build and test an ASP.NET web application.

9. The process for this build pipeline is easy to follow. After getting the source, Azure DevOps will use NuGet to restore any dependent packages. Then, the project will be built and tested. The results will then be published to the configured target.



Pipeline
Build pipeline

Get sources
PartsUnlimited master

Agent job 1
Run on agent

Use NuGet 4.4.1
NuGet Tool Installer

NuGet restore
NuGet

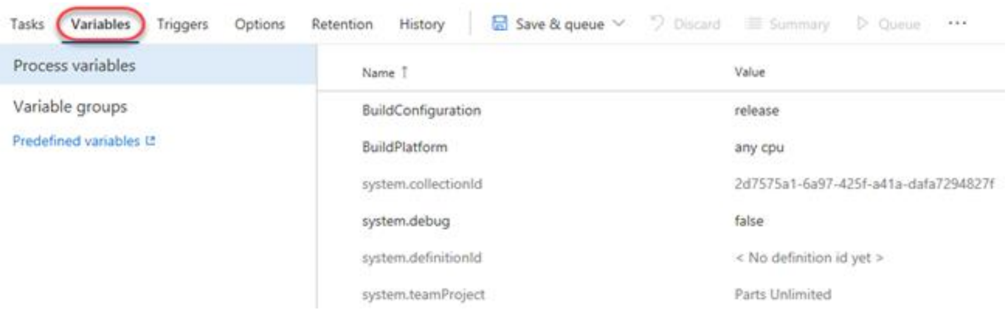
Build solution
Visual Studio Build

Test Assemblies
Visual Studio Test

Publish symbols path
Index Sources & Publish Symbols

Publish Artifact
Publish Build Artifacts

10. Select the **Variables** tab. Here you can configure special parameters to be used during the build, such as the configuration or platform.



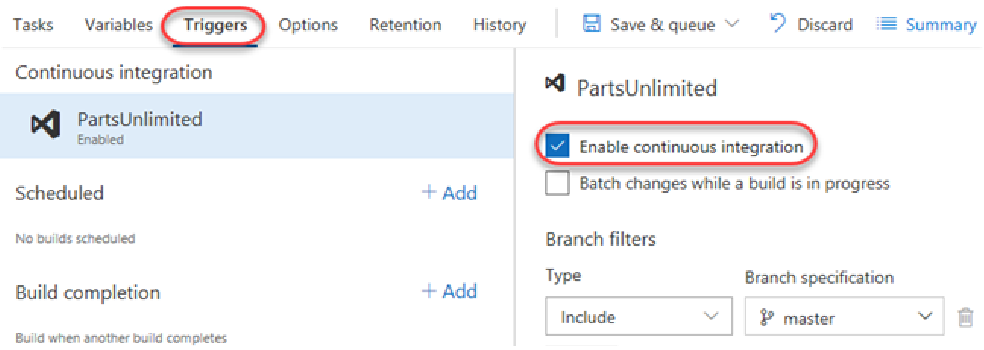
Process variables

Variable groups

Predefined variables

Name ↑	Value
BuildConfiguration	release
BuildPlatform	any cpu
system.collectionId	2d7575a1-6a97-425f-a41a-dafa7294827f
system.debug	false
system.definitionId	< No definition id yet >
system.teamProject	Parts Unlimited

11. Select the **Triggers** tab. These triggers enable you to automatically invoke builds on a schedule, when another build completes, or when changes are made to the source. Check **Enable continuous integration** so that this build will get invoked whenever source changes are committed.



Continuous integration

PartsUnlimited
Enabled

Scheduled + Add

No builds scheduled

Build completion + Add

Build when another build completes

PartsUnlimited

☑ Enable continuous integration

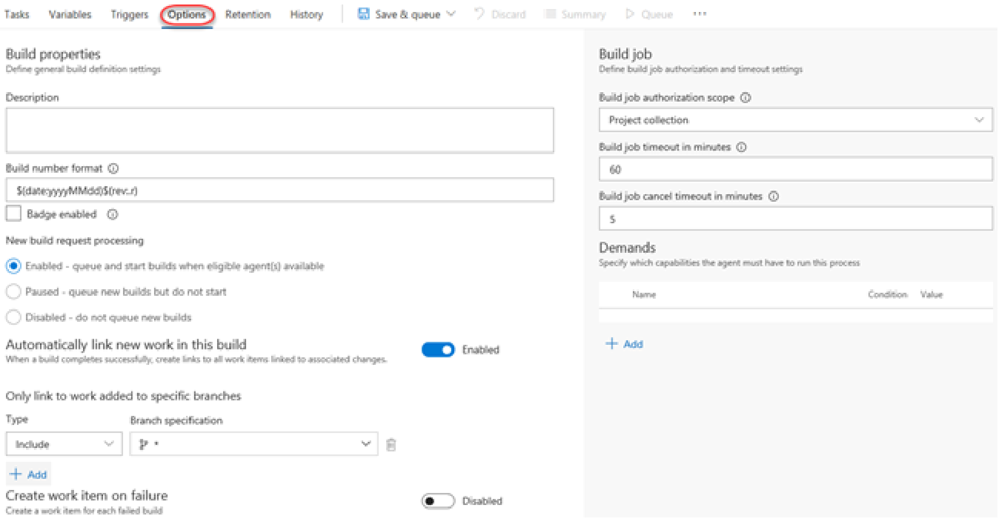
□ Batch changes while a build is in progress

Branch filters

Type: Include

Branch specification: master

12. Select the **Options** tab. This section includes a wide variety of options related to the build workflow. Note that you'll generally configure options for specific build tasks on the configuration views of the tasks themselves.



Build properties
Define general build definition settings

Description

Build number format ⓘ
$(date:yyyyMMdd)$(rev:.r)

□ Badge enabled ⓘ

New build request processing

● Enabled - queue and start builds when eligible agent(s) available

○ Paused - queue new builds but do not start

○ Disabled - do not queue new builds

Automatically link new work in this build — Enabled
When a build completes successfully, create links to all work items linked to associated changes.

Only link to work added to specific branches

Type: Include

Branch specification: *

+ Add

Create work item on failure — Disabled
Create a work item for each failed build

Build job
Define build job authorization and timeout settings

Build job authorization scope ⓘ
Project collection

Build job timeout in minutes ⓘ
60

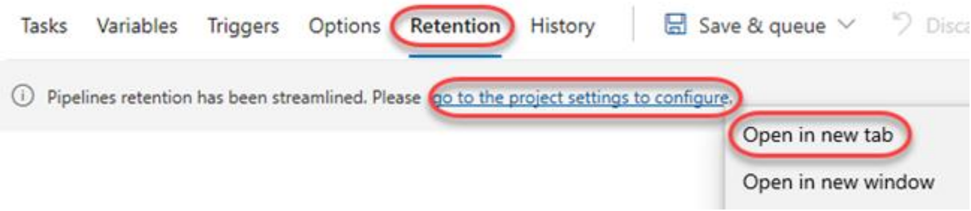
Build job cancel timeout in minutes ⓘ
5

Demands
Specify which capabilities the agent must have to run this process

Name	Condition	Value

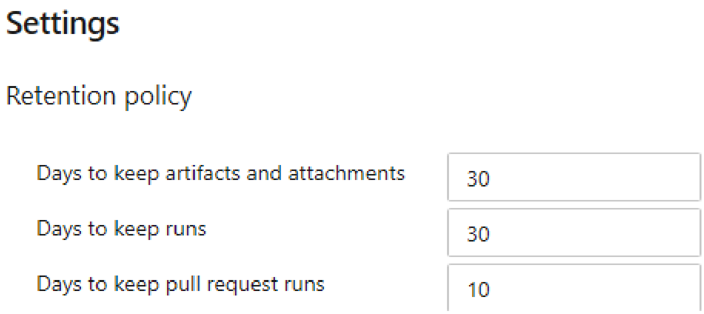
+ Add

13. Select the **Retention** tab. Right-click **go to the project settings to configure** and select **Open in new tab**.



Pipelines retention has been streamlined. Please go to the project settings to configure.

14. This section enables you to configure which pipeline runs are retained and for how long. Close the tab.

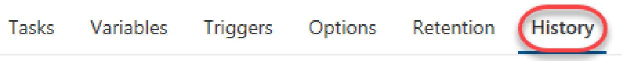


Settings

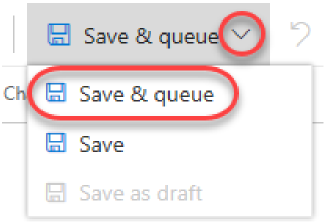
Retention policy

Days to keep artifacts and attachments	30
Days to keep runs	30
Days to keep pull request runs	10

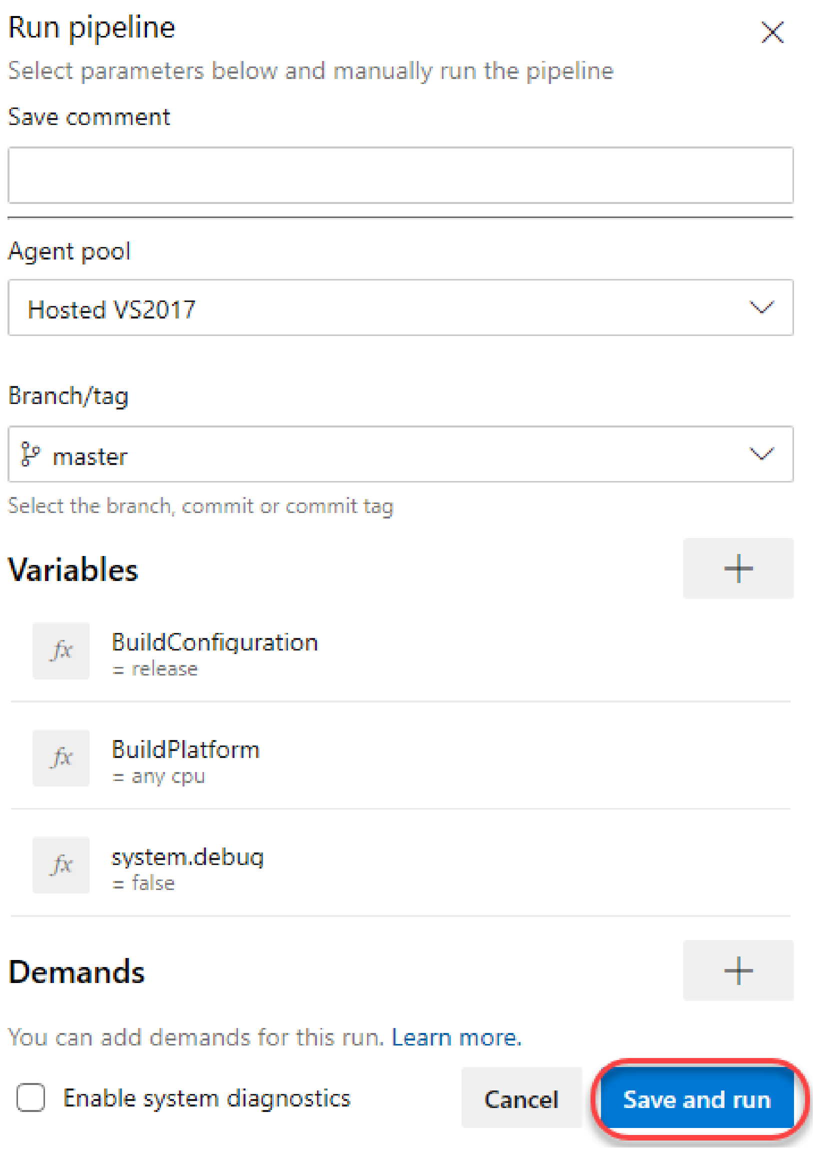
15. Select the **History** tab. There’s nothing here yet, but it will show a history of changes you make to the build definition.



16. Select **Save & Queue | Save & Queue** to save and queue a new build.



17. Accept the default options by clicking **Save and run**.



Run pipeline

Select parameters below and manually run the pipeline

Save comment

Agent pool

Hosted VS2017

Branch/tag

master

Select the branch, commit or commit tag

Variables

BuildConfiguration
= release

BuildPlatform
= any cpu

system.debug
= false

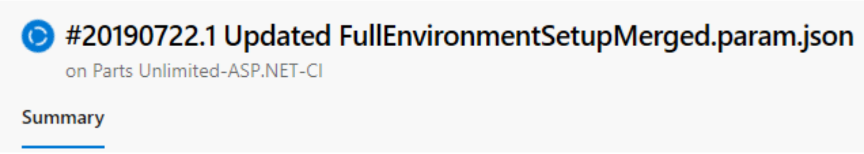
Demands

You can add demands for this run. Learn more.

□ Enable system diagnostics

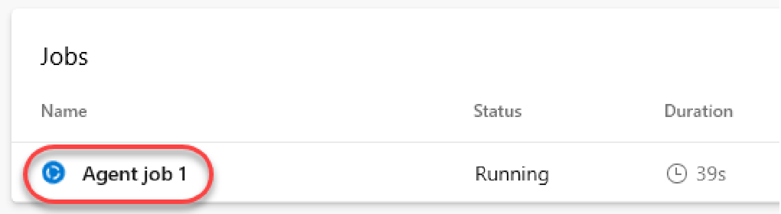
Task 2: Tracking and reviewing a build

1. Depending on load, the build may need to wait in the queue for a moment.



#20190722.1 Updated FullEnvironmentSetupMerged.param.json
on Parts Unlimited-ASP.NET-CI

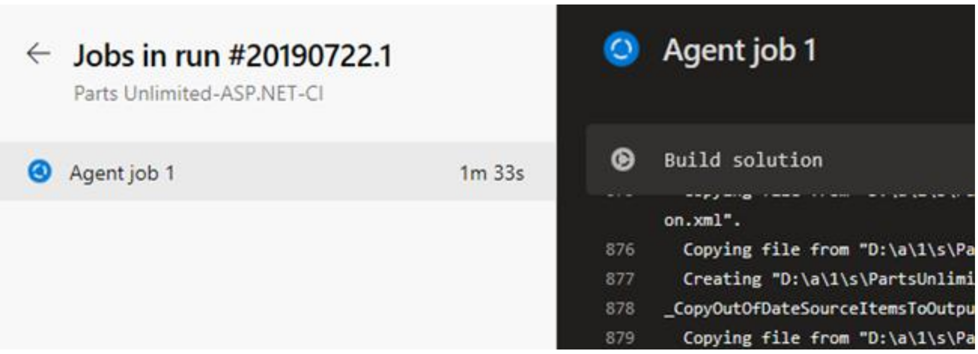
2. Once the build begins, you'll be able to track the console output per task. Click **Agent job 1**.



Jobs

Name	Status	Duration
Agent job 1	Running	39s

3. If you want to review an earlier task, you can scroll the right pane to review its logs.



Jobs in run #20190722.1

Parts Unlimited-ASP.NET-CI

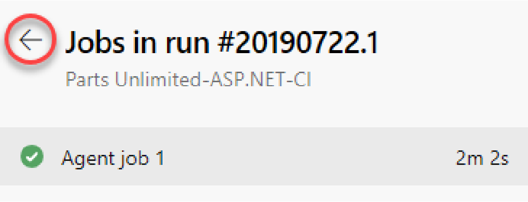
Agent job 1 1m 33s

Agent job 1

Build solution

```
on.xml".
  Copying file from "D:\a\1\s\Pa
  Creating "D:\a\1\s\PartsUnlimi
_CopyOutOfDateSourceItemsToOutpu
  Copying file from "D:\a\1\s\Pa
```

4. The build should eventually succeed. You can return to the summary view by clicking the back button.

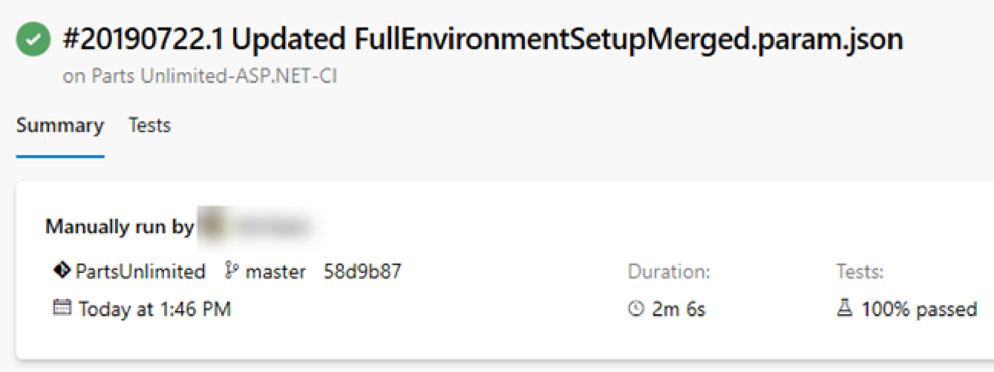


Jobs in run #20190722.1

Parts Unlimited-ASP.NET-CI

Agent job 1 2m 2s

5. The summary view provides overview details about the build, including details about commits, tests, and artifacts.



#20190722.1 Updated FullEnvironmentSetupMerged.param.json

on Parts Unlimited-ASP.NET-CI

Summary Tests

Manually run by [redacted]

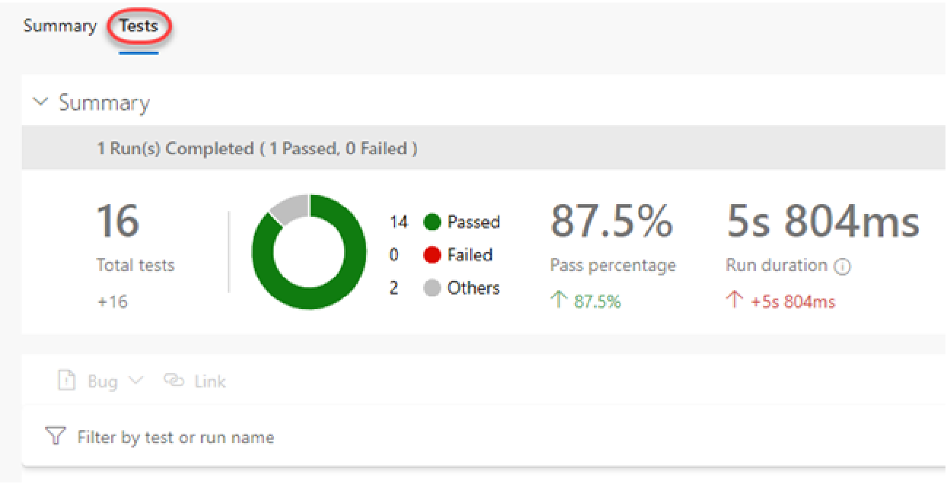
PartsUnlimited master 58d9b87

Today at 1:46 PM

Duration: 2m 6s

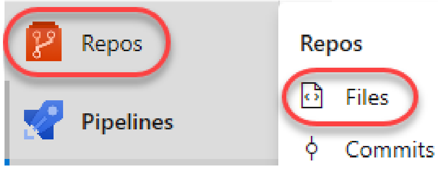
Tests: 100% passed

6. Select the **Tests** tab to review test performance for this build. Note that you also have easy access to the pipeline editor, the ability to queue a new build, and download the artifacts of this build.

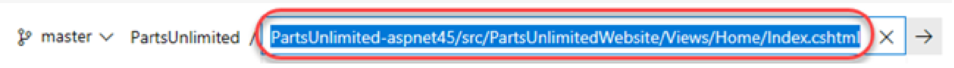


Task 3: Invoking a continuous integration build

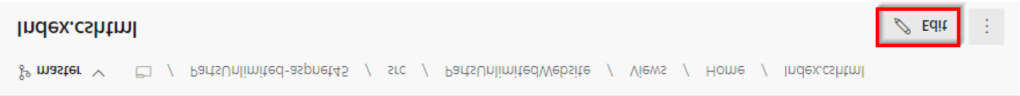
1. The build was configured earlier to support continuous integration. Navigate to the code for this project using **Repos | Files**.



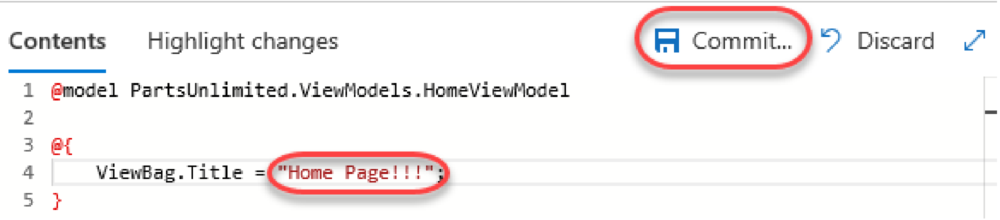
2. Open the file at **PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml**.


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

3. Click **Edit**.


Index.cshtml
Edit
master
PartsUnlimited-aspnet45 \ src \ PartsUnlimitedWebsite \ Views \ Home \ Index.cshtml

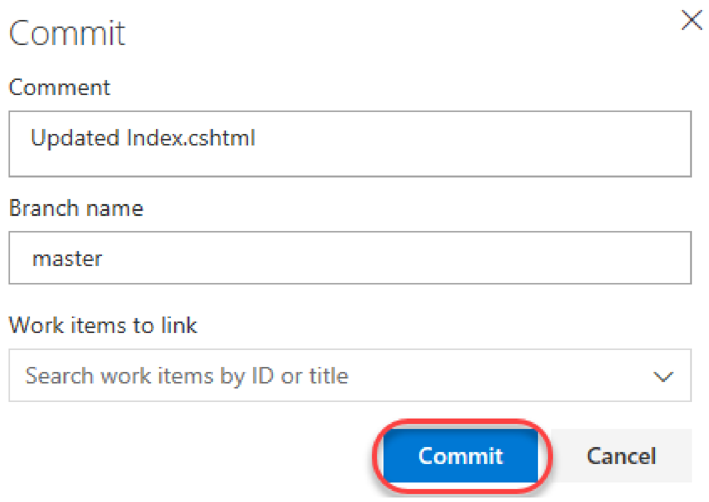
4. Make a minor cosmetic change, such as by tweaking the title of the document. Click **Commit**.



```
@model PartsUnlimited.ViewModels.HomeViewModel

@{
    ViewBag.Title = "Home Page!!!";
}
```

5. Accept the default commit details and click **Commit**.



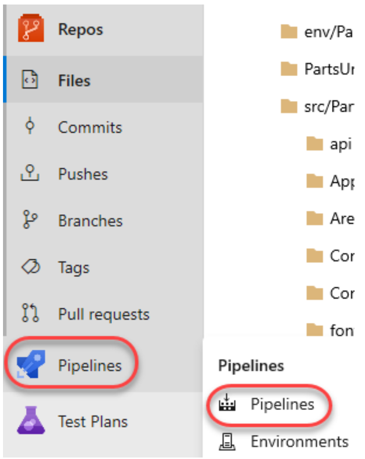
Commit

Comment: Updated Index.cshtml

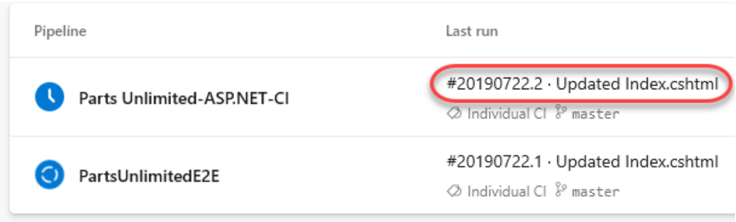
Branch name: master

Work items to link: Search work items by ID or title

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

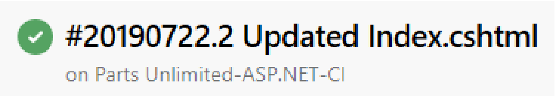


7. You should now see that a new build (note the **.2**) is in progress and that it was triggered by your change. Click the build to track it. Note that it may be queued behind another build pipeline configured for continuous integration.



Pipeline	Last run
Parts Unlimited-ASP.NET-CI	#20190722.2 · Updated Index.cshtml Individual CI master
PartsUnlimitedE2E	#20190722.1 · Updated Index.cshtml Individual CI master

8. This build should run and succeed just like the previous build.



✓ #20190722.2 Updated Index.cshtml

on Parts Unlimited-ASP.NET-CI