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.

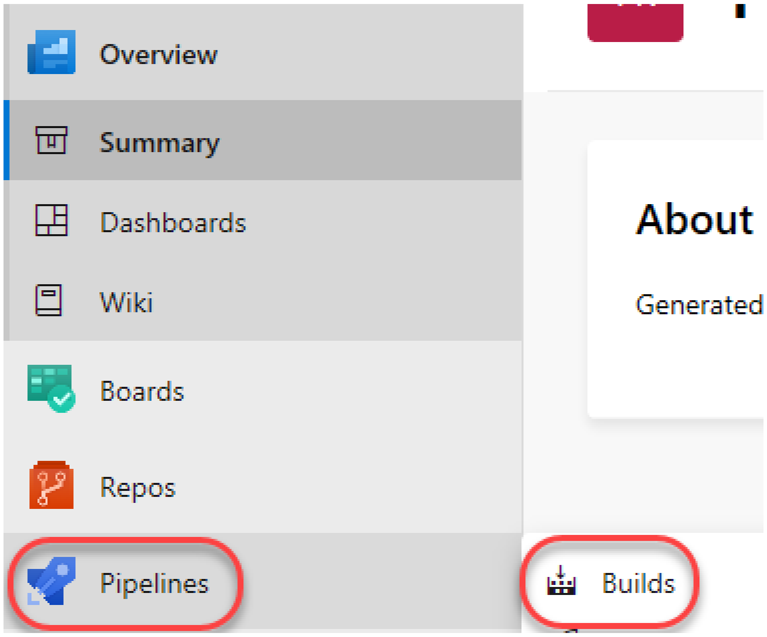
Prerequisites

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

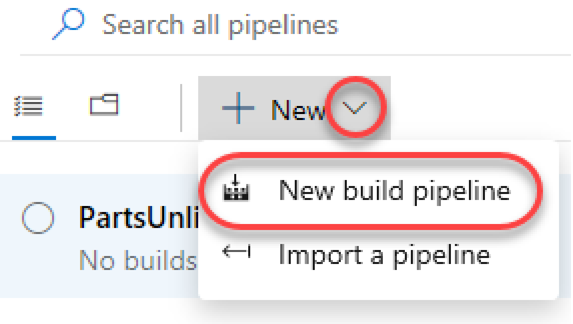
Exercise 1: Introduction to Azure DevOps Build

Task 1: Creating a basic build pipeline from a template

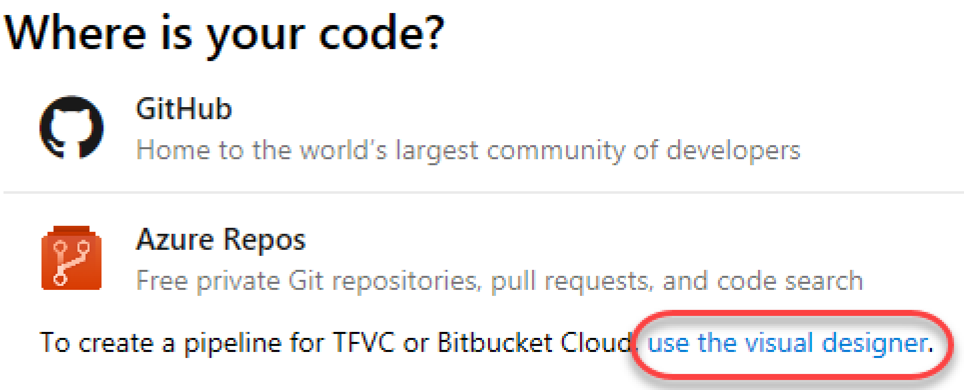
1. Navigate to your team project on Azure DevOps.
2. Navigate to **Pipelines | Builds**.



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



4. 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 visual designer**.



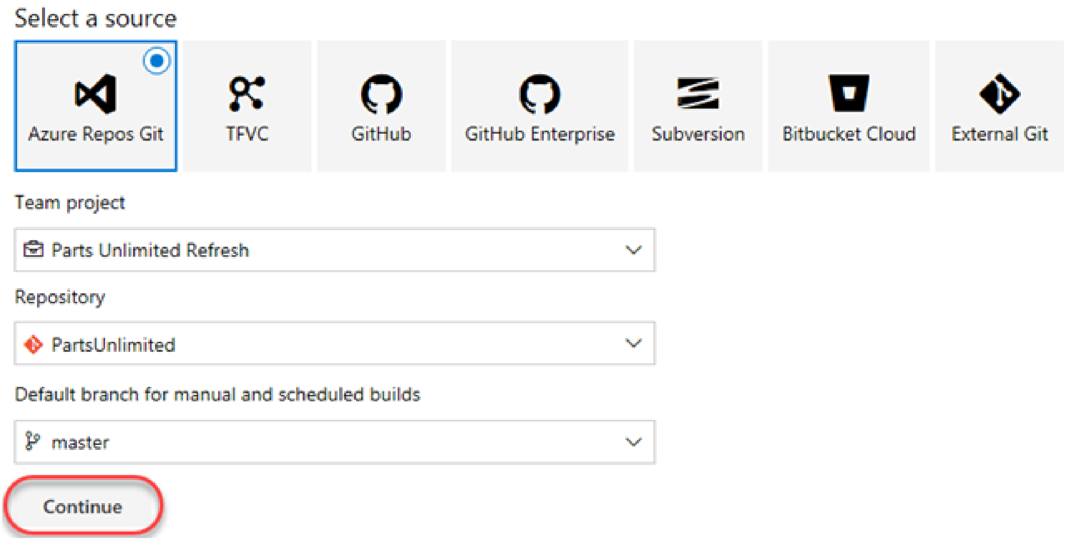
Where is your code?

GitHub
Home to the world's largest community of developers

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

To create a pipeline for TFVC or Bitbucket Cloud, use the visual designer.

5. 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**.

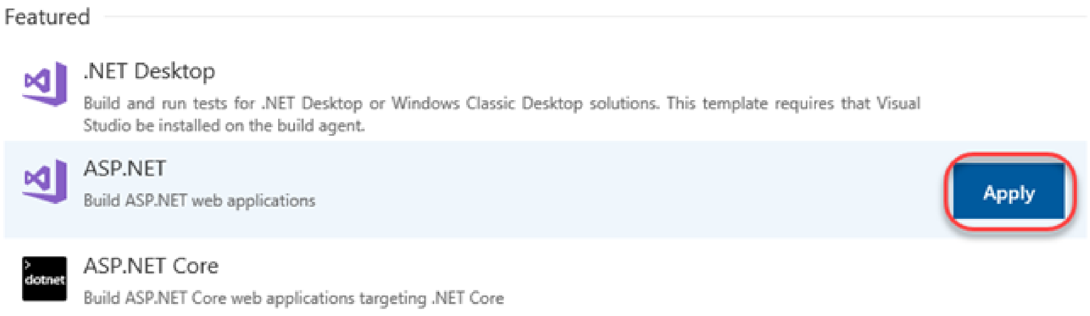


Team project: Parts Unlimited Refresh

Repository: PartsUnlimited

Default branch for manual and scheduled builds: master

6. 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.



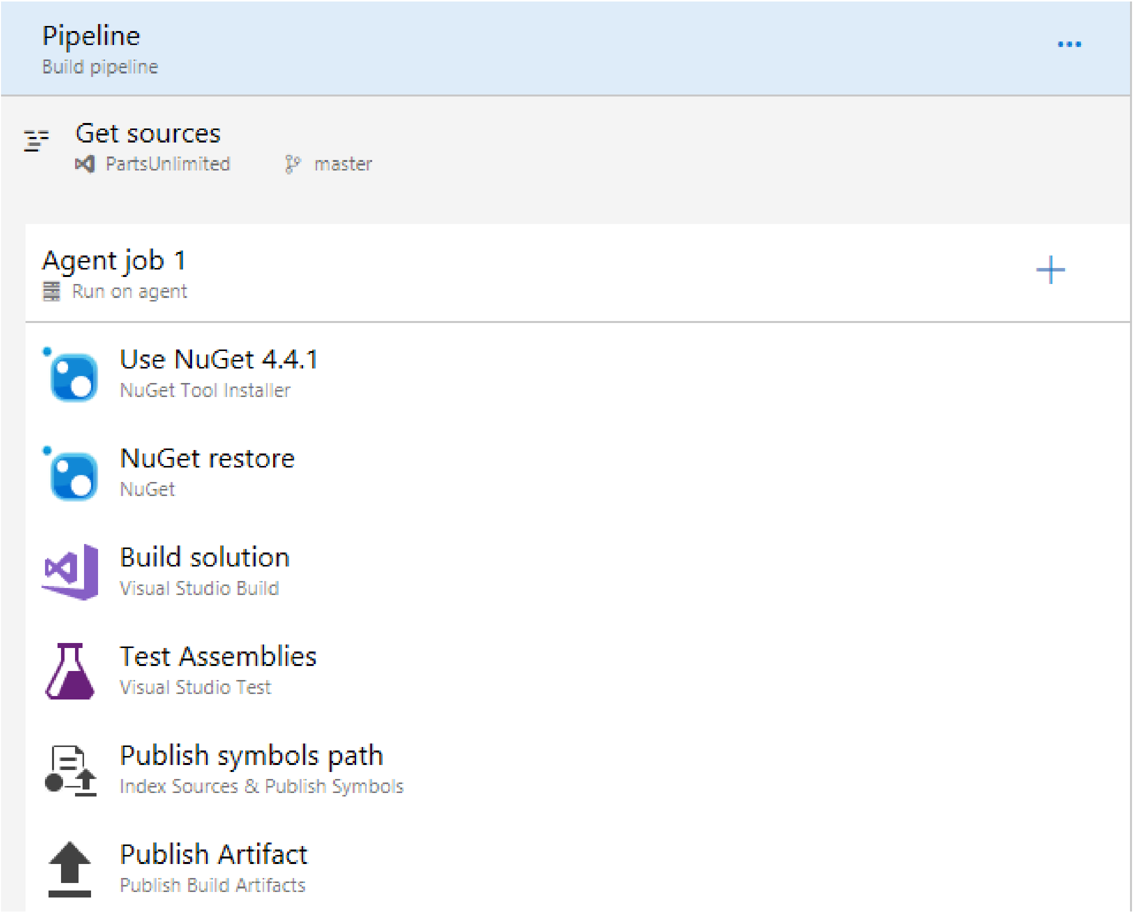
Featured

.NET Desktop
Build and run tests for .NET Desktop or Windows Classic Desktop solutions. This template requires that Visual Studio be installed on the build agent.

ASP.NET
Build ASP.NET web applications

ASP.NET Core
Build ASP.NET Core web applications targeting .NET Core

7. 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

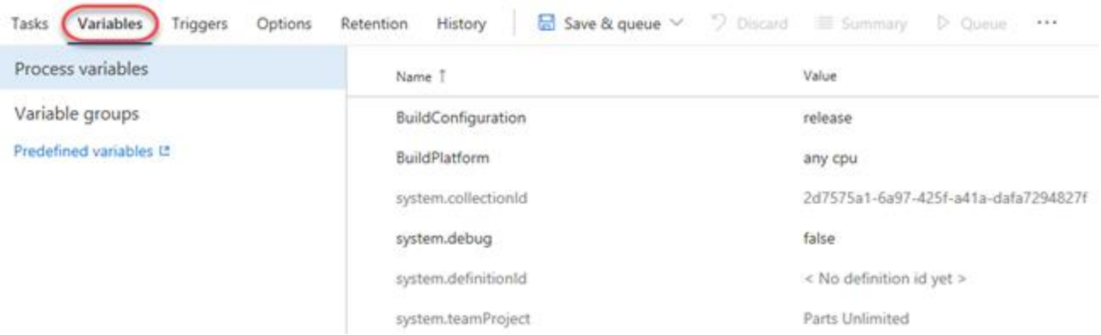
Build solution
Visual Studio Build

Test Assemblies
Visual Studio Test

Publish symbols path
Index Sources & Publish Symbols

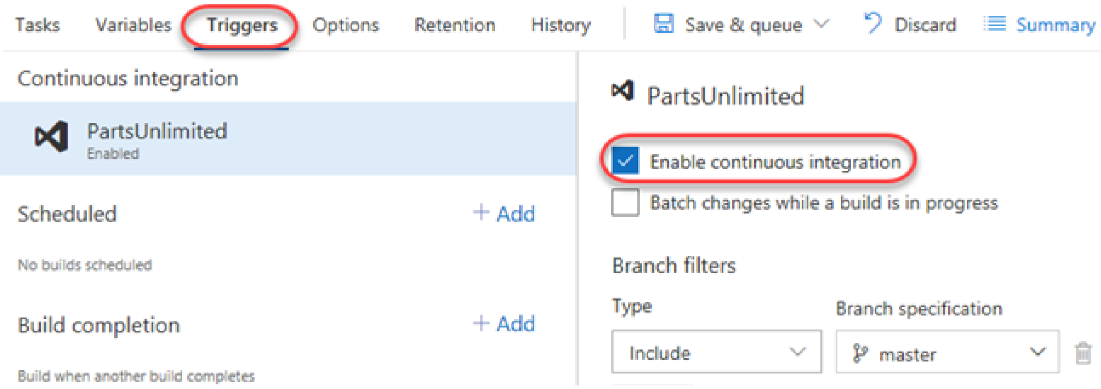
Publish Artifact
Publish Build Artifacts

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



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

9. 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.



PartsUnlimited
Enabled

No builds scheduled

Build when another build completes

PartsUnlimited

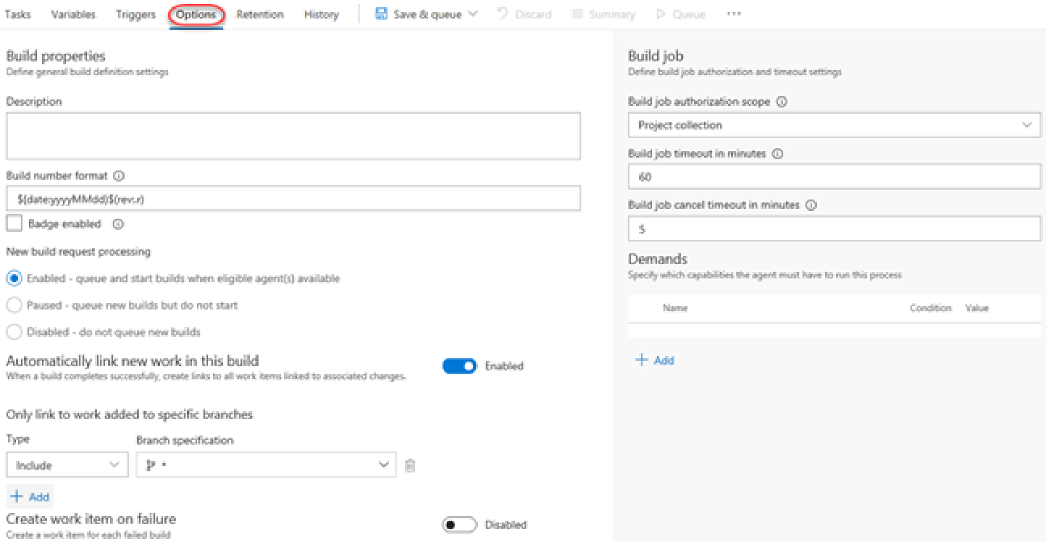
☑ Enable continuous integration

□ Batch changes while a build is in progress

Type: Include

Branch specification: master

10. 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: *

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

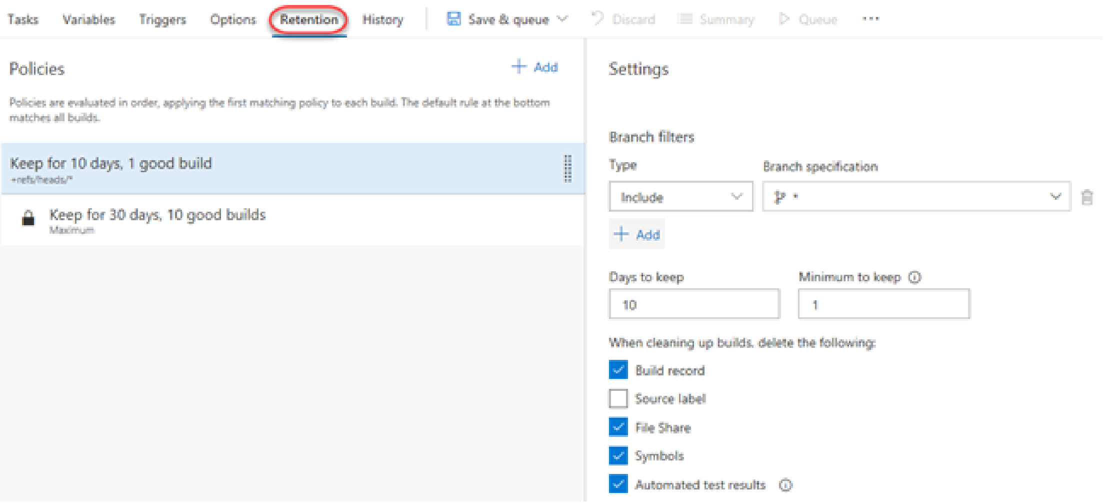
Build job timeout in minutes: 60

Build job cancel timeout in minutes: 5

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

Name	Condition	Value

11. Select the **Retention** tab. This is where you configure which builds are retained and for how long.



Policies

Policies are evaluated in order, applying the first matching policy to each build. The default rule at the bottom matches all builds.

Keep for 10 days, 1 good build
+refs/heads/*

Keep for 30 days, 10 good builds
Maximum

Settings

Branch filters

Type	Branch specification
Include	*

Days to keep: 10

Minimum to keep: 1

When cleaning up builds, delete the following:

☑ Build record

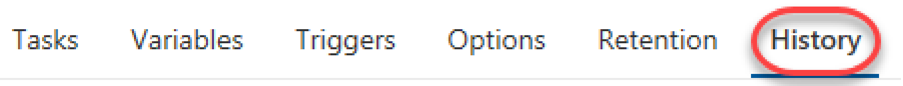
☐ Source label

☑ File Share

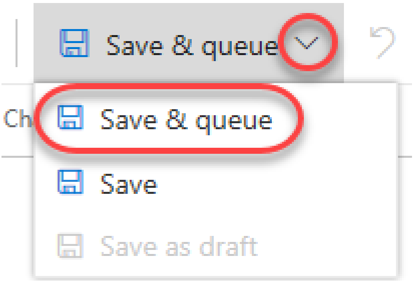
☑ Symbols

☑ Automated test results

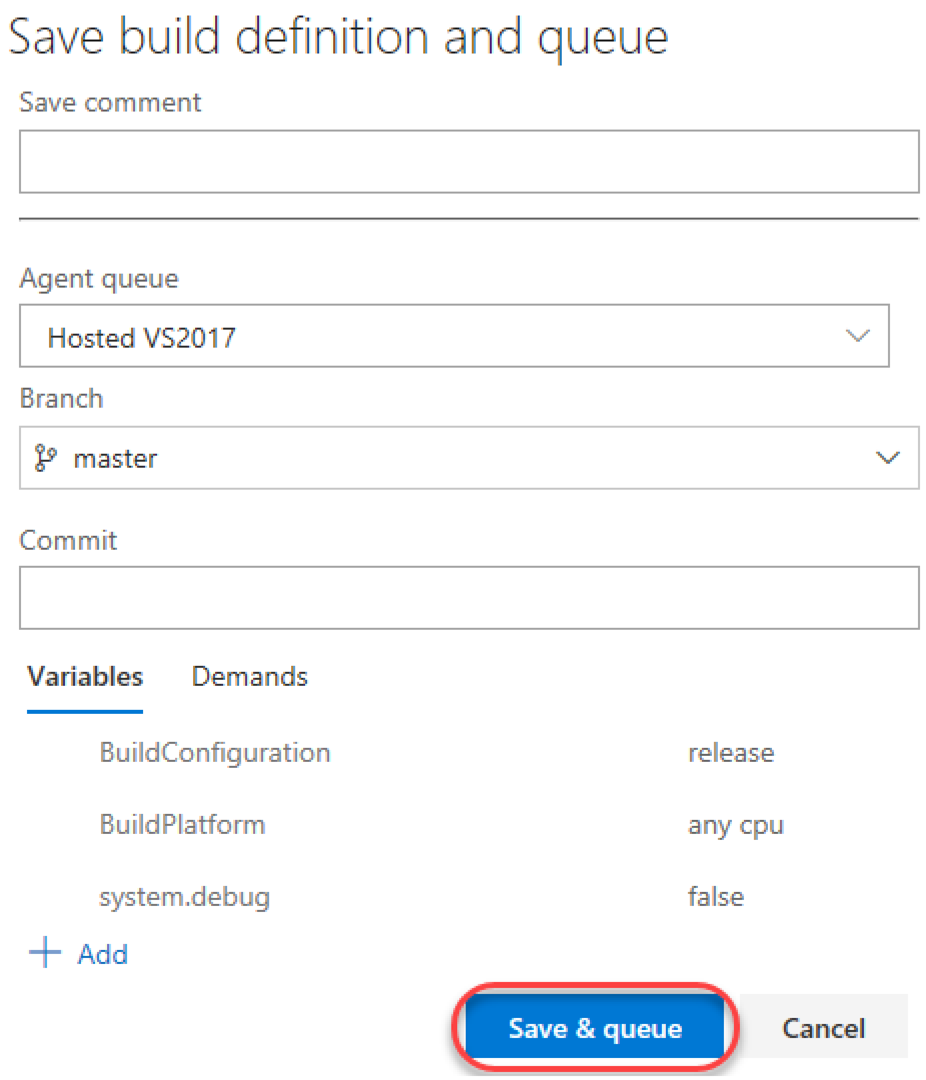
12. Select the **History** tab. There's nothing here yet, but it will show a history of changes you make to the build definition.



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



14. Accept the default options by clicking **Save & queue**.



Save build definition and queue

Save comment

Agent queue

Hosted VS2017

Branch

master

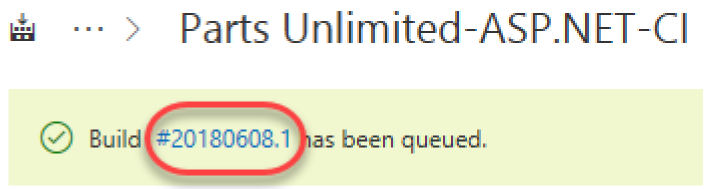
Commit

Variables | Demands

BuildConfiguration	release
BuildPlatform	any cpu
system.debug	false

Task 2: Tracking and reviewing a build

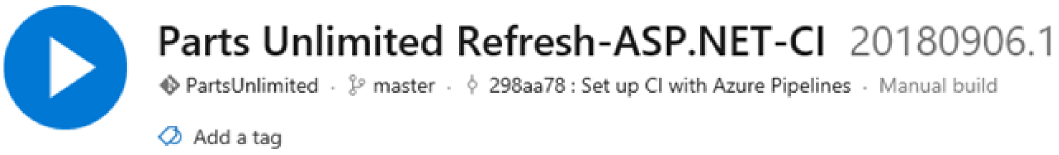
1. Click the link to the newly created build.



Parts Unlimited-ASP.NET-CI

Build #20180608.1 has been queued.

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

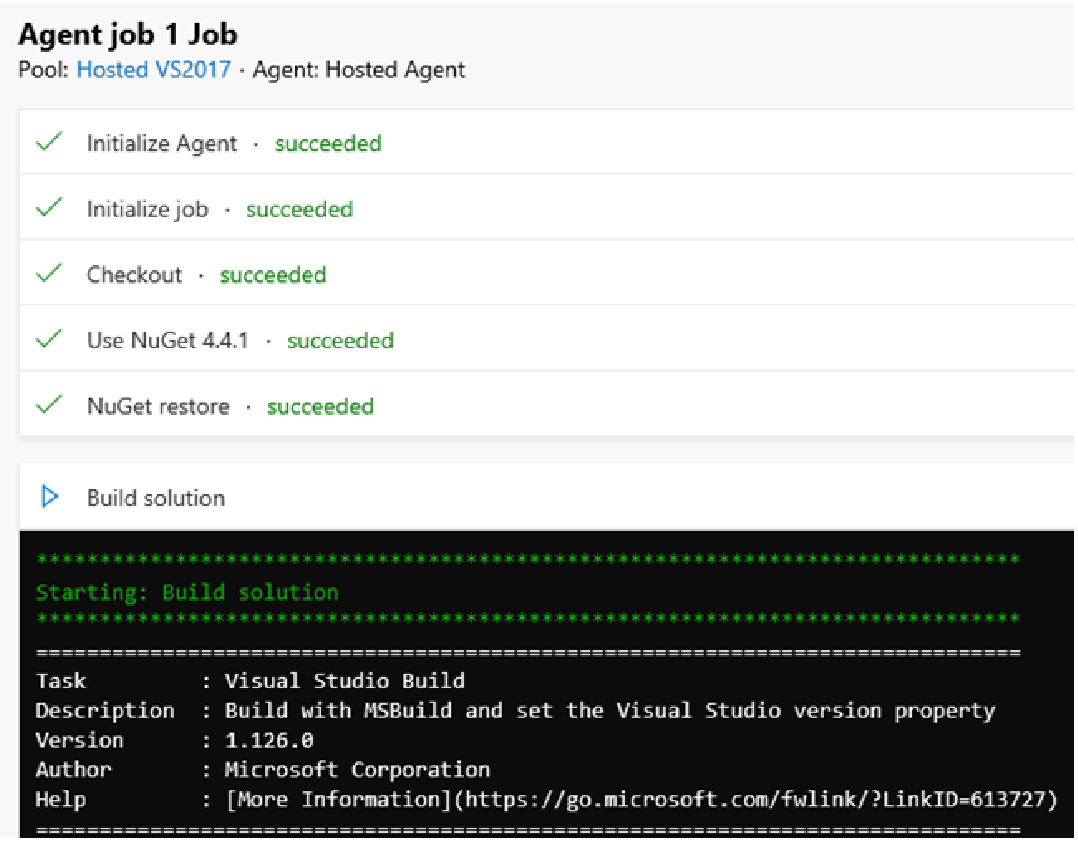


Parts Unlimited Refresh-ASP.NET-CI 20180906.1

PartsUnlimited · master · 298aa78 : Set up CI with Azure Pipelines · Manual build

Add a tag

3. Once the build begins, you’ll be able to track the console output. If you want to review an earlier task, you can click to review its logs.



Agent job 1 Job

Pool: Hosted VS2017 · Agent: Hosted Agent

✓ Initialize Agent · succeeded

✓ Initialize job · succeeded

✓ Checkout · succeeded

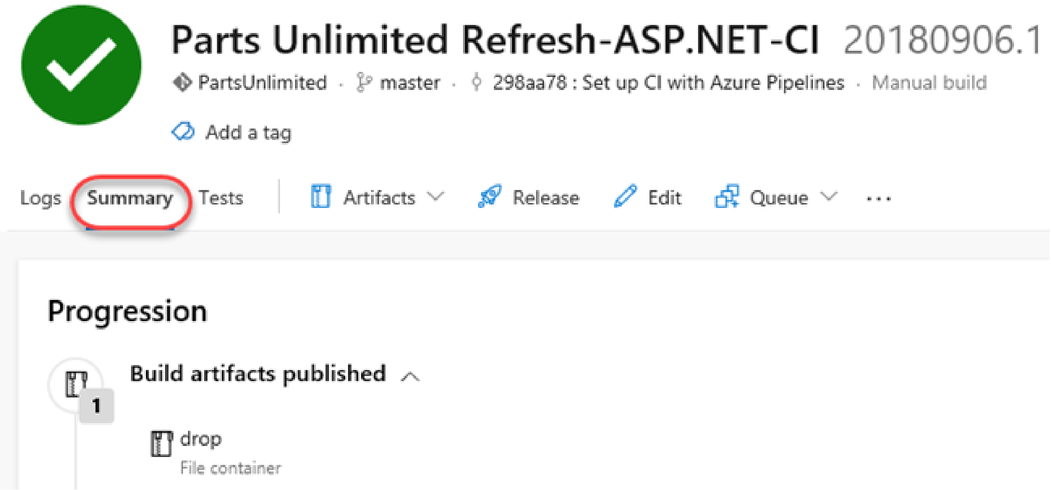
✓ Use NuGet 4.4.1 · succeeded

✓ NuGet restore · succeeded

▷ Build solution

```
******************************************************************************
Starting: Build solution
******************************************************************************
==============================================================================
Task         : Visual Studio Build
Description  : Build with MSBuild and set the Visual Studio version property
Version      : 1.126.0
Author       : Microsoft Corporation
Help         : [More Information](https://go.microsoft.com/fwlink/?LinkID=613727)
==============================================================================
```

4. The build should eventually succeed, which you can review in the **Summary** tab.



Parts Unlimited Refresh-ASP.NET-CI 20180906.1

PartsUnlimited · master · 298aa78 : Set up CI with Azure Pipelines · Manual build

Add a tag

Progression

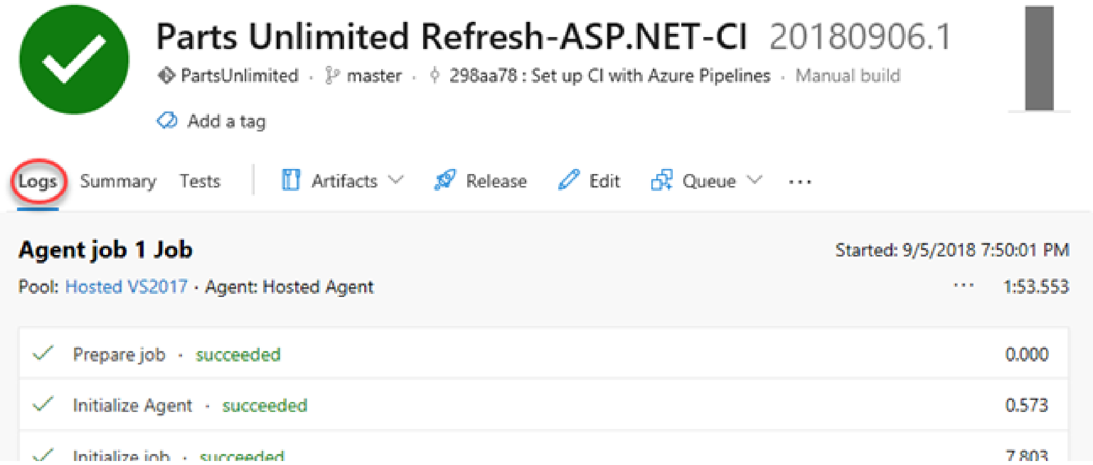
Build artifacts published

1

drop

File container

5. Select the **Logs** tab to see each step and how long it took.



Parts Unlimited Refresh-ASP.NET-CI 20180906.1

PartsUnlimited · master · 298aa78 : Set up CI with Azure Pipelines · Manual build

Add a tag

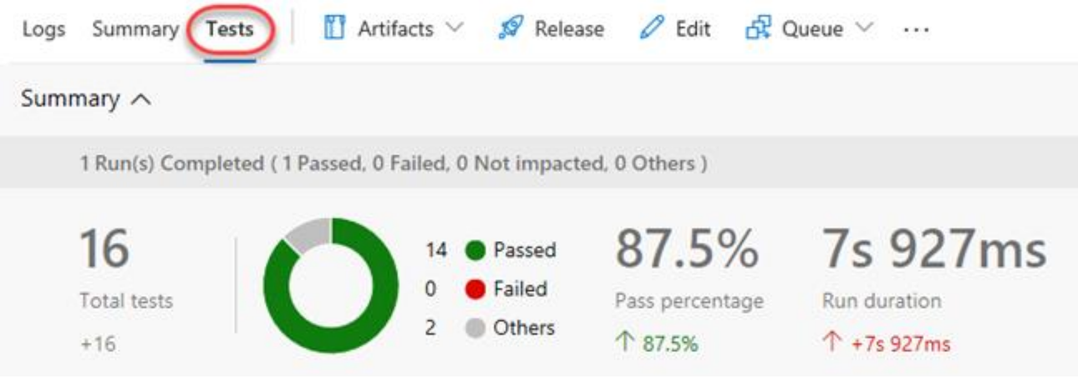
Agent job 1 Job

Started: 9/5/2018 7:50:01 PM

Pool: Hosted VS2017 · Agent: Hosted Agent ··· 1:53.553

```
✓ Prepare job · succeeded          0.000
✓ Initialize Agent · succeeded     0.573
✓ Initialize job · succeeded       7.803
```

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.



Summary

1 Run(s) Completed (1 Passed, 0 Failed, 0 Not impacted, 0 Others)

16 Total tests +16

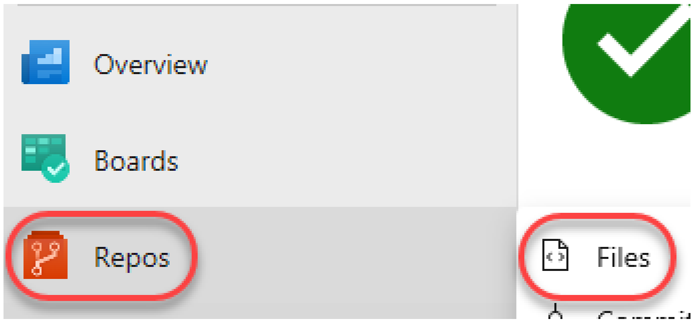
14 ● Passed
0 ● Failed
2 ● Others

87.5% Pass percentage ↑ 87.5%

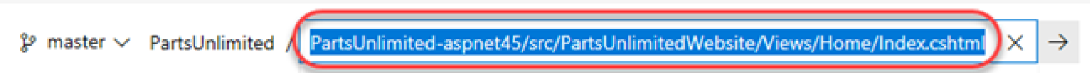
7s 927ms Run duration ↑ +7s 927ms

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**.

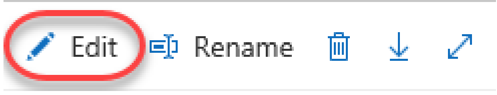


2. Open the file at **PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml**.

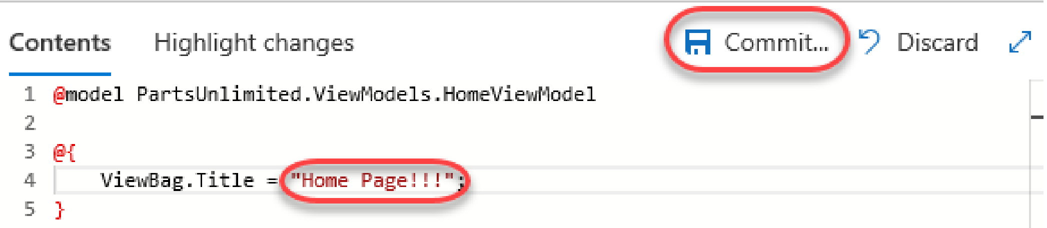


PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml

3. Click **Edit**.



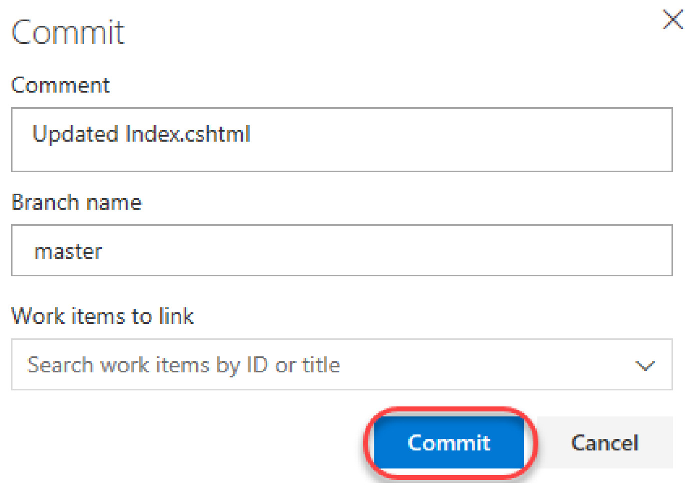
4. Make a minor cosmetic change, such as by tweaking the title of the document. Click **Commit**.



```
@model PartsUnlimited.ViewModels.HomeViewModel

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

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



Commit

Comment

Updated Index.cshtml

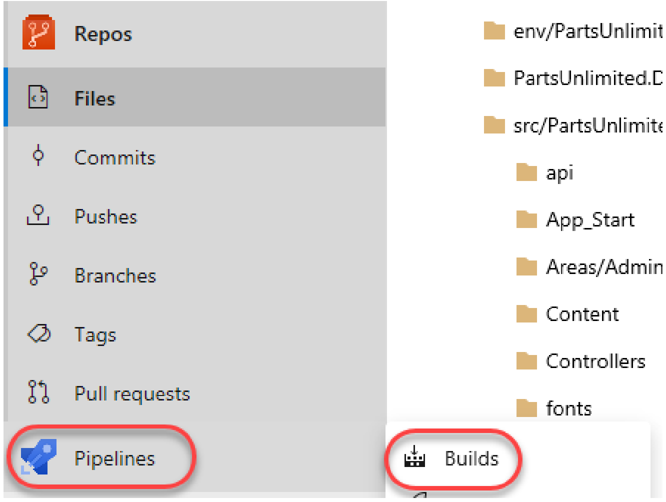
Branch name

master

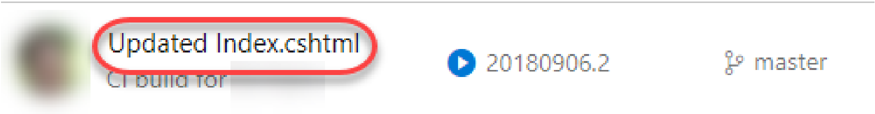
Work items to link

Search work items by ID or title

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



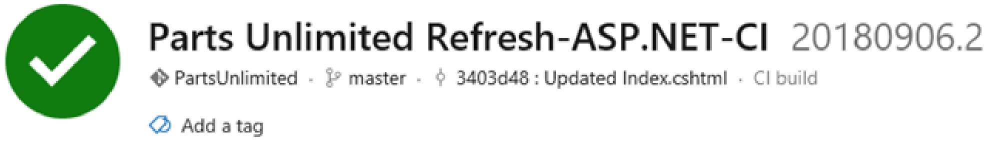
7. You should now see that a new build (note the **.2**) is in progress and that it was triggered by your change. Click the build to track it.



Updated Index.cshtml | 20180906.2 | master

CI build for

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



Parts Unlimited Refresh-ASP.NET-CI 20180906.2

PartsUnlimited · master · 3403d48 : Updated Index.cshtml · CI build