

# Azure DevOps Lab

## Overview

Certain Azure DevOps labs require a preconfigured **Parts Unlimited** team project. This document outlines the required steps to set up the required data.

### *Task 1: Configuring the Parts Unlimited team project*

1. Navigate to <https://azuredevopsdemogenerator.azurewebsites.net>. This utility site will automate the process of creating a new Azure DevOps project within your account that is prepopulated with content (work items, repos, etc.) required for the lab. For more information on the site, please see <https://docs.microsoft.com/en-us/azure/devops/demo-gen>.
2. Sign in using the Microsoft account associated with your Azure DevOps subscription.



#### AZURE DEVOPS DEMO GENERATOR

Azure DevOps Demo Generator helps you create projects on your Azure DevOps Organization with pre-populated sample content that includes source code, work items, iterations, service endpoints, build and release definitions based on a template you choose. [Read more](#)

The purpose of this system is to simplify working with the [Azure DevOps hands-on-labs](#), demos and other education material provided by the Microsoft Visual Studio Marketing team.



Don't have an Azure DevOps Organization?

[Get started for free >](#)


3. **Accept** the permission requests for accessing your subscription.
4. Select your Azure DevOps organization and enter the project name **"Parts Unlimited"**. Click **Choose Template**.


Select Organization :


New Project Name :


Selected Template :

5. Select the **PartsUnlimited** template and click **Select Template**.

**Tailwind Traders**  
    
Tailwind Traders is an ASP.NET & React application, which uses Azure App Service, AKS, Cosmos DB, Logic App and the Function App

**SmartHotel360**  
    
This template contains work items, code and pipeline definitions for the public web site of SmartHotel360, an E2E reference sample app with several consumer and line-of-business apps and an Azure backend. For more information, please see the project page on [GitHub](#)

**PartsUnlimited**  
    
Use this lab to provision a scrum based team project containing sample work items, complete source code and pipeline definitions to deploy Parts Unlimited a scrum

**PartsUnlimited-YAML**  
    
Use this lab to provision a scrum based team project containing sample work items, complete source code and pipeline definitions to deploy Parts Unlimited a scrum

6. Click **Create Project** and wait for the process to complete.

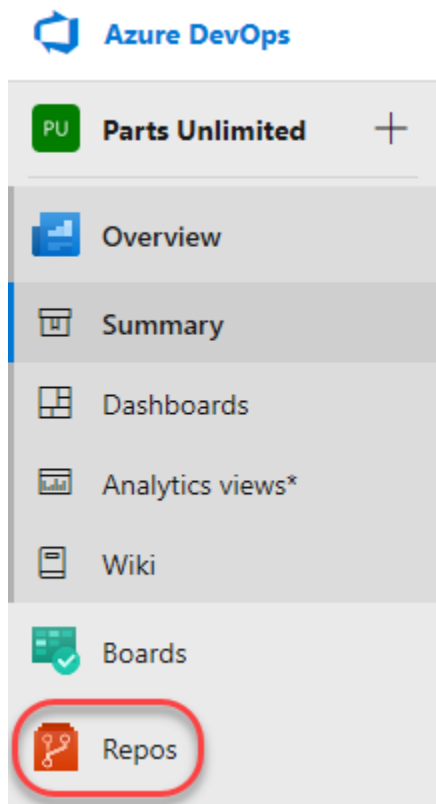
Select Organization :

New Project Name :

Selected Template :

### ***Task 2: Configuring the Parts Unlimited solution in Visual Studio***

1. Some labs will require you to open the **Parts Unlimited** solution in **Visual Studio**. If your lab doesn't require this, you can skip this task.
2. Navigate to your Azure DevOps team project for **Parts Unlimited**. It will be something like <https://dev.azure.com/YOURACCOUNT/Parts%20Unlimited>.
3. Navigate to the **Repos** hub.



- Click **Clone** and select **Clone in Visual Studio** (choose it in the dropdown if other option shown as default).

The screenshot shows the Visual Studio interface with the 'PartsUnlimited' repository open. The 'Files' tab is active, showing a table of files and their commit history. The 'Clone' button in the top right corner is highlighted with a red box. Below this, the 'Clone repository' dialog is open, showing the 'Clone in Visual Studio' option highlighted with a red oval.

**Files**

Name ↑	Last change	Commits
PartsUnlimited-aspnet45	Apr 29	7cad2fde Updated README.md Sriramdas Balaji
.gitattributes	Nov 18, 2015	45f7fff9 Add PartsUnlimited source Colin Dembovsky
.gitignore	Nov 18, 2015	45f7fff9 Add PartsUnlimited source Colin Dembovsky

**Clone repository**

Clone Git repository using command line or IDE

Command line

**HTTPS** SSH

https://[redacted]@dev.azure.com/[redacted]/Parts...

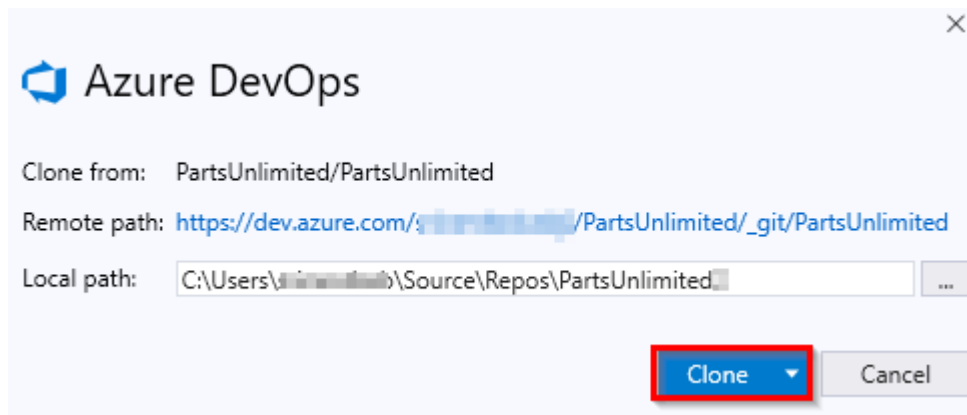
**Generate Git credentials**

IDE

**Clone in Visual Studio**

Having problems authenticating in Git? Be sure to get the latest version of [Git for Windows](#) or our plugins for [IntelliJ](#), [Eclipse](#), [Android Studio](#) or [Windows command line](#).

- Follow the workflow to clone and configure the project in Visual Studio. Click **Clone** to copy the repo locally.

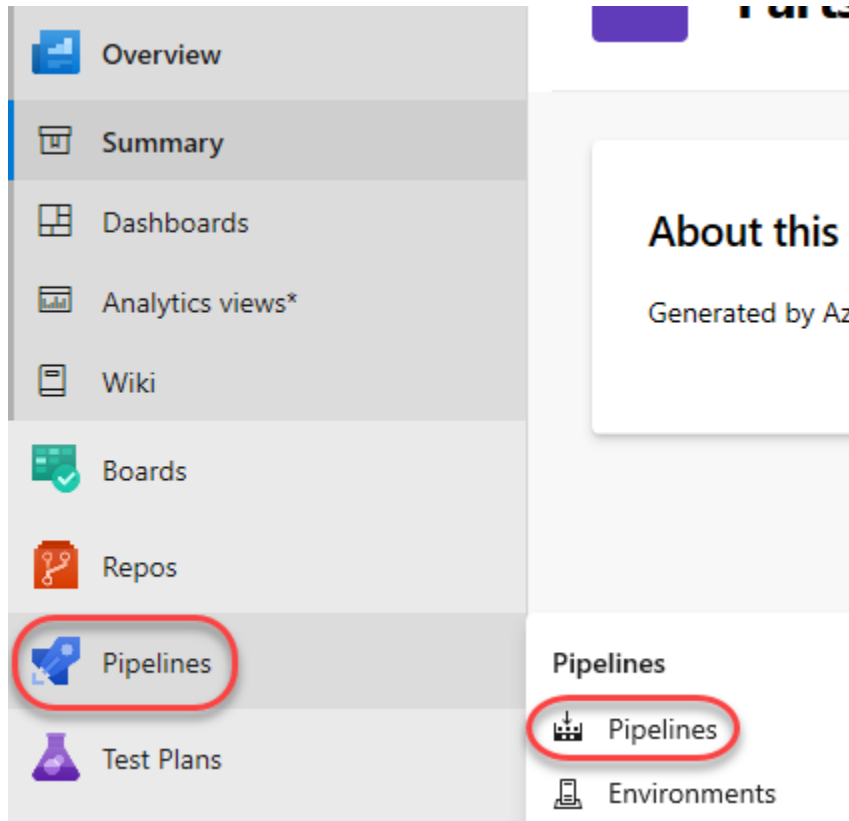


6. From **Team Explorer**, double-click **PartsUnlimited.sln** from the **Solutions** section to open the solution. You can ignore if you see any warnings about unsupported project types (just click OK on the prompted window and ignore the migration report opened on the browser)

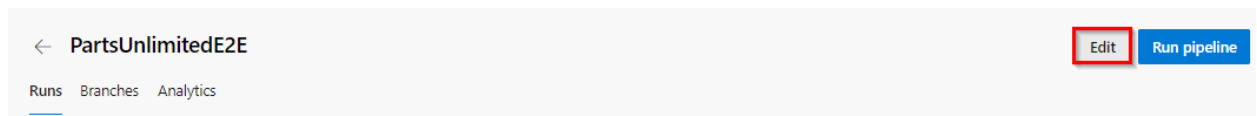
# Introduction to Azure DevOps Build

## *Task 1: Creating a basic build pipeline from a template*

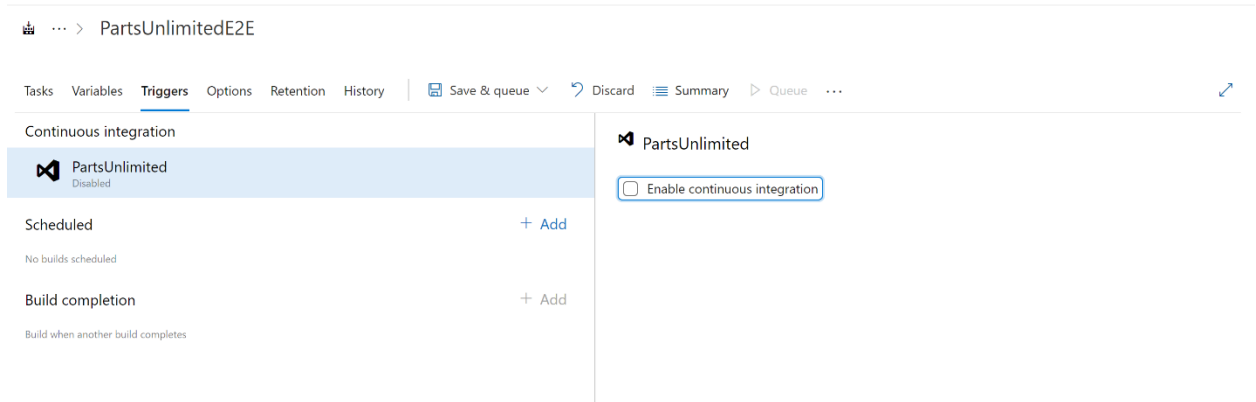
1. Navigate to your team project on Azure DevOps.
2. Navigate to **Pipelines | Pipelines**.



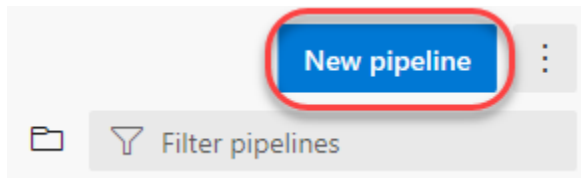
3. Open (click) the existing **PartsUnlimitedE2E** pipeline (which was created by the demo generator tool) and click on **Edit**



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



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**.





**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.

Apply

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.

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

**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.

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

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 Branch specification

Include

master

- 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.

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

Build properties  
Define general build definition settings

Description

Build number format ①  
\$(date:yyyyMMdd)\$(rev:r)

☐ Badge enabled ①

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  
When a build completes successfully, create links to all work items linked to associated changes. ☒ Enabled

Only link to work added to specific branches

Type Branch specification

Include

\* \*

+ Add

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

Build job  
Define build job authorization and timeout settings

Build job authorization scope ①  
Project collection

Build job timeout in minutes ①  
60

Build job cancel timeout in minutes ①  
5

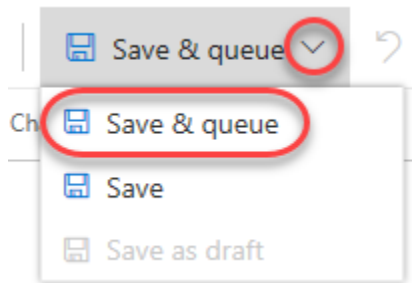
Demands  
Specify which capabilities the agent must have to run this process

Name	Condition	Value
+ Add		

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

Tasks Variables Triggers Options Retention **History**

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



15. 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.](#)

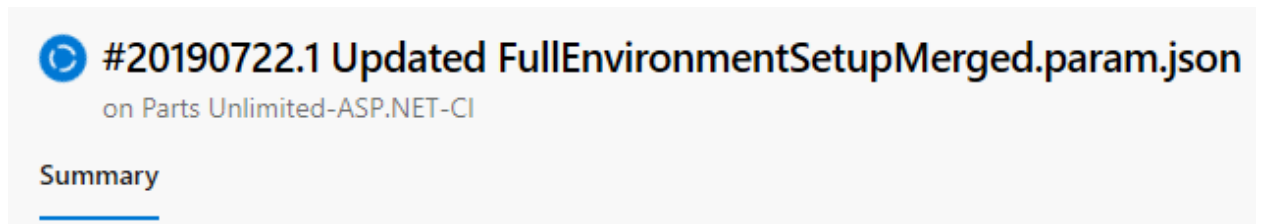
☐ Enable system diagnostics

Cancel

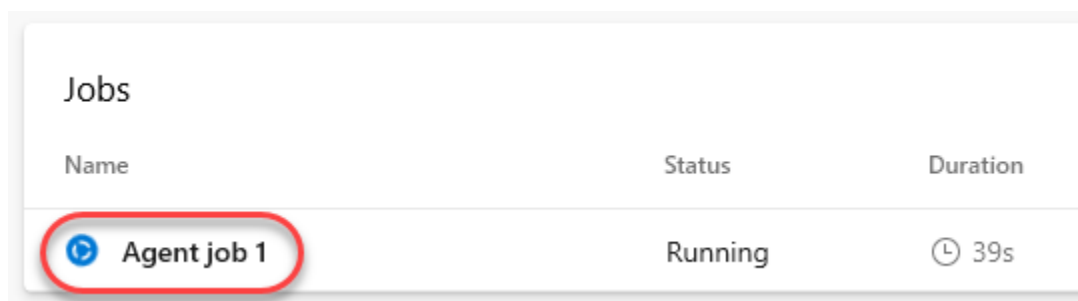
Save and run

## Task 2: Tracking and reviewing a build

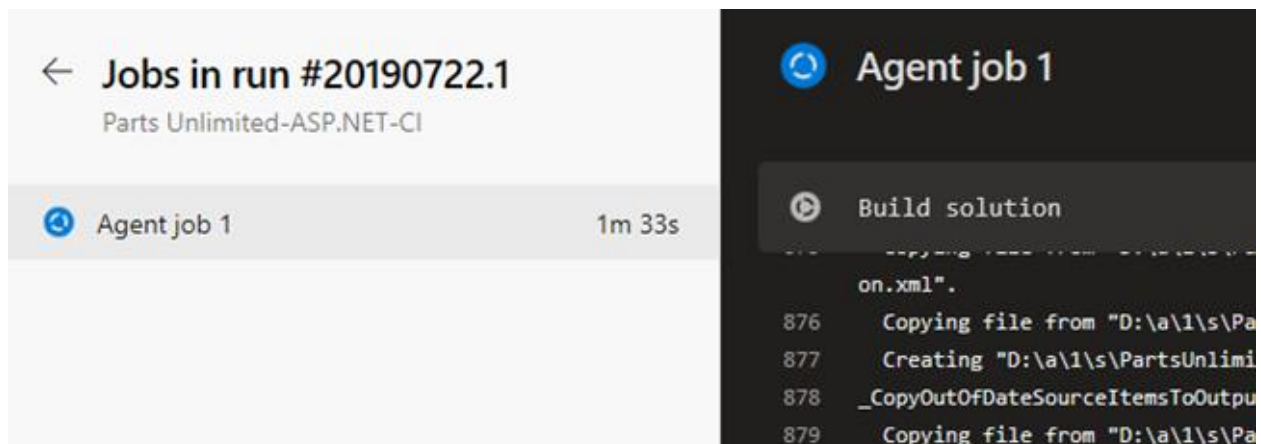
1. Depending on load, the build may need to wait in the queue for a moment.




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



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



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 

◆ 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.

Summary **Tests**

▽ Summary

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



16  
Total tests  
+16



14 ● Passed  
0 ● Failed  
2 ● Others

87.5%  
Pass percentage  
↑ 87.5%

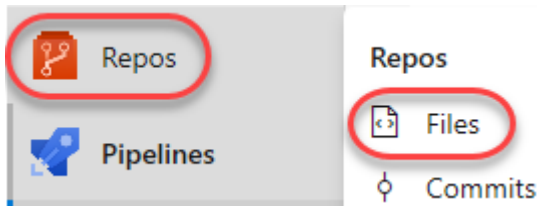
5s 804ms  
Run duration ⓘ  
↑ +5s 804ms

 Bug ▾  Link

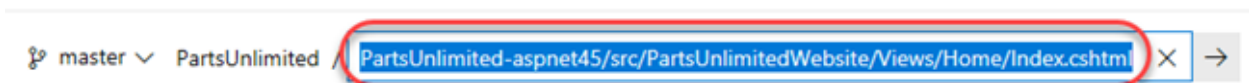
🔍 Filter by test or run name

### Task 3: Invoking a continuous integration build

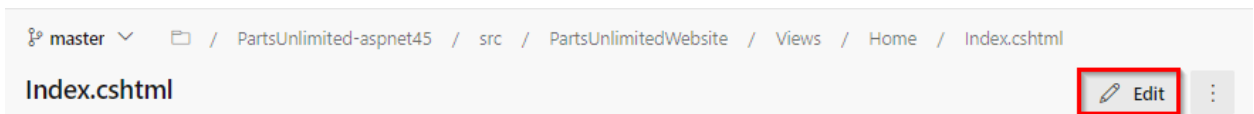
1. The build was configured earlier to support continuous integration. Navigate to the code for this project using **Repos | Files**.



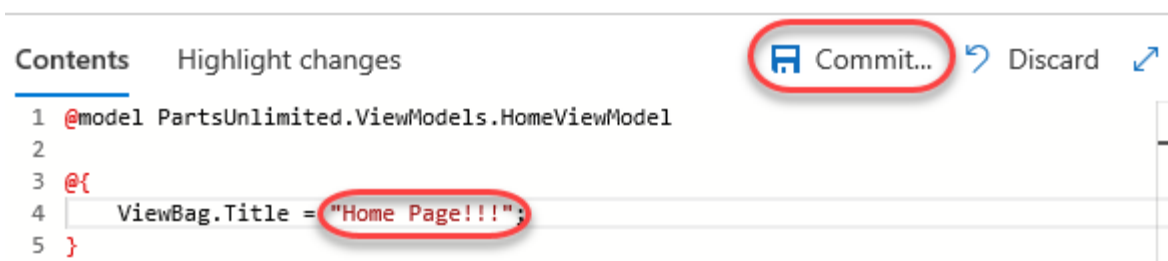
2. Open the file at **PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml**.



3. Click **Edit**.



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



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

Commit ×

Comment

Updated Index.cshtml

Branch name

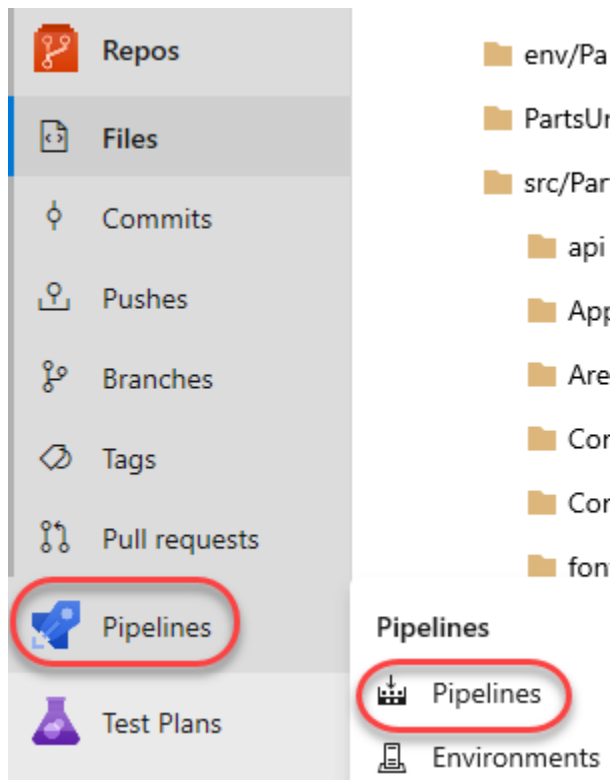
master

Work items to link







Search work items by ID or title ▼

Commit Cancel


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
 <b>Parts Unlimited-ASP.NET-CI</b>	<b>#20190722.2 · Updated Index.cshtml</b>  Individual CI  master
 <b>PartsUnlimitedE2E</b>	<b>#20190722.1 · Updated Index.cshtml</b>  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