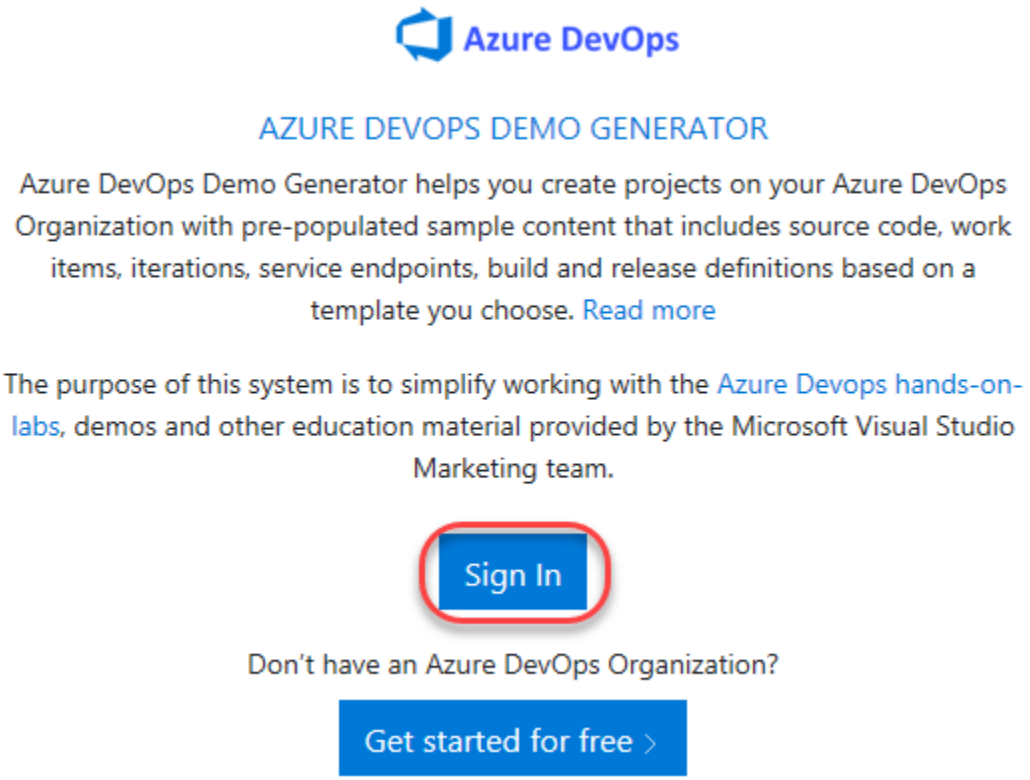
Azure DevOps Lab

Overview

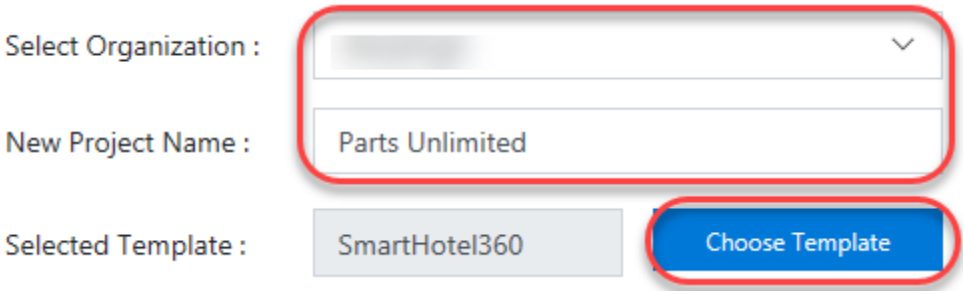
Certain Azure DevOps labs require a preconfigured **Parts Unlimited** team project. This document outlines the required steps to set up the required data.

Task 1: Configuring the Parts Unlimited team project

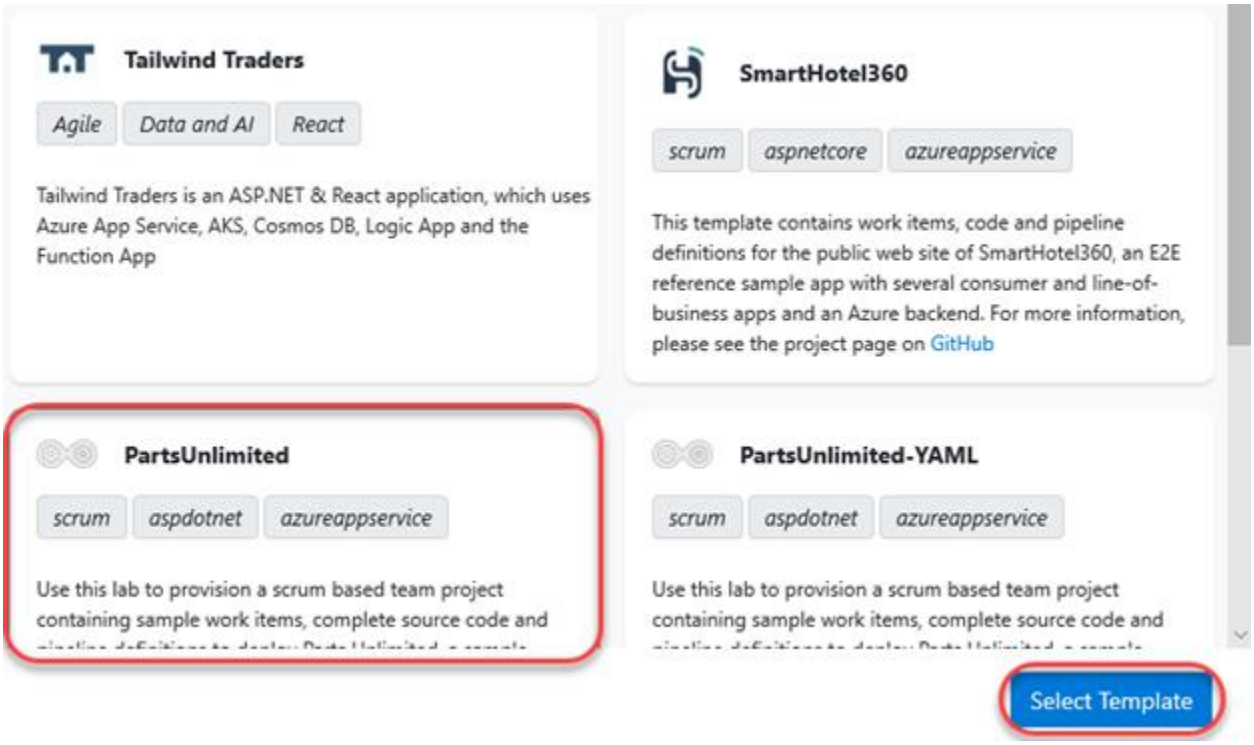
1. Navigate to https://azuredevopsdemogenerator.azurewebsites.net. This utility site will automate the process of creating a new Azure DevOps project within your account that is prepopulated with content (work items, repos, etc.) required for the lab. For more information on the site, please see https://docs.microsoft.com/en-us/azure/devops/demo-gen.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



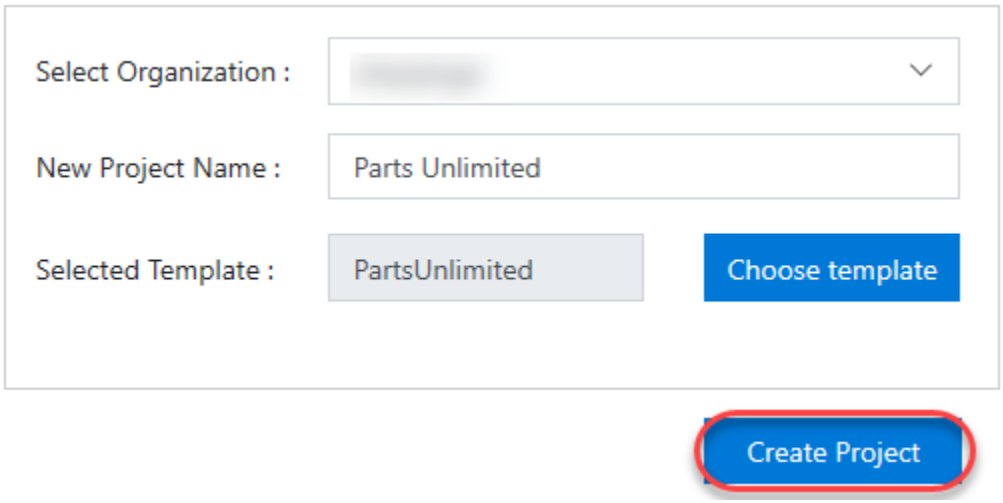
3. **Accept** the permission requests for accessing your subscription.
4. Select your Azure DevOps organization and enter the project name **"Parts Unlimited"**. Click **Choose Template**.



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

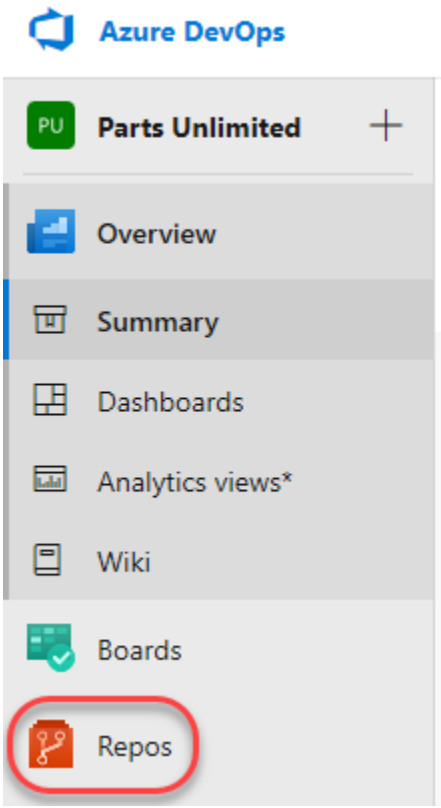


6. Click **Create Project** and wait for the process to complete.

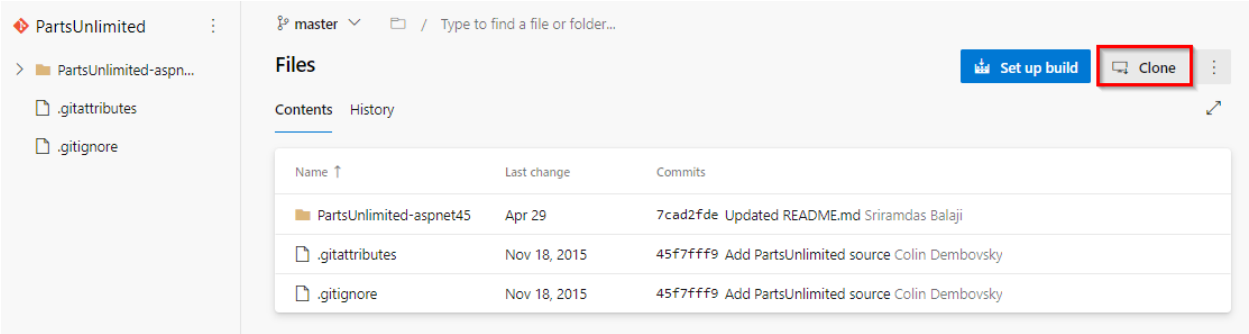


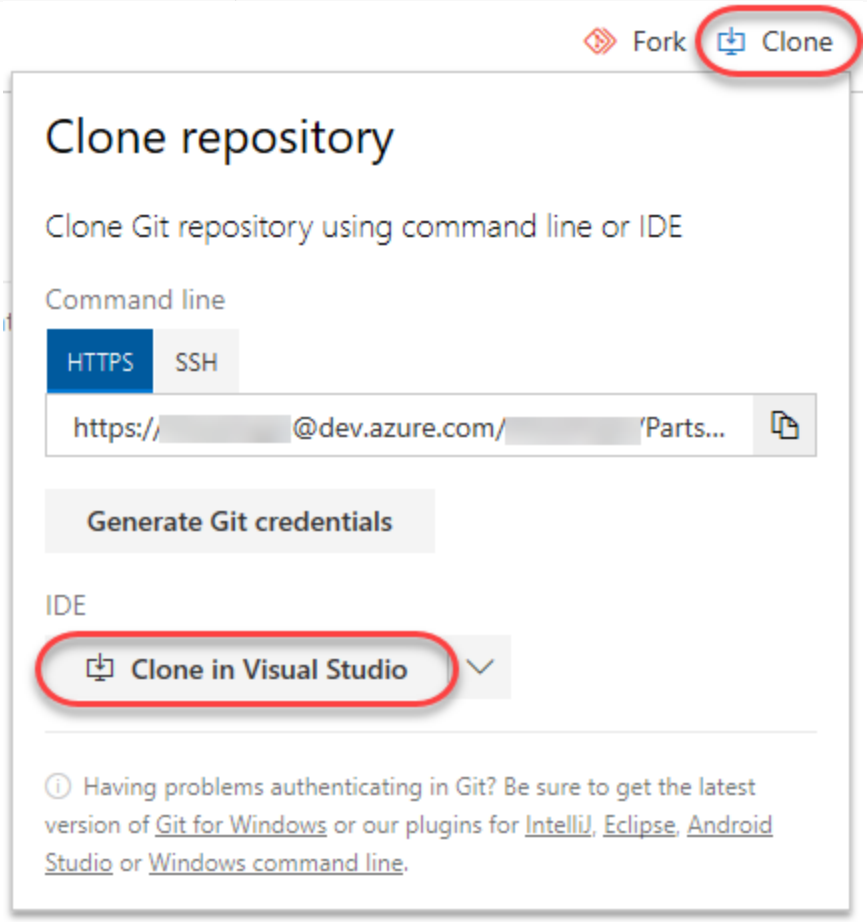
Task 2: Configuring the Parts Unlimited solution in Visual Studio

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.

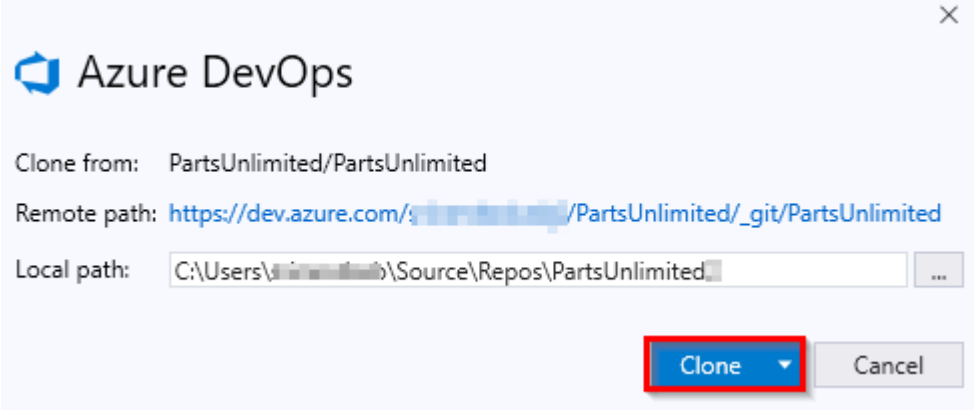


4. Click **Clone** and select **Clone in Visual Studio** (choose it in the dropdown if other option shown as default).





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

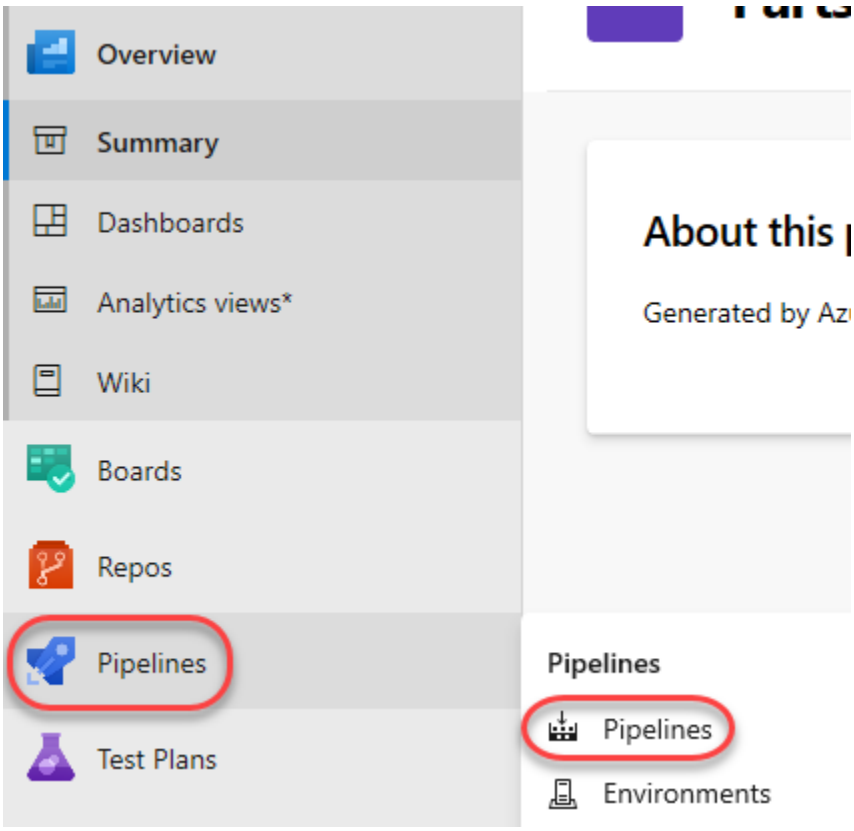


6. From **Team Explorer**, double-click **PartsUnlimited.sln** from the **Solutions** section to open the solution. You can ignore if you see any warnings about unsupported project types (just click OK on the prompted window and ignore the migration report opened on the browser)

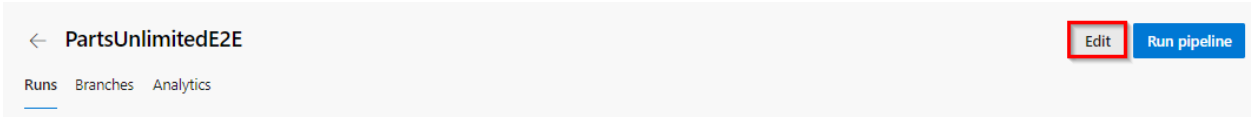
Introduction to Azure DevOps Build

Task 1: Creating a basic build pipeline from a template

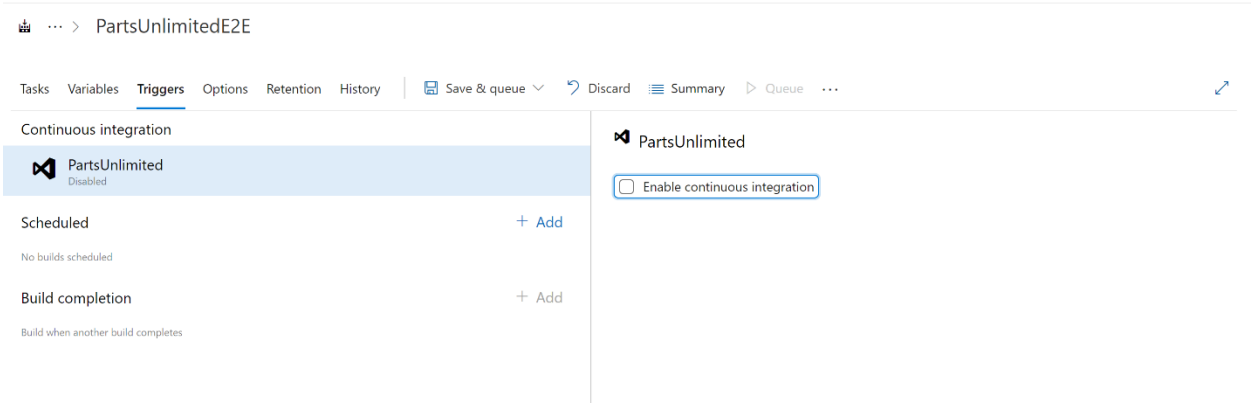
1. Navigate to your team project on Azure DevOps.
2. Navigate to **Pipelines | Pipelines**.



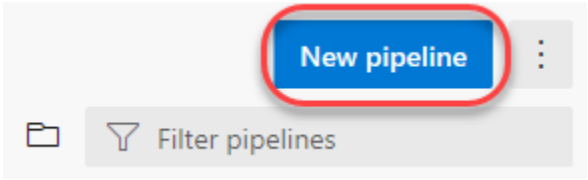
3. Open (click) the existing **PartsUnlimitedE2E** pipeline (which was created by the demo generator tool) and click on **Edit**



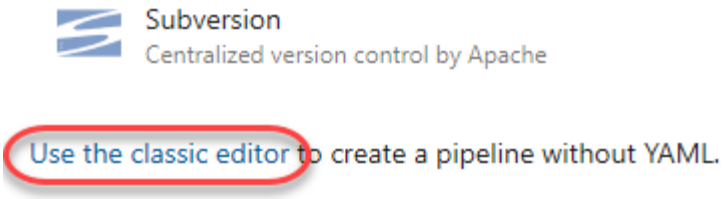
4. Not to have two pipelines triggered later in the lab, disable the CI trigger for the template created pipeline (uncheck) and **Save**.



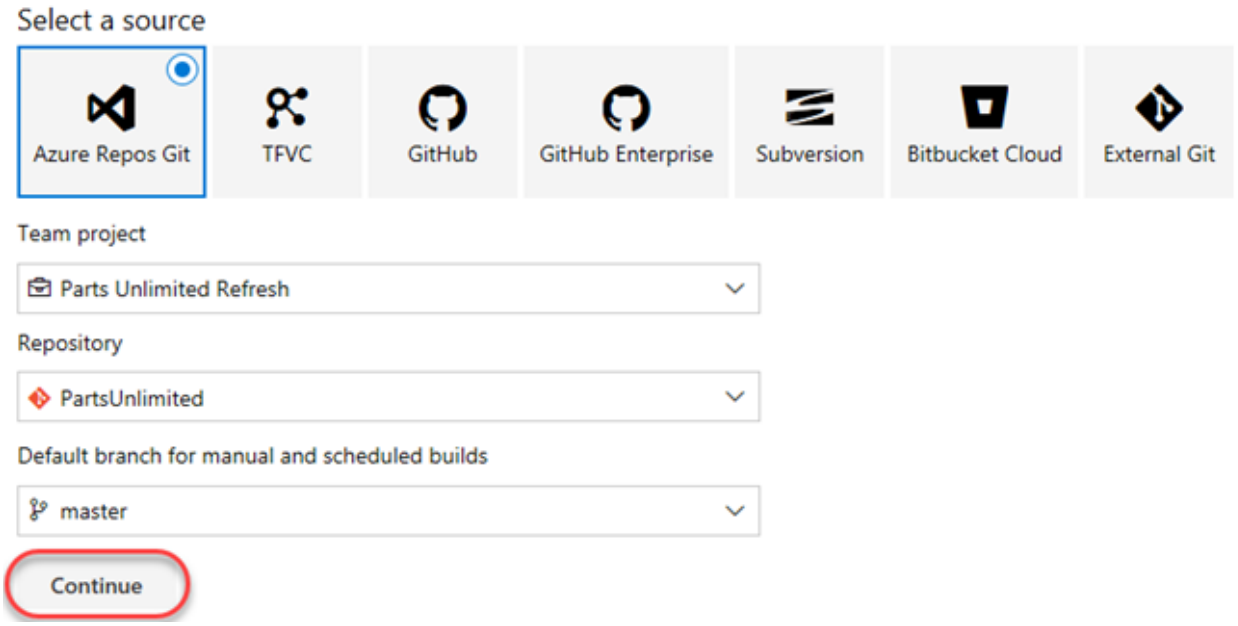
5. Navigate back to **Pipelines | Pipelines** and click **New pipeline** to create a new build pipeline.



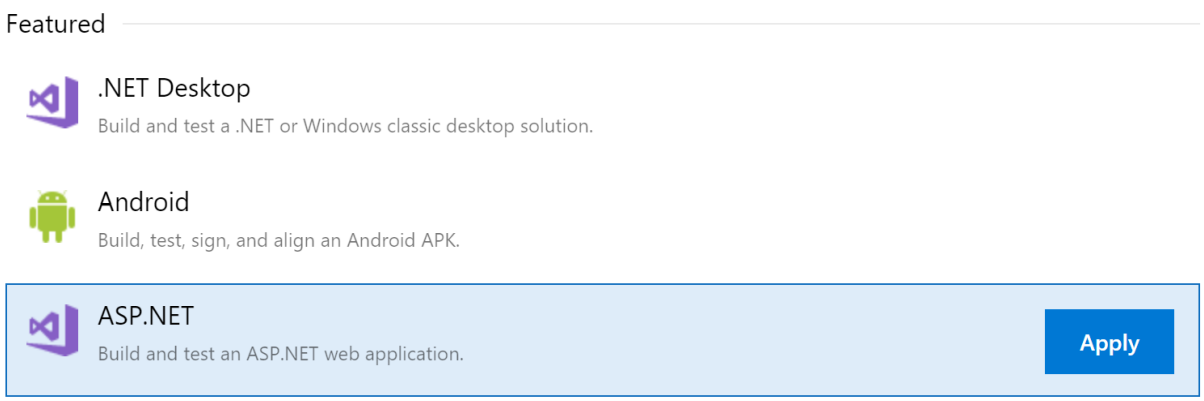
6. The default option for build pipelines involves using YAML to define the process. If you are interested in that, please check out that lab. For this lab, click **use the classic editor**.



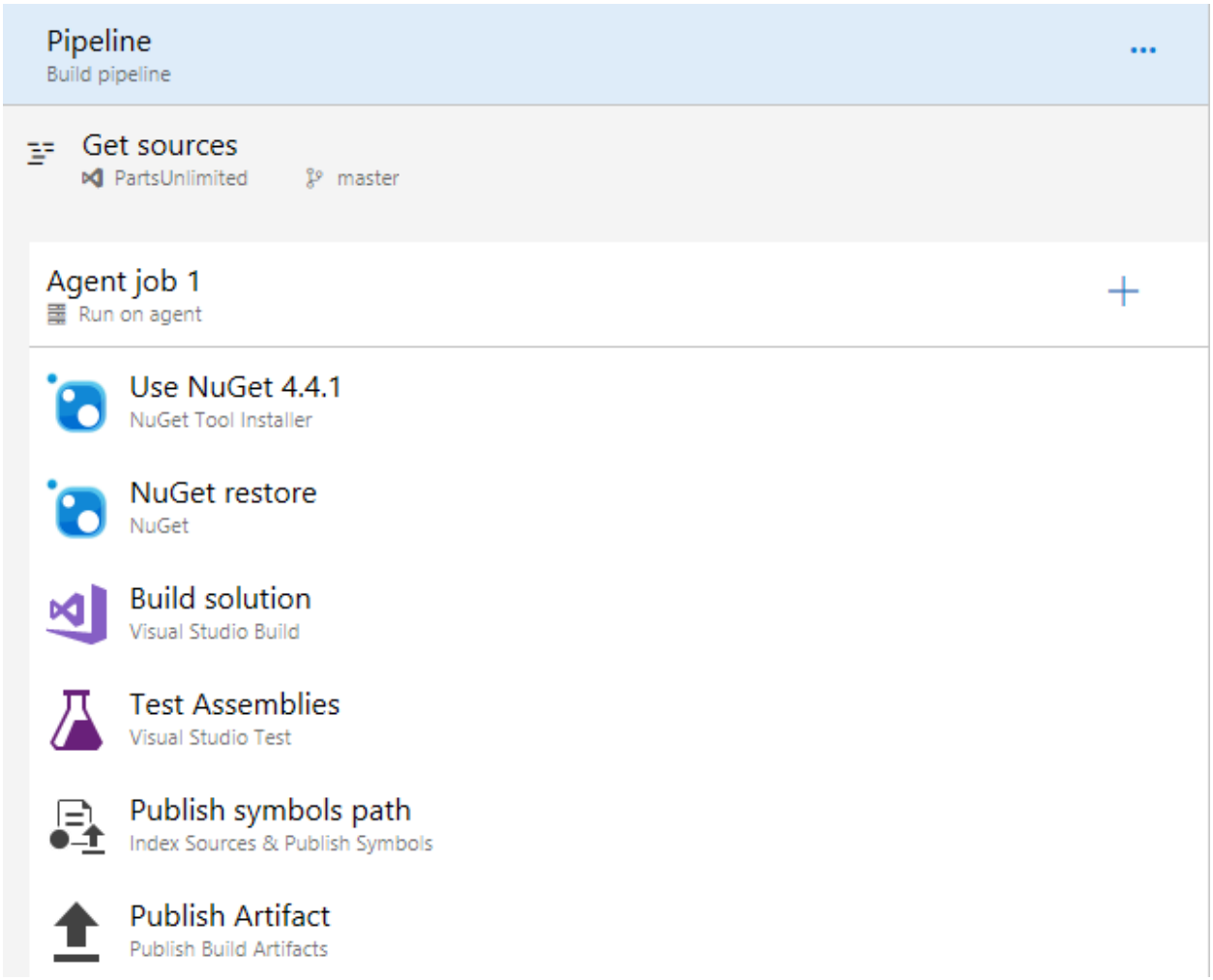
7. The first thing you'll need to do is to configure the source repository. Every major platform is available, but the default options are all we need here. This build will use the **master** branch of the **PartsUnlimited** repo. Leave the defaults and click **Continue**.



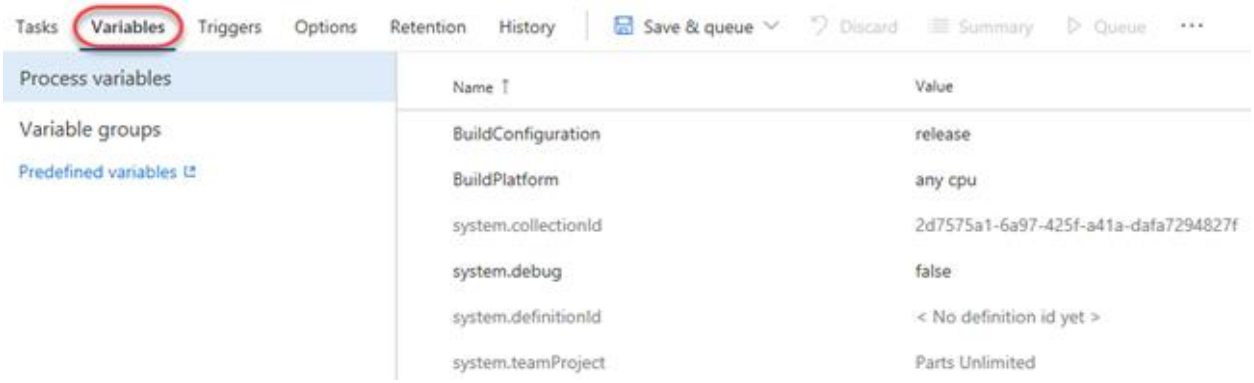
8. Locate the **ASP.NET** template and click **Apply** to apply this template to the build definition. Note that there are many options that should cover all of our mainstream scenarios. For our purposes here, we'll just build the project using the baseline ASP.NET template.



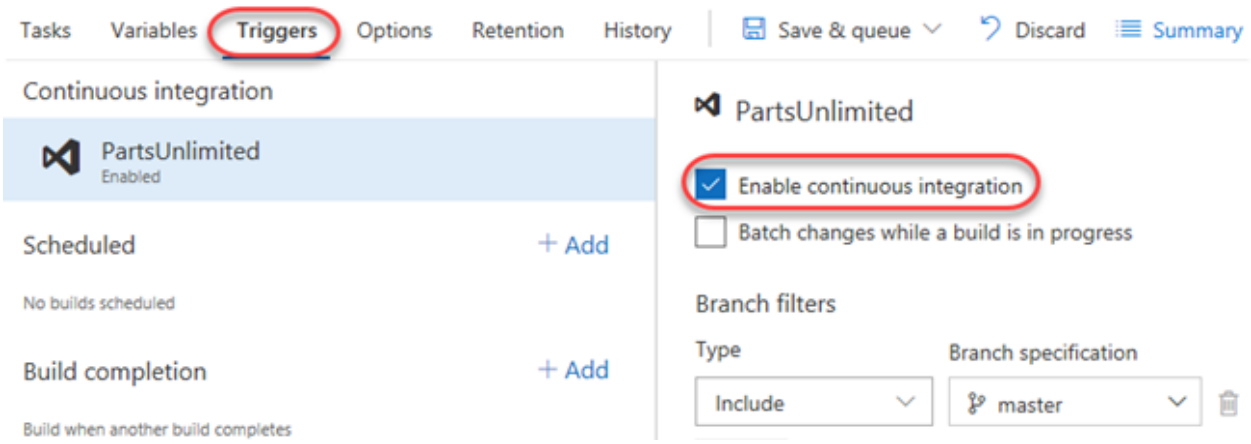
9. The process for this build pipeline is easy to follow. After getting the source, Azure DevOps will use NuGet to restore any dependent packages. Then, the project will be built and tested. The results will then be published to the configured target.



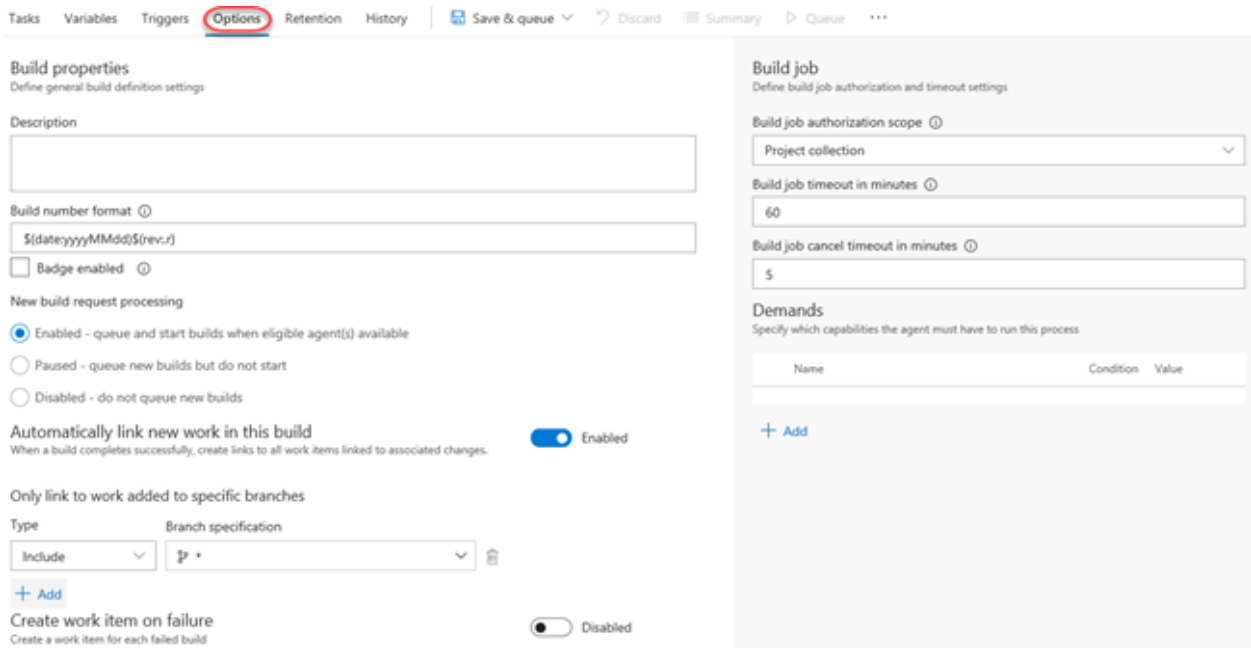
10. Select the **Variables** tab. Here you can configure special parameters to be used during the build, such as the configuration or platform.



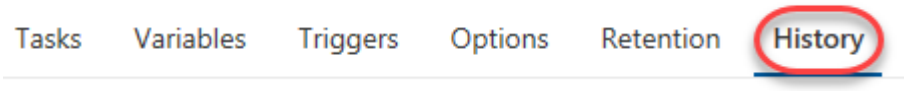
11. Select the **Triggers** tab. These triggers enable you to automatically invoke builds on a schedule, when another build completes, or when changes are made to the source. Check **Enable continuous integration** so that this build will get invoked whenever source changes are committed.



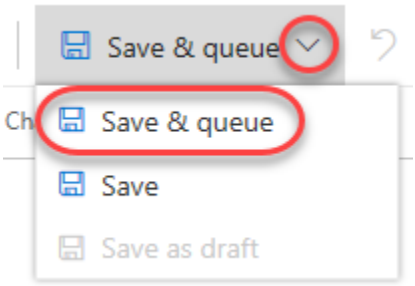
12. Select the **Options** tab. This section includes a wide variety of options related to the build workflow. Note that you'll generally configure options for specific build tasks on the configuration views of the tasks themselves.



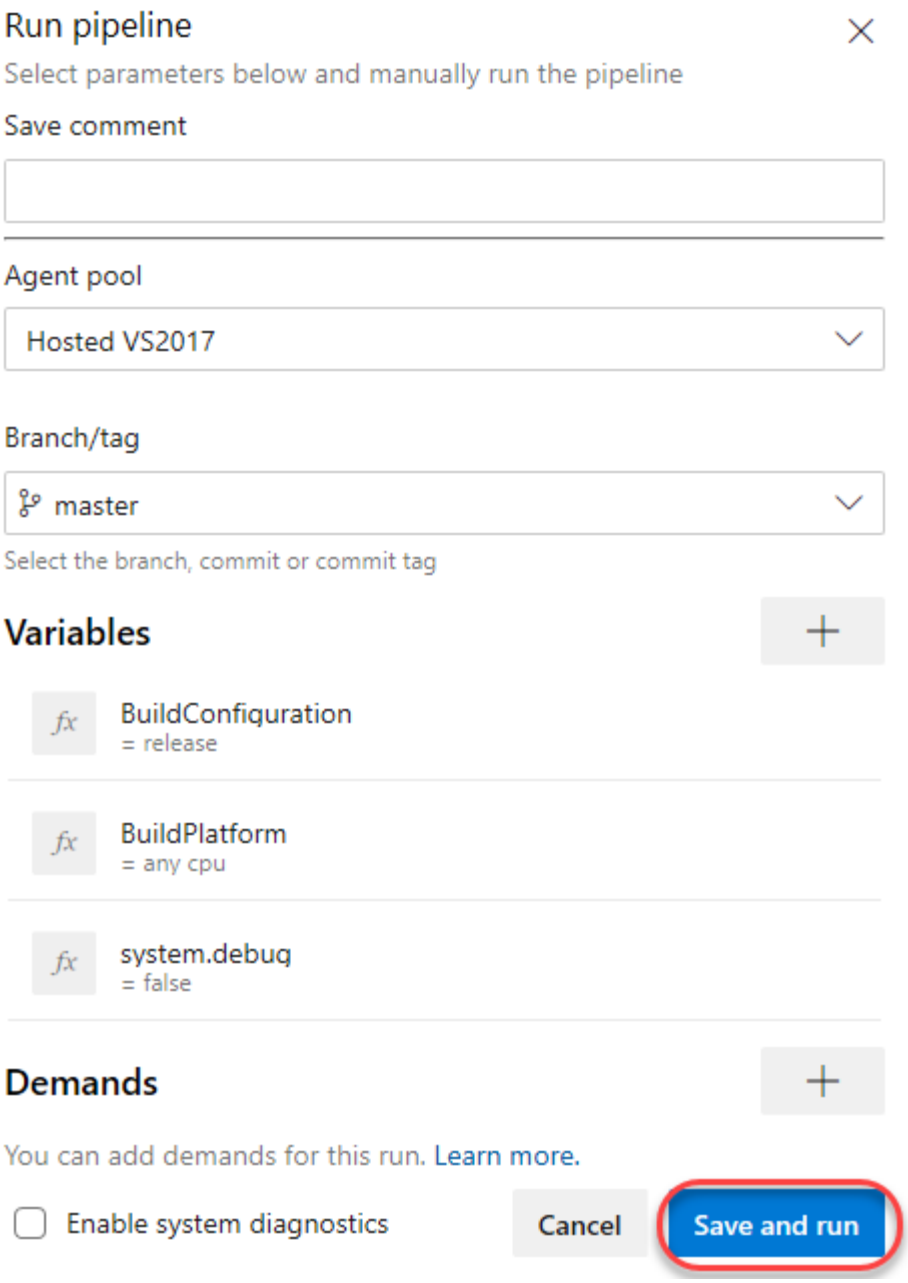
13. Select the **History** tab. There's nothing here yet, but it will show a history of changes you make to the build definition.



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



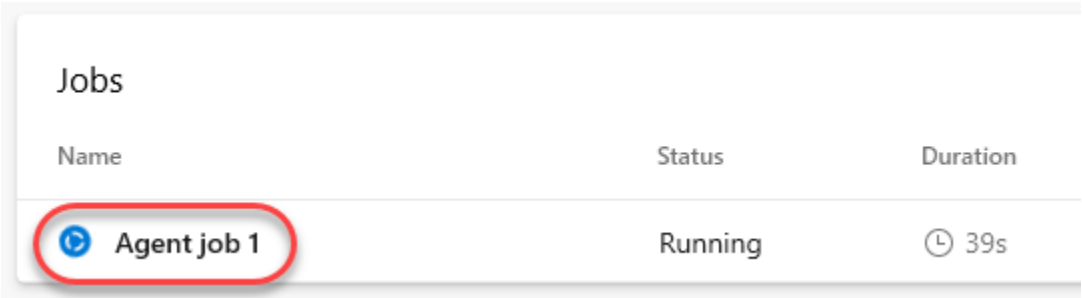
15. Accept the default options by clicking **Save and run**.



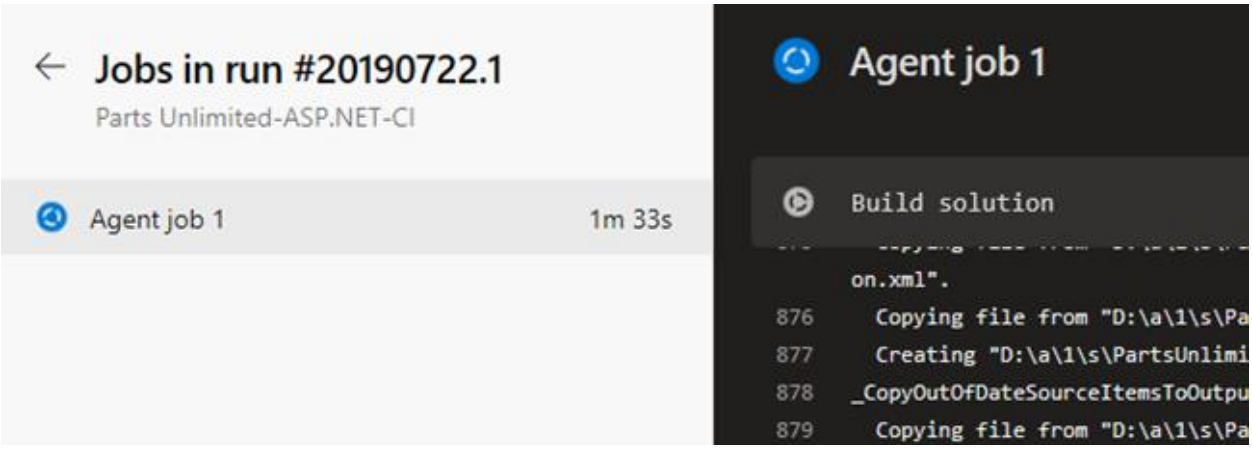
Task 2: Tracking and reviewing a build

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

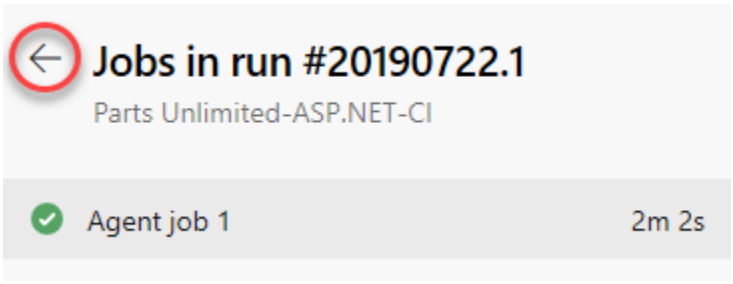
2. Once the build begins, you'll be able to track the console output per task. Click **Agent job 1**.



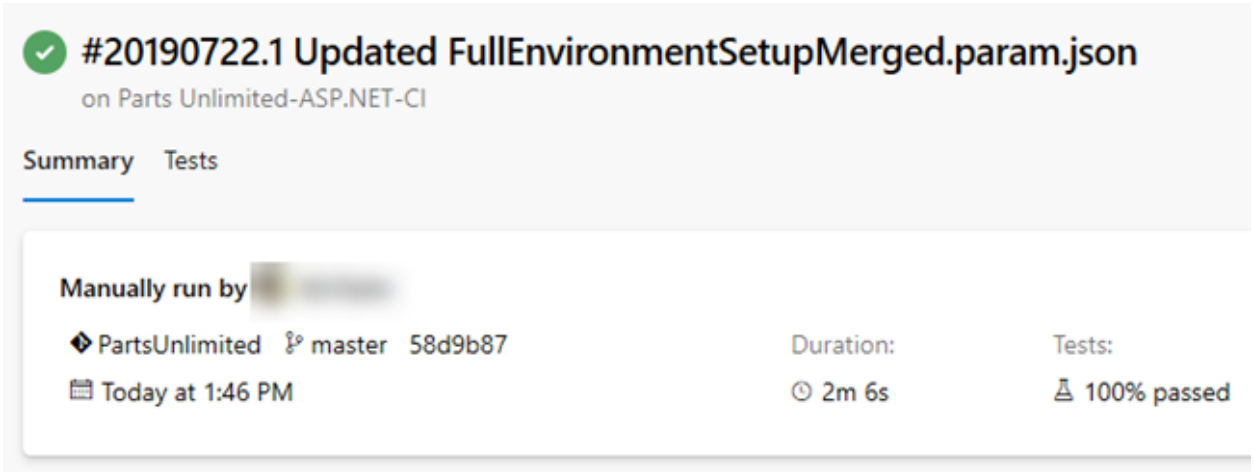
3. If you want to review an earlier task, you can scroll the right pane to review its logs.



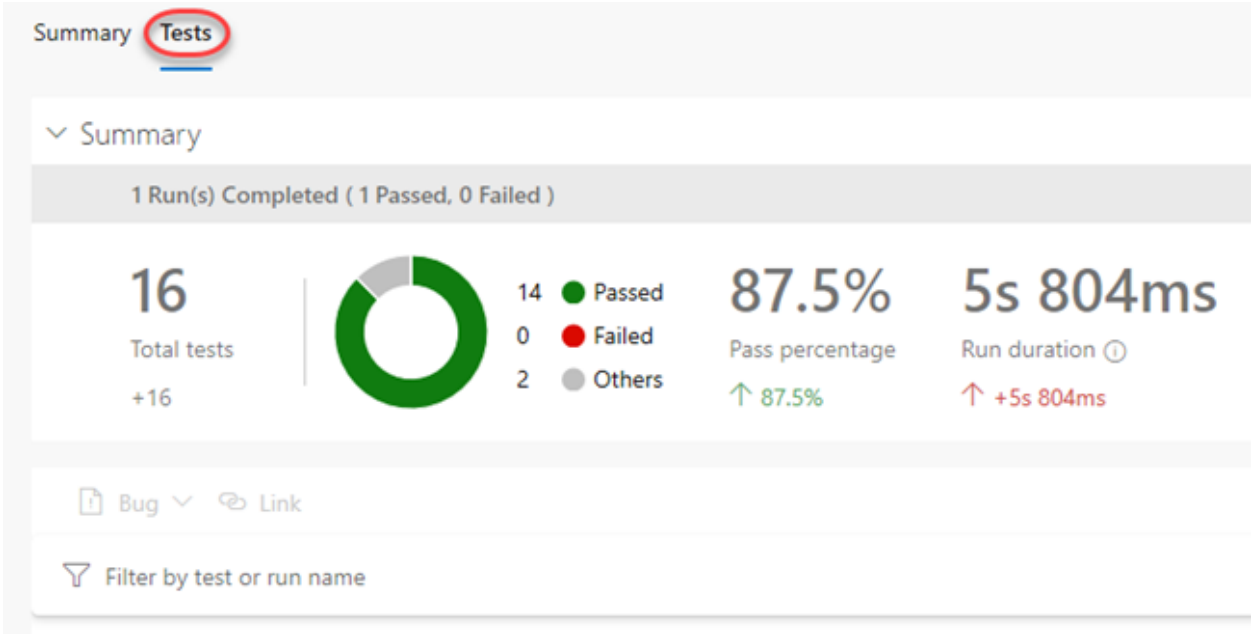
4. The build should eventually succeed. You can return to the summary view by clicking the back button.



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

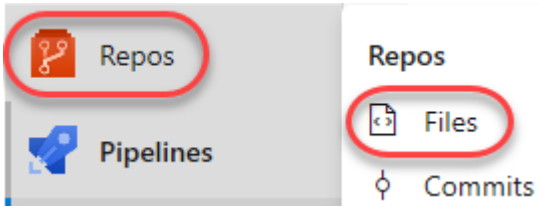


6. Select the **Tests** tab to review test performance for this build. Note that you also have easy access to the pipeline editor, the ability to queue a new build, and download the artifacts of this build.



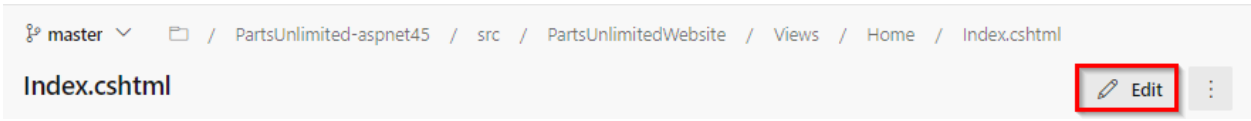
Task 3: Invoking a continuous integration build

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

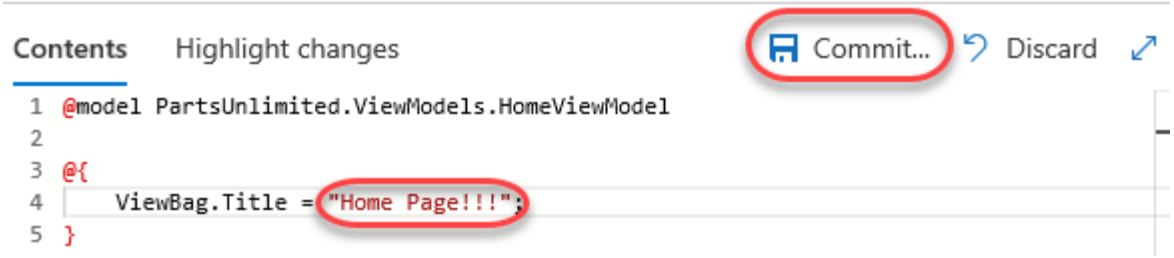


2. Open the file at **PartsUnlimited-aspnet45/src/PartsUnlimitedWebsite/Views/Home/Index.cshtml**.

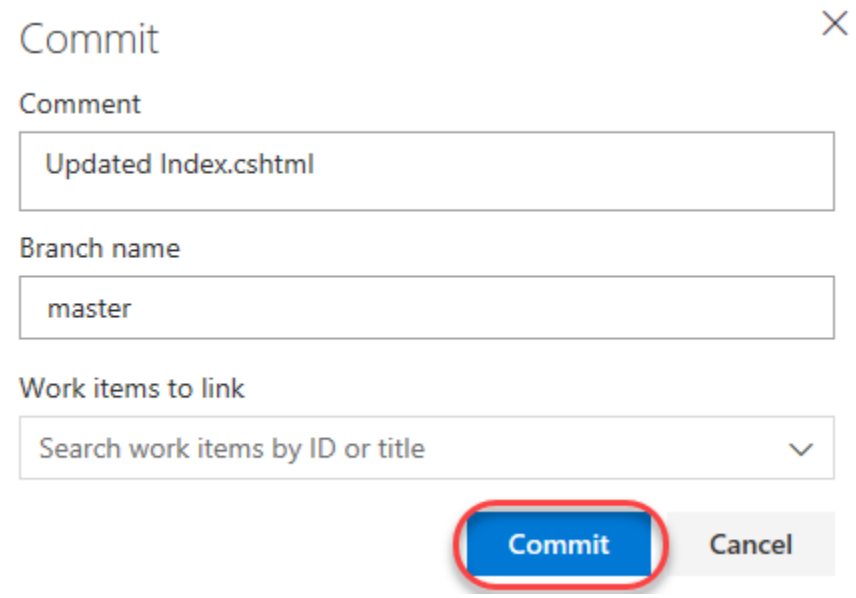
3. Click **Edit**.



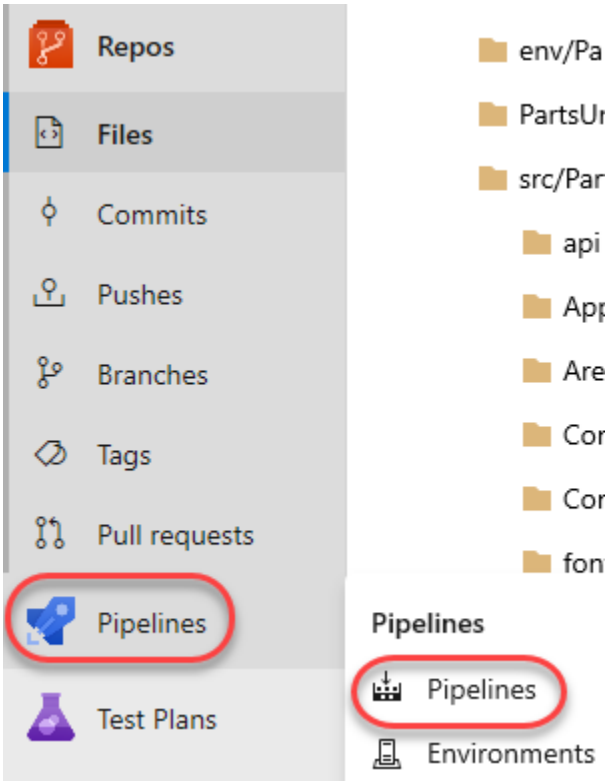
4. Make a minor cosmetic change, such as by tweaking the title of the document. Click **Commit**.



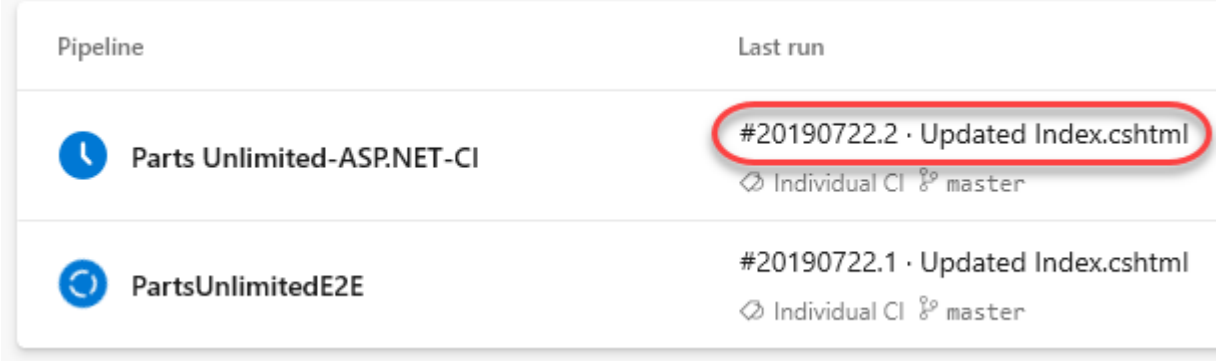
5. Accept the default commit details and click **Commit**.



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.



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