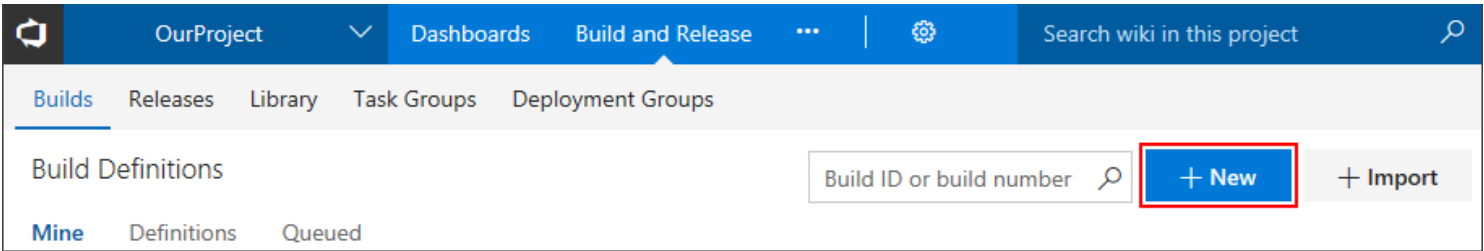


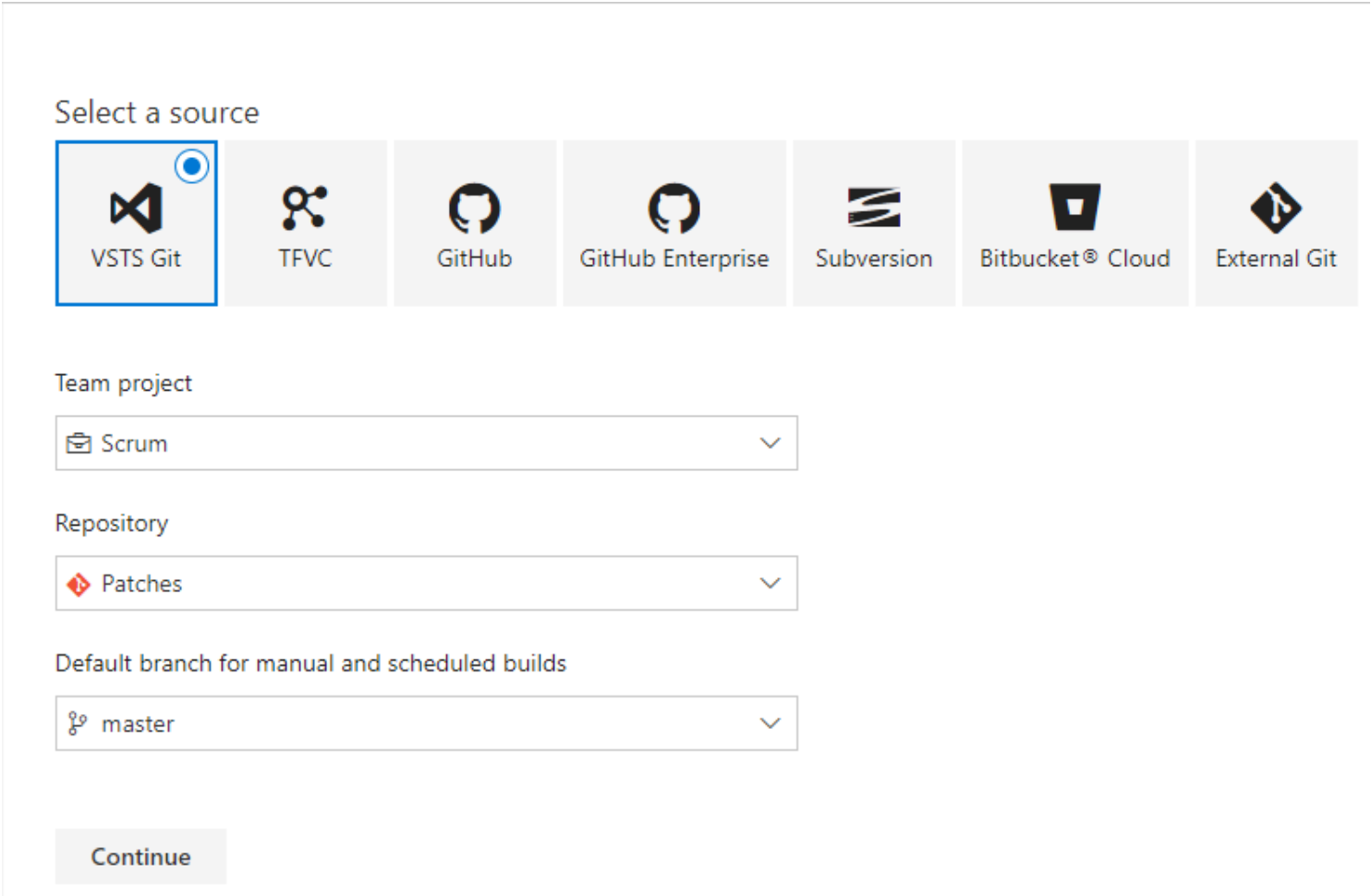
Step By Step: Setup a CI Build

Create a Build Definition

- 1. Navigate to VSTS and select your project <https://globaldevopsbootcamp.visualstudio.com>
- 2. Navigate to the "Build and Release" hub
- 3. Create a "New" definition



- 1. Configure the sources, select your correct team project, repository and branch. Then click "Continue".



- 1. Select a template, use "ASP.NET", click "Apply"



Config as code



YAML

Configure your build as code.

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

Apply

1. Specify / change your build name and select the private build agent that you have installed as agent queue
2. If your repository contains multiple solutions, select the specific solution

View YAML

Name *

Scrum-ASP.NET-CI

Agent queue * | [Manage](#)

Hosted VS2017

Parameters | [Unlink all](#)

Path to solution or packages.config *

***.sln

Artifact Name *

drop

1. Select the Visual Studio Build step, notice the MSBuild Arguments, this will provide a published

Tasks

Variables

Triggers

Options

Retention

History

Save & queue

Process

Build process



Get sources

Patches

master

Phase 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


1. Select the "Test Assemblies" step, review the settings. Change the value for testassemblies from `**\$(BuildConfiguration)*test*.dll` to `**\$(BuildConfiguration)*nunit.unittest*.dll`. This will only run the NUnit unit tests after the build.


Tasks Variables Triggers Options Retention History |  Save & queue ▾


Process
Build process


 Get sources
 Patches  master


Phase 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

Enable Continuous Integration for your build ##

1. While editing your build definition, navigate to the "Triggers" tab
2. Check the "Enable continuous integration" option

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

Continuous integration

Patches
Disabled

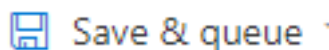
☐ Enable continuous integration

Scheduled + Add
No builds scheduled

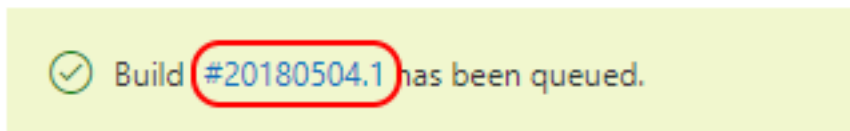
Build completion + Add
Build when another build completes

Run the build

1. Press "Save & queue" to save and trigger the new build



1. Notice the feedback for the build being queued. Select the link to navigate to the build



1. VSTS Provides direct feedback from the build agent

Build 20180504.2

Build

Job

- Initialize Agent
- Initialize Job
- Get Sources
- NuGet restore ***.sln
- Build solution Sample Web Ap...
- Test Assemblies **\release*tes...
- Publish symbols path:
- Copy Files to: \$(build.artifactst...
- Publish Artifact: drop
- Post Job Cleanup

Sample Web Application / Build 20180504.2 / Build / Job

Edit build definition Cancel Queue new build... Download all logs

Build Started

Job 8
Running for 7 seconds (Hosted Agent)

Console Timeline Code coverage* Tests WhiteSource Bolt Build Report

```

Downloading task: VSBuild
Downloading task: VSTest
Downloading task: PublishSymbols
Downloading task: CopyFiles
Downloading task: PublishBuildArtifacts
*****
Finishing: Initialize Job
*****
Starting: Get Sources
*****
Syncing repository: Scrum (TfsGit)
Prepending Path environment variable with directory containing 'git.exe'.
git version

```

```
git version 2.14.3.windows.1
git lfs version
git-lfs/1.5.2 (GitHub; windows amd64; go 1.7.3; git 547a06aa)
git init "D:\a\1\s"
```

1. After the build succeeds, refresh the browser
2. Notice the new "Artifacts" tab, and click it.

The screenshot shows the Azure DevOps build interface for a 'Sample Web Application' build 20180504.3. The build status is 'Build succeeded'. The left sidebar shows the build phases and steps, including 'Initialize Agent', 'Initialize Job', 'Get Sources', 'NuGet restore', 'Build solution', 'Test Assemblies', 'Publish symbols path', 'Copy Files to', 'Publish Artifact: drop', and 'Post Job Cleanup'. The main area shows the 'Artifacts' tab, which lists the artifact 'drop' with a 'Download' button and an 'Explore' button. The 'Explore' button is circled in red. A red arrow points from the 'Explore' button to the 'Artifacts Explorer' dialog box. The 'Artifacts Explorer' dialog box shows a tree view with the 'drop' folder expanded, and the file 'Sample Web Application.zip' is highlighted with a red circle. Other files in the list include 'Sample Web Application.deploy-readme.txt', 'Sample Web Application.deploy.cmd', 'Sample Web Application.SetParameters.xml', and 'Sample Web Application.SourceManifest.xml'.

1. Select the "Explore" button behind the created artifact.
2. Expand the "Drop" folder
3. Check & Verify the desired output is part of the artifact. In this case a ".zip" file containing the application.

Check that CI is triggered when a change to the code is pushed

1. Open the MVC Music Store solution in Visual Studio
2. Make a change to the web.config by adding a newline or a space in the file
3. Right click the web.config and click on Commit
4. Enter the required commit message and select Commit all and Push in the dropdown. This will push your change to the VSTS repo and trigger a new CI build.
5. Open VSTS in the browser
6. Click on the Build and Release tab
7. Click on Builds. You should see the build that you just triggered