Azure Developer Immersion

VSTS Build

No matter what tools you use, in whatever language you prefer, Visual Studio Team Services Build builds your app, your way, for your platforms. Just open your web browser to tell it how you want it done. You can build for Windows, iOS, Android, Java (Ant, Maven, or Gradle), or Linux using the same domain-specific languages (DSL) you use every day on your dev machine. The build services even build Xamarin apps for both iOS and Android and run tests on Xamarin's Test Cloud as part of the build.

In this walkthrough, you will create a "team build" in Visual Studio Team Services. You will start with a manual build, and then configure one for continuous integration.

There are two exercises in this walkthrough:

1. Create a Manual Build
2. Create a CI Build

Expected duration: 30 minutes

Exercise 1: Create a Manual Build

In this exercise, you will learn how to create a manual Team Build that runs in Visual Studio Team Services. This makes it quick and easy to start building your code, either manually or in an automated fashion, without having to worry about any build server configuration.

1. Open the **Developer Command Prompt for VS 2015**.

2. Change to the directory **Rg.ClientApp** project's directly by typing the following and pressing **ENTER**:

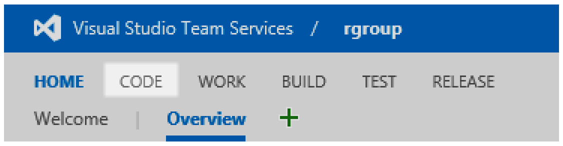
```
cd C:\VSTS\Repos\rGroup\Before\Rg.ClientApp
```

3. You're now going to "add" the **Rg.ClientApp_TemporaryKey.pfx** file so you can commit and push it to VSTS. Be sure to capitalize the file exactly as it is seen below or your file will not be staged.

```
git add Rg.ClientApp_TemporaryKey.pfx -f
```

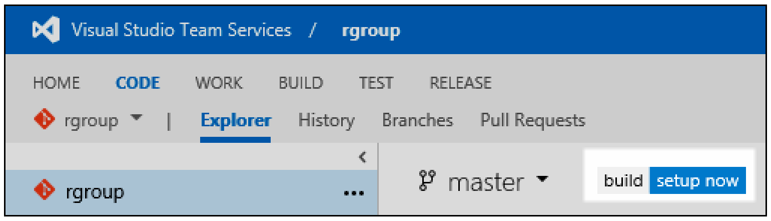
Typically, you don't commit PFX files to your repo, which is why it is ignored by Visual Studio and Git by default (this is controlled via the .gitignore file). However, you need the PFX file to complete the build process for the Windows Store app. As this is a temporary key that can't be used for publishing to the store, it's not critical.

4. Type **exit** and press **ENTER** to close the command prompt.
5. Return to Visual Studio.
6. In Team Explorer, click the **Changes** button. If you don't see the button, click the **Home** button and try again.
7. For a commit message, enter **Add Windows Store app certificate to repo**.
8. Click the **Commit Staged and Push** button.
9. On the **Home** screen in **Team Explorer**, click the **Web Portal** link to open your VSTS Team Project's Home page. If you already have a tab open in a browser window, feel free to use it.
10. Click the **CODE** hub link.

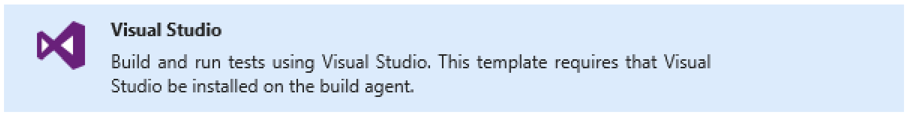


In the CODE hub, you can easily examine your repo's history, manage branches, and view pull requests. In addition, it provides an easy way to get started when creating a team build.

11. To the right of the master branch, click the **setup now** button.



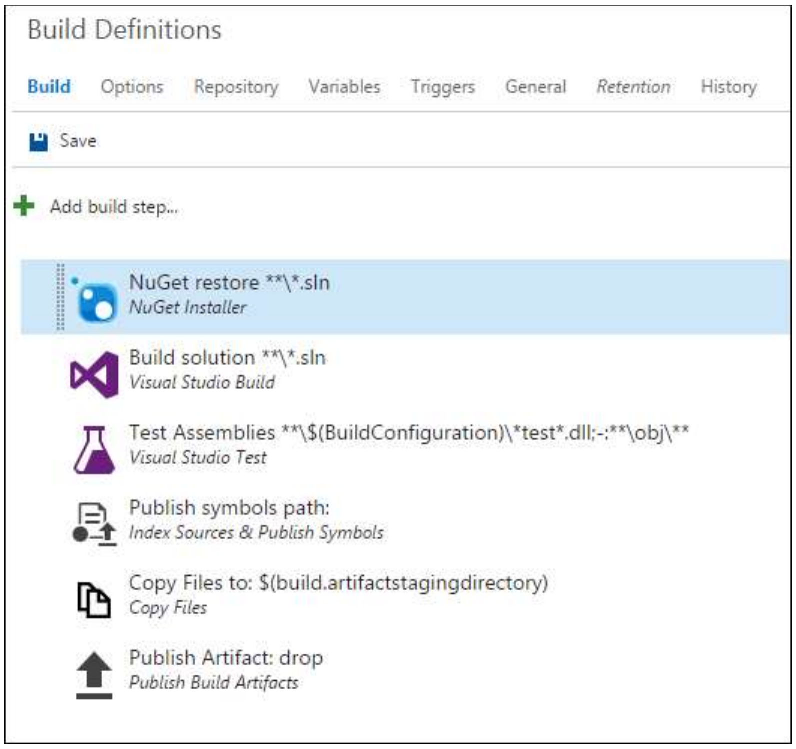
12. In the **Create new build definition** dialog, select the Visual Studio template and click **Next**.



Build and run tests using Visual Studio. This template requires that Visual Studio be installed on the build agent.

13. On the next page, you'll see you can configure which repo and branch to use if you have more than one of either. Accept the defaults and click **Create**.

Visual Studio Team Services opens a new browser tab in the BUILD hub where you can continue editing the Build Definition.


Build Definitions

Build | Options | Repository | Variables | Triggers | General | *Retention* | History

Save

Add build step...

NuGet restore ***.sln
NuGet Installer

Build solution ***.sln
Visual Studio Build

Test Assemblies **\$(BuildConfiguration)*test*.dll;-:**\obj**
Visual Studio Test

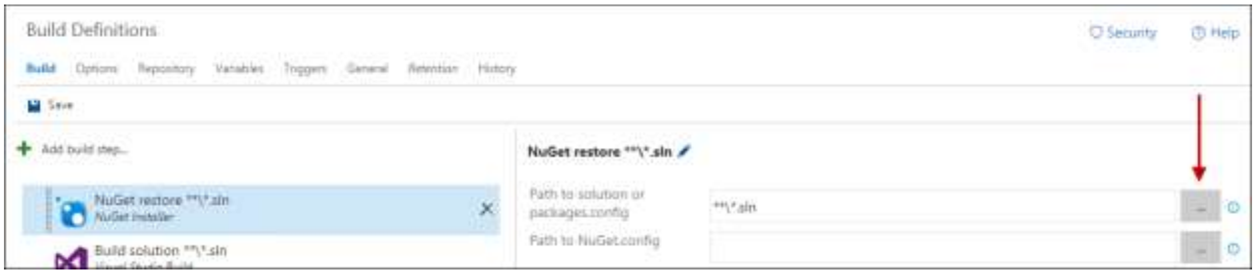
Publish symbols path:
Index Sources & Publish Symbols

Copy Files to: $(build.artifactstagingdirectory)
Copy Files

Publish Artifact: drop
Publish Build Artifacts

A Team Build is composed of multiple steps or tasks that are executed in sequence. The default build definition template for Visual Studio solutions does a number of steps including running tests and saving your build output to VSTS, where you can download the compiled version. For your purposes, the default template will “just work”, but you’re going to tweak it a bit to be more efficient.

14. The first step should already be selected—NuGet Installer. Click the **…** button at far right of the **Path to Solution** option.


Build Definitions

Security | Help

Build | Options | Repository | Variables | Triggers | General | Retention | History

Save

Add build step...

NuGet restore ***.sln
NuGet Installer

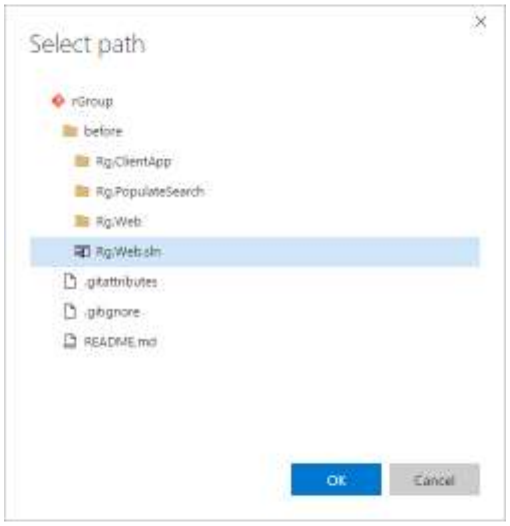
Build solution ***.sln

NuGet restore ***.sln

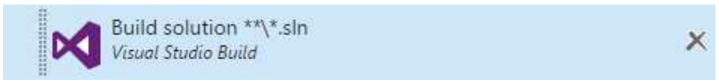
Path to solution or packages.config: ***.sln

Path to NuGet.config:

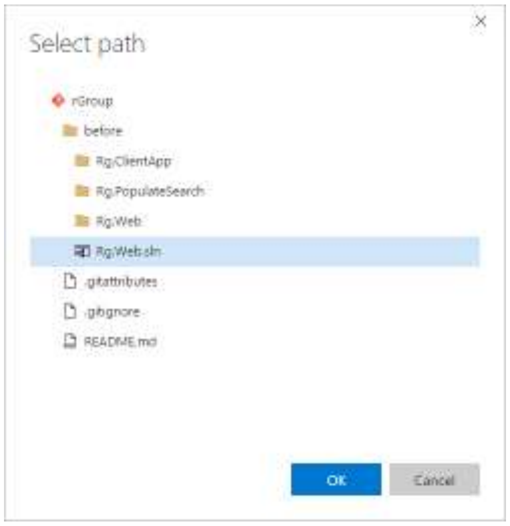
15. In the SELECT PATH dialog, select the **Before/Rg.Web.sln** file and click **OK**.



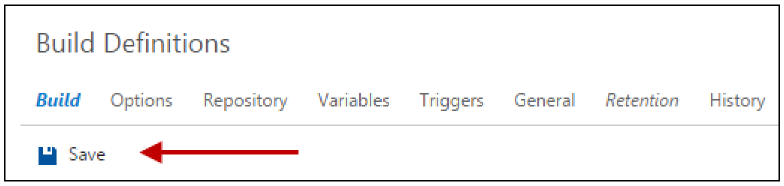
16. Select the Visual Studio Build step.



17. To the right, click the **…** button at far right of the **Solution** option.
18. In the SELECT PATH dialog, select the **Before/Rg.Web.sln** file and click **OK**.

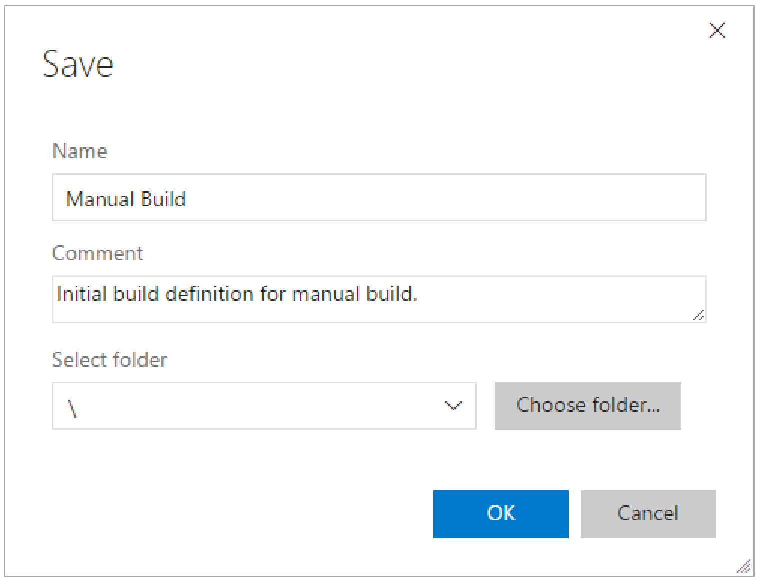


19. Click the **Save** button to save your build definition.



20. In the SAVE dialog type **Manual Build** for the **Name**.

21. In the **Comment** field, type **Initial build definition for manual build**. A great feature of Team Build is that VSTS versions your build definitions so you can see when and why a change was made. Once you have two or more, you can diff the JSON-based definition.



Save

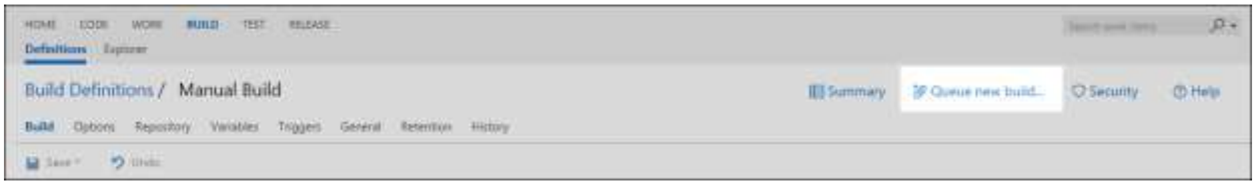
Name: Manual Build

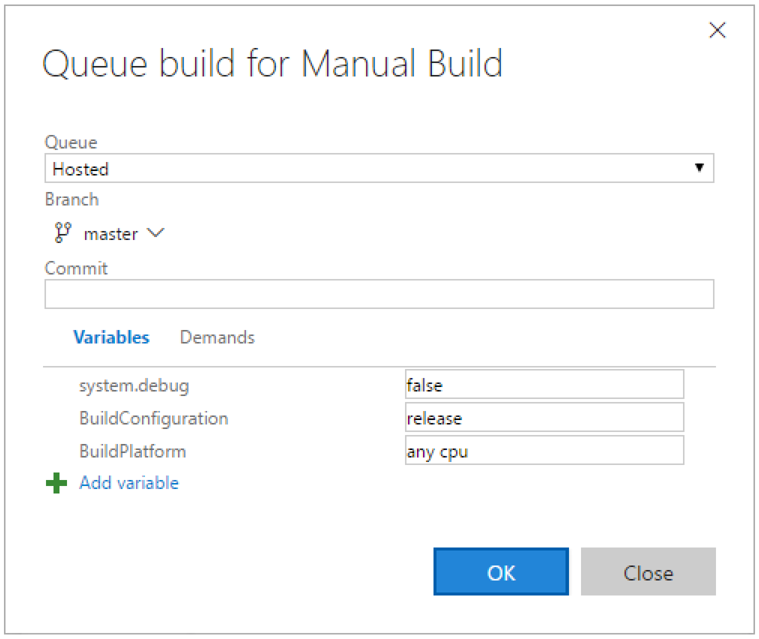
Comment: Initial build definition for manual build.

Select folder: \

22. Click **OK**.

23. Click the **Queue new build...** button to open the **QUEUE BUILD FOR MANUAL BUILD** dialog.





Queue build for Manual Build

Queue: Hosted

Branch: master

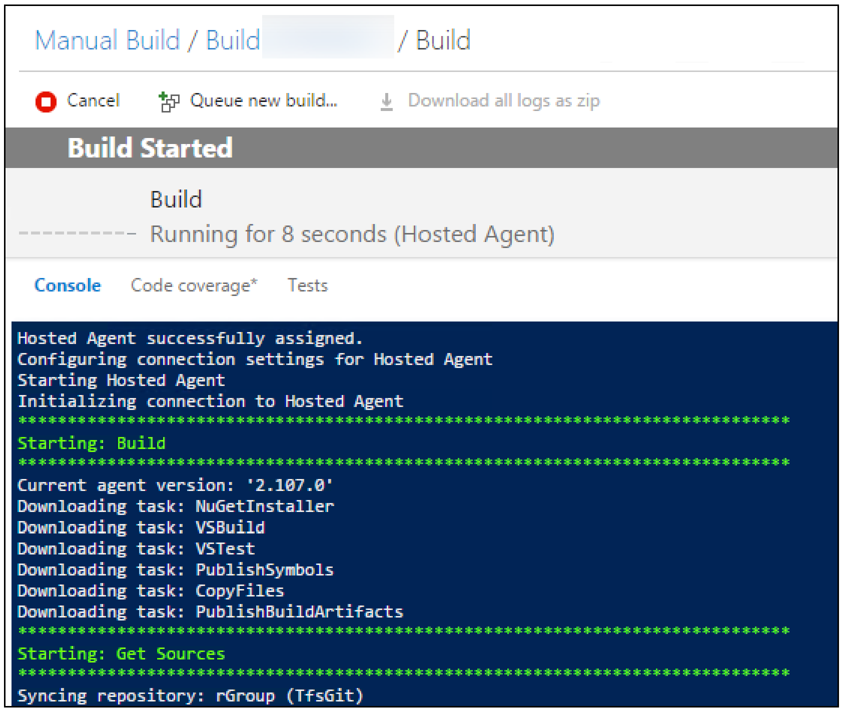
Commit:

system.debug	false
BuildConfiguration	release
BuildPlatform	any cpu

The **Queue build for Manual Build** dialog provides a number of options to run your team build. These are all read from the build definition. You can change these on a build by build basis or just accept the default settings. For this build, you'll be using the VSTS Hosted Agent. Your VSTS account comes with 240 build minutes a month. If you need more than that, you can rent a hosted agent or set up your own agent in a VM in Azure or on premises.

24. Click **OK** to queue the build.

    VSTS will queue the build and keep you posted on its progress.

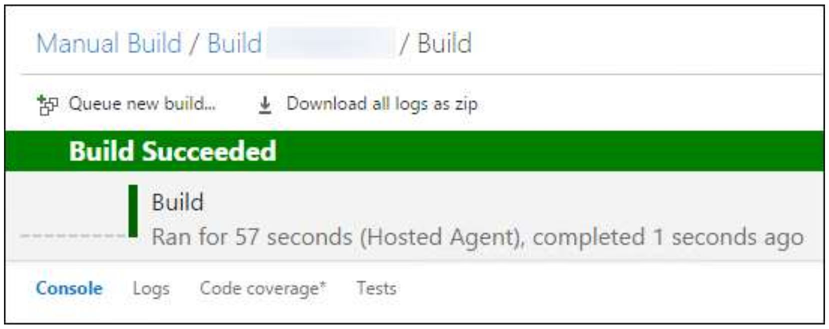

Manual Build / Build [redacted] / Build

Build Started

Build
Running for 8 seconds (Hosted Agent)

```
Hosted Agent successfully assigned.
Configuring connection settings for Hosted Agent
Starting Hosted Agent
Initializing connection to Hosted Agent
******************************************************************************
Starting: Build
******************************************************************************
Current agent version: '2.107.0'
Downloading task: NuGetInstaller
Downloading task: VSBuild
Downloading task: VSTest
Downloading task: PublishSymbols
Downloading task: CopyFiles
Downloading task: PublishBuildArtifacts
******************************************************************************
Starting: Get Sources
******************************************************************************
Syncing repository: rGroup (TfsGit)
```

25. Monitor the build until it completes. It can take anywhere from 45 seconds to a few minutes depending upon load on the hosted build service.

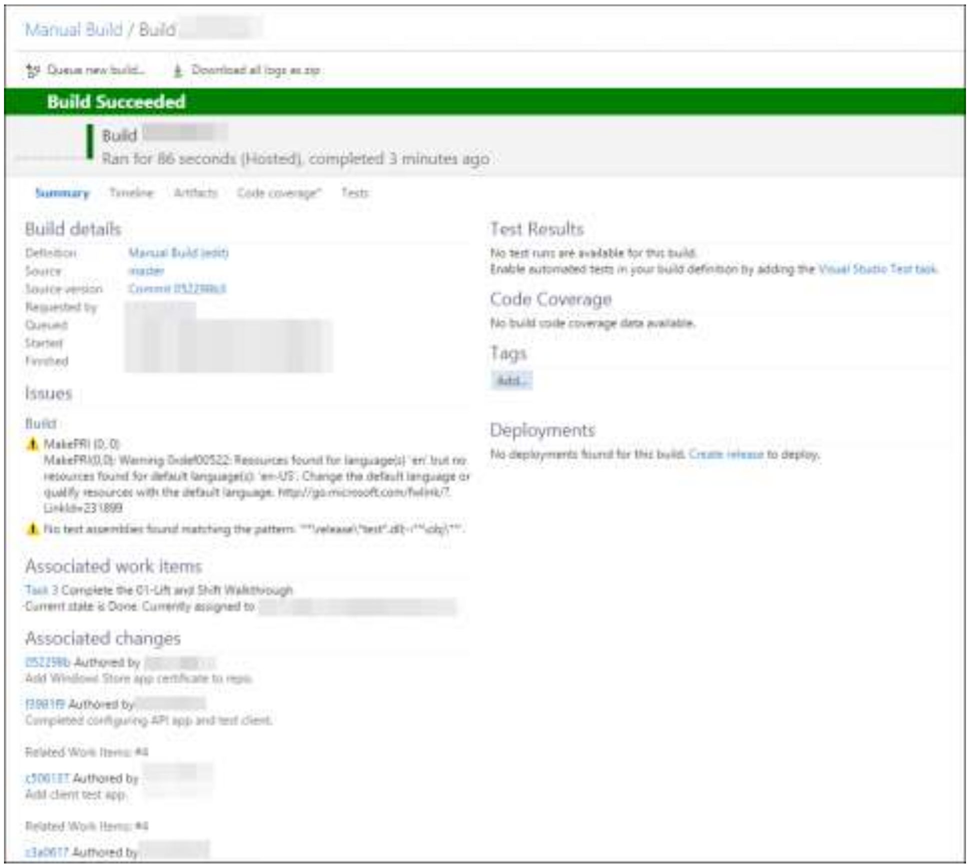

Manual Build / Build [redacted] / Build

Build Succeeded

Build
Ran for 57 seconds (Hosted Agent), completed 1 seconds ago

26. Once it's done, you examine the build details by click the "Build #" link, where # is the build run number.

The build report will tell you many details, including related commits, associated work items, and test results.



Manual Build / Build [redacted]

Queue new build... Download all logs as zip

Build Succeeded

Build [redacted]
Ran for 86 seconds (Hosted), completed 3 minutes ago

Summary Timeline Artifacts Code coverage* Tests

Build details

Definition	Manual Build (edit)
Source	master
Source version	Commit 0522986
Requested by	[redacted]
Queued	[redacted]
Started	[redacted]
Finished	[redacted]

Issues

Build

⚠ MakePRI (0, 0)
MakePRI(0,0): Warning 0xdef00522: Resources found for language(s) 'en' but no resources found for default language(s): 'en-US'. Change the default language or qualify resources with the default language. http://go.microsoft.com/fwlink/?LinkId=231899

⚠ No test assemblies found matching the pattern: "**\release*test*.dll;-:**\obj**".

Associated work items

Task 3 Complete the 01-Lift and Shift Walkthrough
Current state is Done. Currently assigned to [redacted]

Associated changes

052298b Authored by [redacted]
Add Windows Store app certificate to repo.

f3981f9 Authored by [redacted]
Completed configuring API app and test client.

Related Work Items: #4

c506137 Authored by [redacted]
Add client test app.

Related Work Items: #4

c3a0617 Authored by [redacted]

Test Results

No test runs are available for this build.
Enable automated tests in your build definition by adding the Visual Studio Test task.

Code Coverage

No build code coverage data available.

Tags

Add...

Deployments

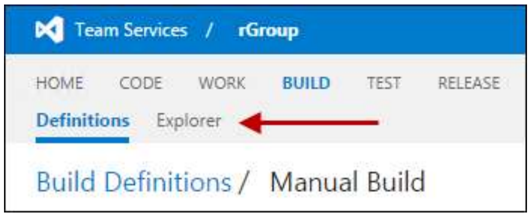
No deployments found for this build. Create release to deploy.

27. At this point, a build will only run if you ask queue it manually. Continue to the next exercise where you will add support for continuous integration.

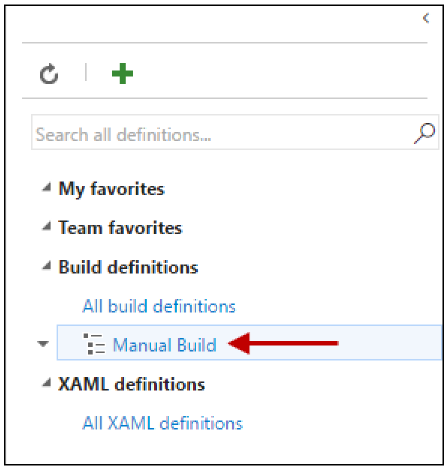
Exercise 2: Create a CI Build

Let’s take the manual build we created and create a continuous integration or “CI” build.

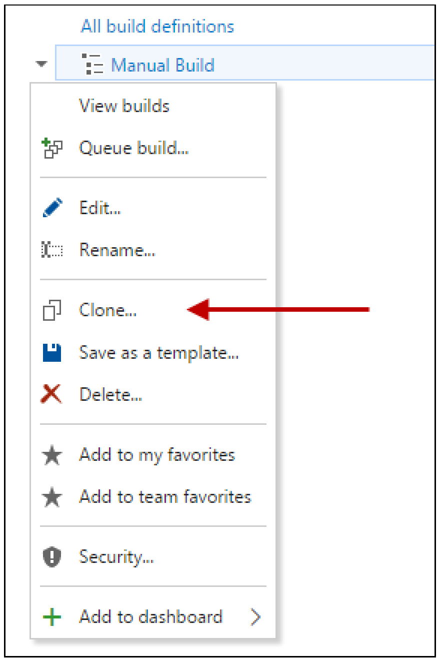
1. Click the **BUILD** hub link.
2. Click the **Explorer** link.



3. Select the **Manual Build**.

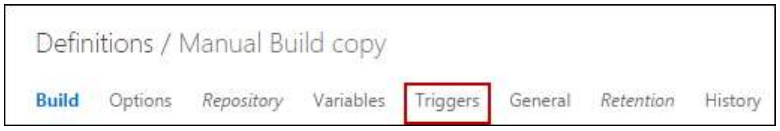


4. Select the arrow next to **Manual Build** and select **Clone…** from the menu options.

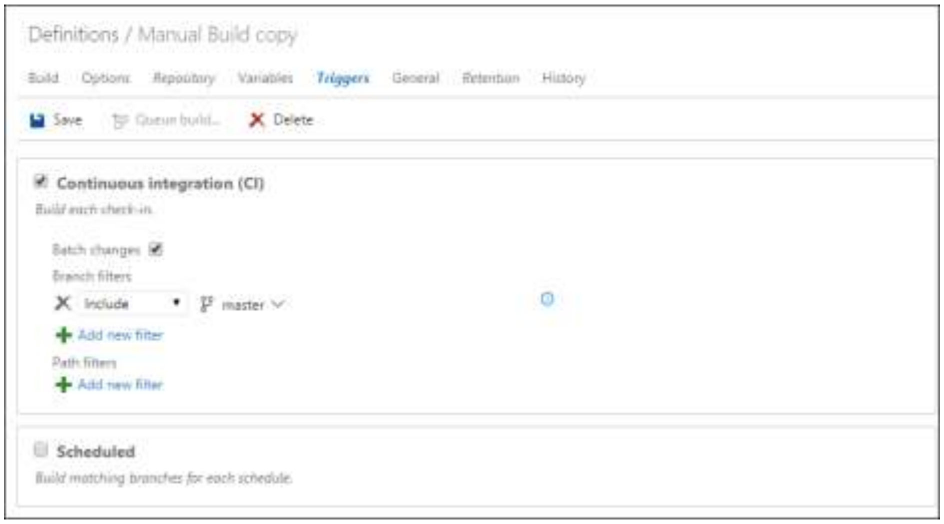


This creates a copy of the build definition called **Manual Build copy**. It also opens to the new build definition.

5. Select the **Triggers** tab.



6. On the **Triggers** tab, check the box next to **Continuous Integration (CI)**. Accept the defaults.



✓ Continuous integration (CI)
Build each check-in
Batch changes ✓
Branch filters
Include master
Path filters

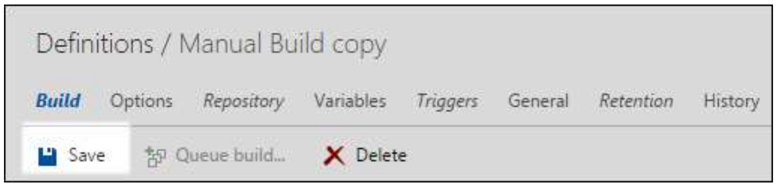
☐ Scheduled
Build matching branches for each schedule.

7. Click the **Build** tab. You're going to remove some unnecessary steps to make the CI build run faster.

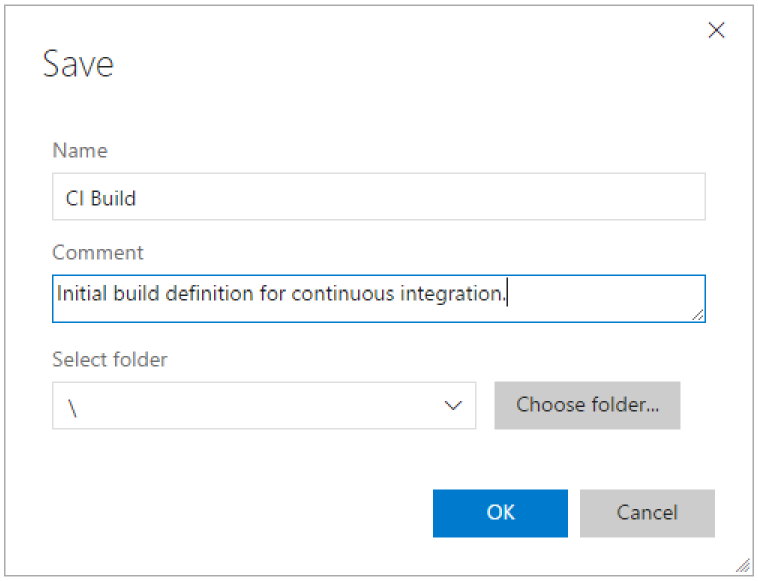
8. Remove the following steps by clicking the button:
   * Publish Symbols
   * Copy Files to
   * Publish Artifact

   Generally, you want your CI builds to run as fast as possible to give you and your team members immediate feedback if you missed any files or broke something. Since the current sample doesn't have any tests, you could also remove the **Test Assemblies** task. But it's better to leave it as a reminder that you should add some tests later.

9. Click the **Save** button to save your build definition.



10. In the **Save** dialog, type **CI Build** for the **Name**
11. In the **Comment** field, enter **Initial build definition for continuous integration**.



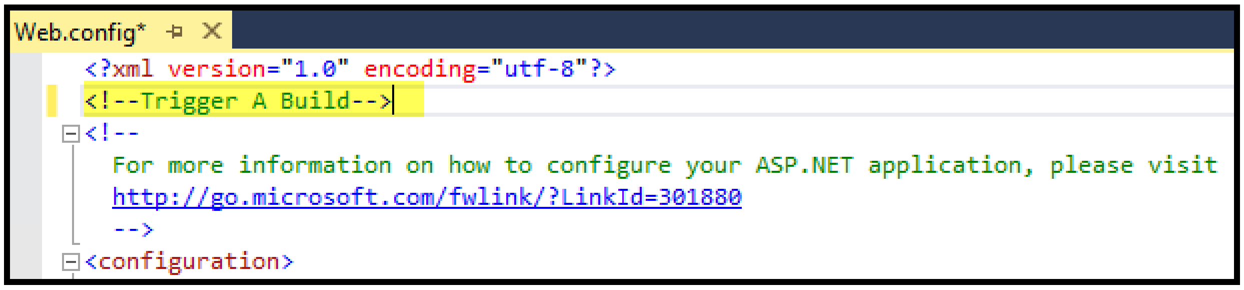
Save

Name: CI Build

Comment: Initial build definition for continuous integration.

Select folder: \

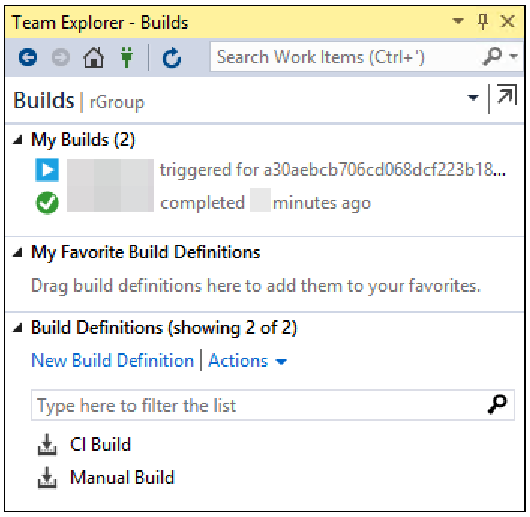
12. Click **OK.**
13. Now you need to make a change in your solution, commit and push it, and see if the build runs. Return to **Visual Studio**.
14. In the **Rg.Web** project, open the **Web.Config** file.
15. At the top of the file, add the following comment:



```
<?xml version="1.0" encoding="utf-8"?>
<!--Trigger A Build-->
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=301880
  -->
<configuration>
```

16. Save your changes.
17. In **Team Explorer**, click the **Changes** button.
18. Enter the following commit message: **Made a quick change to trigger a CI Build**.
19. Click the **Commit All and Push** button. If prompted to save changes, click **Yes**.
20. Click the **Home** button in Team Explorer.

21. Click the **Builds** button on the **Home** screen.



Team Explorer - Builds

Search Work Items (Ctrl+')

Builds | rGroup

My Builds (2)

[redacted] triggered for a30aebcb706cd068dcf223b18...
completed [redacted] minutes ago

My Favorite Build Definitions

Drag build definitions here to add them to your favorites.

Build Definitions (showing 2 of 2)

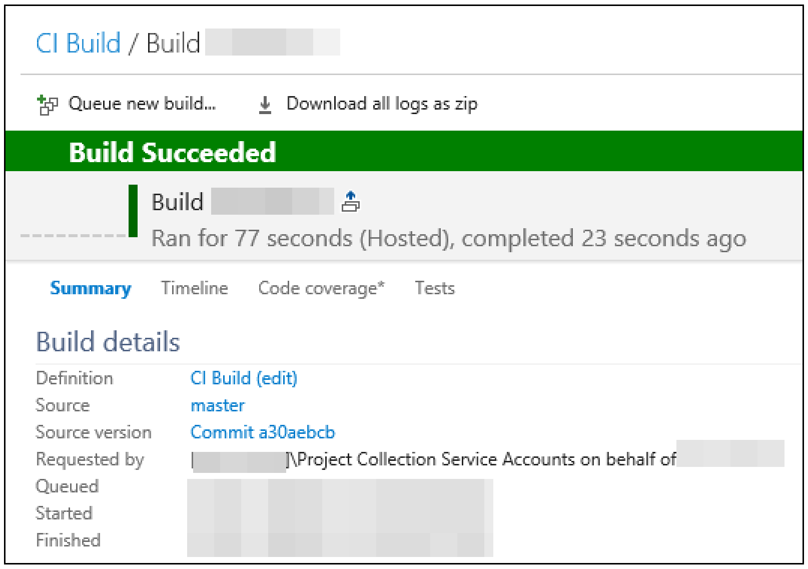
New Build Definition | Actions

Type here to filter the list

CI Build
Manual Build

22. You'll see a build has been initiated. Double-click the build to open it in the VSTS web portal and watch it run.

23. As before, once the build finishes running, click the build name to view the report.



CI Build / Build [redacted]

Queue new build... Download all logs as zip

Build Succeeded

Build [redacted]
Ran for 77 seconds (Hosted), completed 23 seconds ago

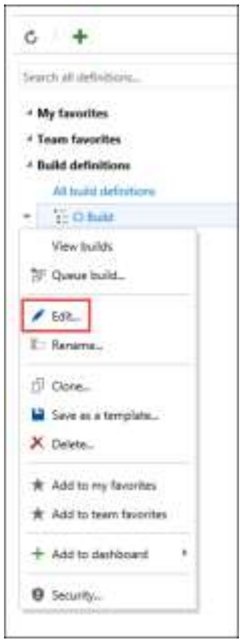
Summary Timeline Code coverage* Tests

Build details

Definition	CI Build (edit)
Source	master
Source version	Commit a30aebcb
Requested by	[[redacted]]\Project Collection Service Accounts on behalf of [redacted]
Queued	[redacted]
Started	
Finished	

24. For now, you can disable the CI Build so it doesn't use all your build minutes up during the lab. Click the **Explorer** link again.

25. Select the arrow next to **CI Build** and select **Edit** from the menu.



26. Click on the **Triggers** tab. Uncheck the **Continuous Integration (CI)** checkbox.
27. Click the **Save** button.
28. In the **Comment** field, enter **Disabled CI**.
29. Click **OK.**
30. You're now ready to continue working on the next walkthrough.

Last Updated: September 23, 2016.