

Developer Network

Azure App Service Team Blog

How the App Service team functions

Deploying Visual Studio 2017 Function Projects with VSTS

★★★★★

June 1, 2017 by [Donna Malayeri](#) // [21 Comments](#)

Share 47

179

35

With the new [Visual Studio 2017 support for Azure Functions](#), you can now author functions using C# class libraries. With the new project type, triggers and bindings are defined using attributes, which are then converted to function.json as a build task.

To build the project on the server with continuous integration, you have two options: 1) the [Continuous Integration](#) feature of Functions, or 2) [Visual Studio Team Services](#) (VSTS). The code can be hosted on VSTS or an external service such as GitHub or Bitbucket.

The process is quite easy, thanks to a new build template: ASP.NET Core on .NET Framework. If you're not familiar with VSTS build definitions, read [CI/CD for newbies](#).

To create a build definition, do the following:

1. From the Build Definitions view in VSTS, select **+New**.
2. Choose the template **NET Core (.NET Framework)**. Even though we're not deploying an ASP.NET Core app, this template has the correctly configured tasks for a Functions project.
3. Add a build task for **Azure App Service Deploy**.
4. Choose an Azure subscription and select your Function App under **App Service name**.
5. Modify the **Package or folder** setting to use `$(build.artifactstagingdirectory)/**/*.zip`
6. Save and queue the build.

Here's an animated GIF that walks through the VSTS configuration steps: