# Continuous Deployment With Jenkins And .NET

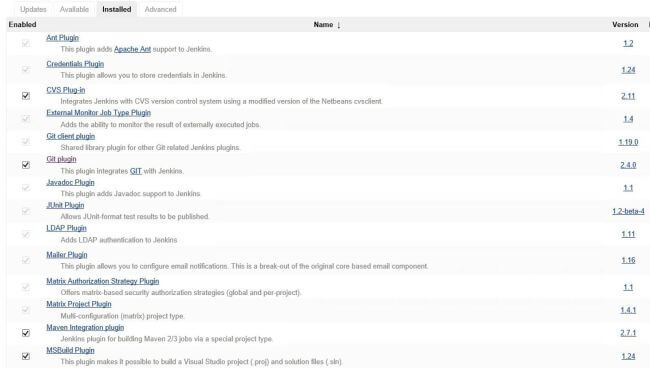
* Platform: .NET 4.5.2
* IDE: Visual Studio 2015 (MSBuild files)
* Application type: Windows Service
* NuGet: Package Manager used for all references.
* Source Control: Git (github.com)

**Configure Jenkins**

Jenkins has plugins for understanding Git repositories, MSBuild files, and various other technologies. In the current set-up, we only need to extend Jenkins with those two plugins.

**Install the Git Plugin for Jenkins**

1. Open Jenkins, http://localhost:8080/.
2. Navigate to "Manage Jenkins".
3. Navigate to "Manage Plugins".
4. Direct link:http://localhost:8080/pluginManager.
5. Use the "filter" box to search for "Git plugin" and install the plugin.



**Install the MSBuild Plugin for Jenkins**

* Open Jenkins, http://localhost:8080/.
* Navigate to "Manage Jenkins".
* Navigate to "Manage Plugins".
* Direct link: http://localhost:8080/pluginManager.
* Use the "filter" box to search for "MSBUILD plugin" and install the plugin.

Jenkins now understands MSBuild files and Git source control but we still need to configure the MSBuild Plugin with a path to the msbuild.exe that we would like to use.

**MSBuild configuration**

When the MSBuild plugin was installed, it added its own configuration options to the Jenkins global configuration page.

1. Navigate to <http://localhost:8080/.>
2. Click "Manage Jenkins".
3. Click "Configure System".
4. Scroll down the list until you find "MSBuild".
5. Click the "MSBuild installations..." button.
6. Click “Add MSBuild”.
7. Give the new MSBuild configuration a name like "MSBuild-default".
8. In the path field, insert the fully qualified path to exe.
9. On my server, the path is C:\Program Files.

(x86)\MSBuild\14.0\Bin\msbuild.exe, but this can be different on your system.

1. Read below for more information about MSBuild and how to install it.
2. Click "Save".



**MSBuild installation**

MSBuild is installed with Visual Studio, it’s the build system that Visual Studio uses when you select "build" or hit "F5".

It’s not always feasible or even possible to install Visual Studio on your build machine.

This could be due to license and security issues, etc.

To accommodate this Microsoft has released a separate package called: "Microsoft Build Tools 2015" that contains all you need for using MSBuild.

Direct download

https://www.microsoft.com/en-us/download/details.aspx?id=48159

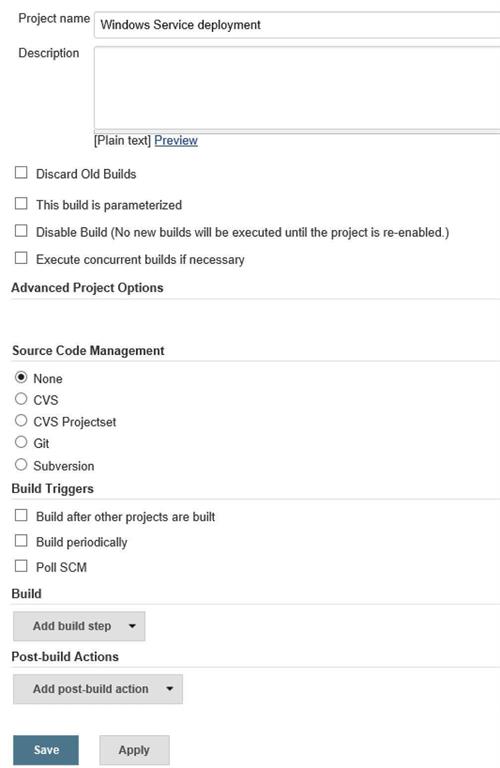
After successful installation, you have MSBuild available on the build server and with that, you get the path value for step 8, above.

With this step done Jenkins is ready to build and deploy with MSBuild and Git.

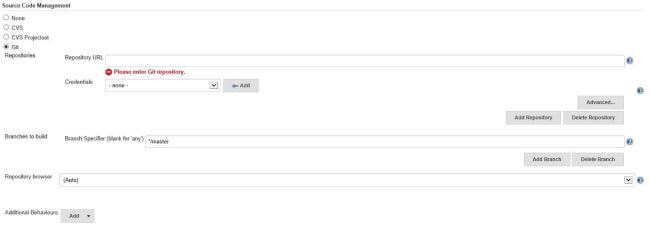
**Create a new Jenkins build project**

It’s now time to point Jenkins to the source code and start building.

* Open Jenkins, http://localhost:8080/.
* Select "new".
* Direct link http://localhost:8080/view/All/newJob.
* Give the project the name "Windows Service Deployment" or something you can remember.
* Select “Freestyle project”.
* Select “Ok”.



* Next, expand the “Source Code Management” region by selecting “Git”.



Complete the Git configuration by filling in the blanks with a URL to your repository and optional credentials (if needed).

Jenkins can also work with branches. In this set-up, I will leave the branch as the default (Master) but you can select whatever fits your needs.

If you don't already have a Git repository ready for testing, you can use the pre-cooked repository here:

https://github.com/martinesmann/jenkins-ci-template

The same was built and deployed in our case.

If you haven’t done so already, click “Save” to persist your changes and navigate back to the main "Project" page.

**Testing the Git configuration**

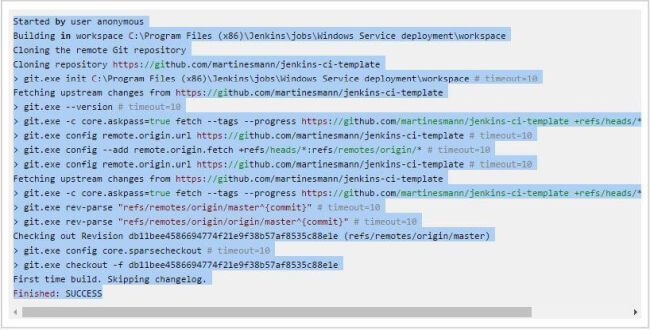
Now we can test if the Git Source Management tab has been configured correctly and if we can clone the source code.

We are not yet building anything, only cloning the source. Building part will come in a moment. First, let's make sure Jenkins can clone the source from the repository.

* Navigate to the "Project" page.
* Click the “Build Now” to start a "Build".



* In the "Build History" region you should now see a build in progress with the name #1".
* If everything completes as expected, then “SUCCESS” and the bubble maker stays blue, if not it turns red.
* To see the build details, click the build number "#1".
* Then, click "Console Output". You should see something similar to what I have here.



**\*\*\*What’s important to note here is the pathto the "workspace" as this is where the source code is downloaded to and built from. This knowledge can be very helpful when debugging the CI setup.\*\*\***

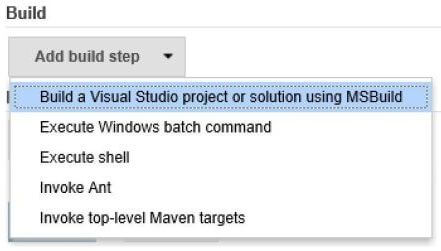
**Building the source**

The next step is to compile and build the source code.

Navigate to the "project" page

Select “Configure”

Find the "Add build step"



Select the "Build a Visual Studio project or solution using MSBuild".

We need to configure a few values here:

* First, select the MSBuild version (we configured this in a previous step).
* Then give the path to the \*.sln or \*.proj file for your project.

For the pre-cooked repository, the path is src\MyWindowsService\MyWindowsService\Deploy-Windows-Service-Via-MSBuild.proj

Please note We are not pointing to the solution file, rather we are pointing to a custom MSBuild file that is in the project. This MSBuild file is handling all steps involved with compiling and deploying the Windows Service.

You can read more about this MSBuild file from here: http://mrchief.calepin.co/deploying-windows-service-via-msbuild

If you are using the pre-cooked repository, your setup should look like this,



Click Save.

**NuGet package restore**

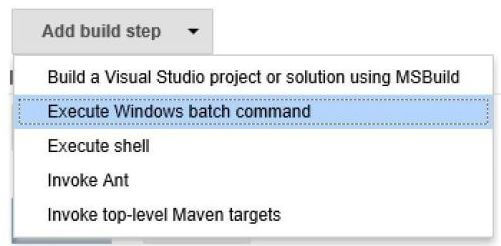
If we were to build the project right now it would fail due to missing NuGet packages. Therefore we need to restore the NuGet packages before attempting to build the source.

nuget.exemakes this task very easy, we simply need to fire this command on the solution file:

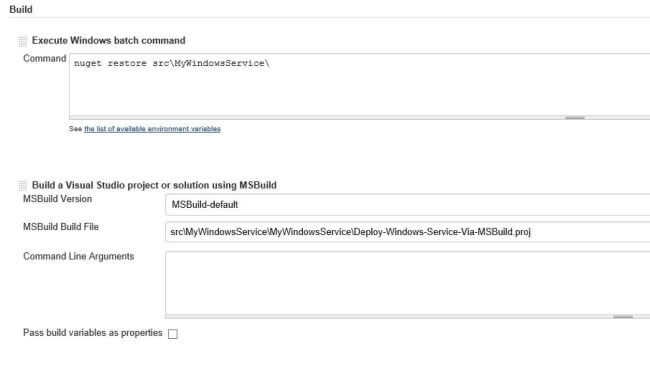
nuget restore "path to \*.sln" file

Jenkins can handle multiple build steps, so let's add a build step to enable NuGet to restore.

1. Navigate to the "project" page.
2. Select "Configure".
3. Find the "Add build step".
4. Click "Add build step".



1. Select "Execute Windows batch command".
2. Re-arrange the build order by dragging the new build step to the top.



1. In the Command field insert,

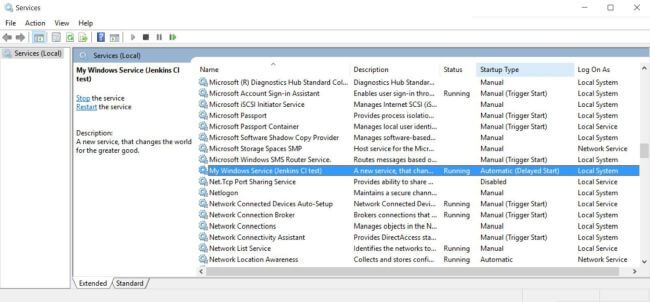
nuget restore src\MyWindowsService\

1. Click "Save".

We are now ready to build the project!

* Navigate to the project page
* Click the "Build now" link
* Wait for the build to complete
* Check that the Service is installed and running,

* 1. On Windows open the "Services" management window
  2. Scroll down to you find the service "My Windows Service"



You have now successfully configured Jenkins to download, compile, and deploy a .NET Windows Service