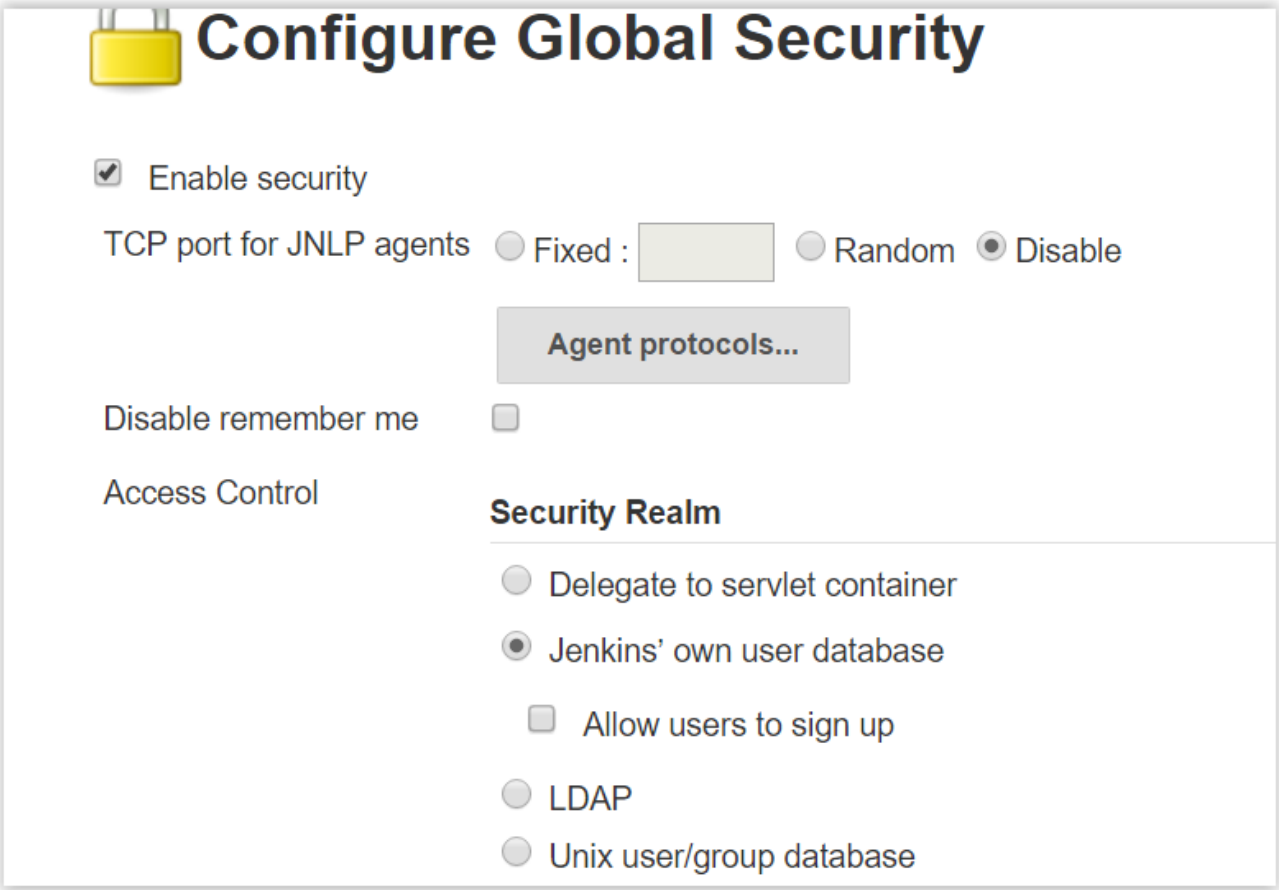
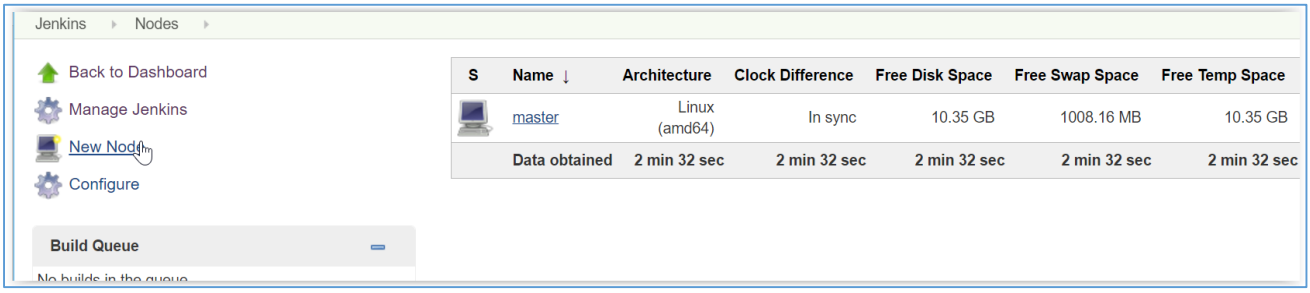
Exercise 3 Jenkins - Quick Guide

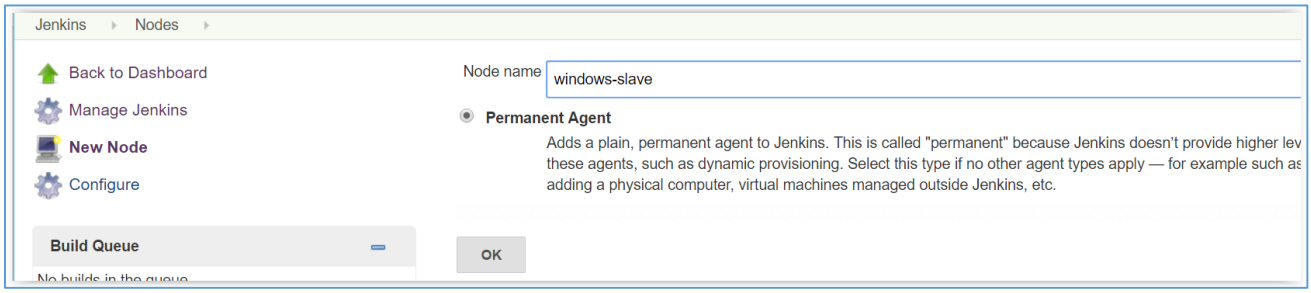
1. Add windows agent to Jenkins like in the Demo and install it as a service:
2. Go to windows and make a new folder c:\jenkins
3. Also need to install git in windows
4. Install jre : http://download.oracle.com/otn-pub/java/jdk/8u121-b13/e9e7ea248e2c4826b92b3f075a80e441/jre-8u121-windows-x64.exe
5. Now open Jenkins window
6. Go to manage Jenkins
7. Go to configure global security



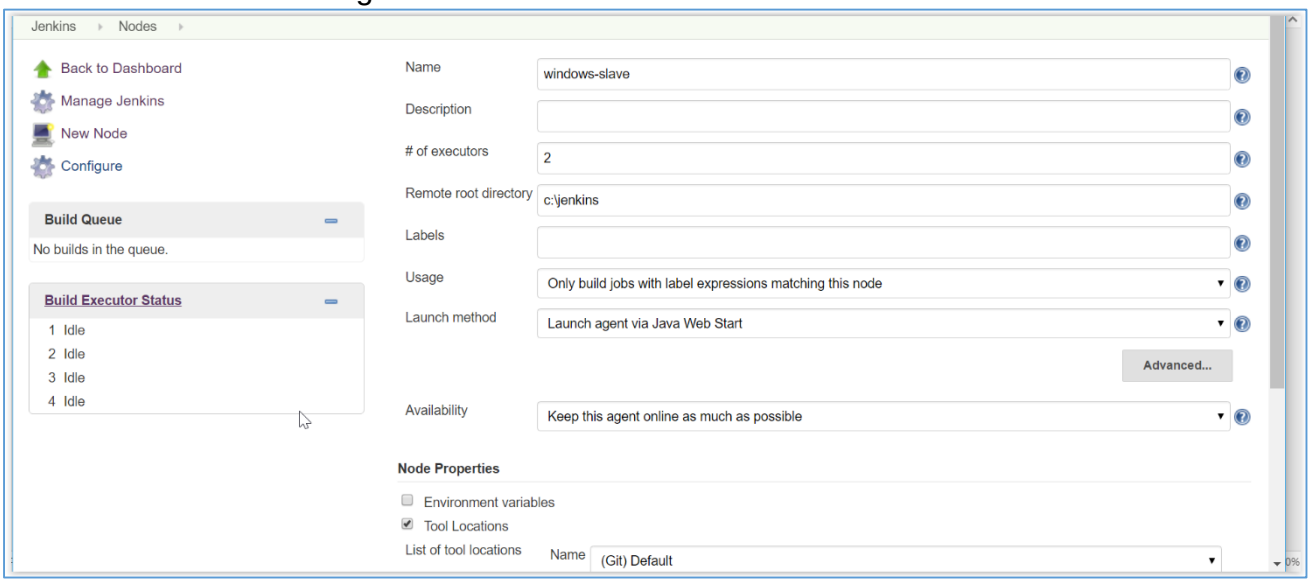
8. Change TCP port for JNLP agents to random
9. And click save
10. Go to manage nodes



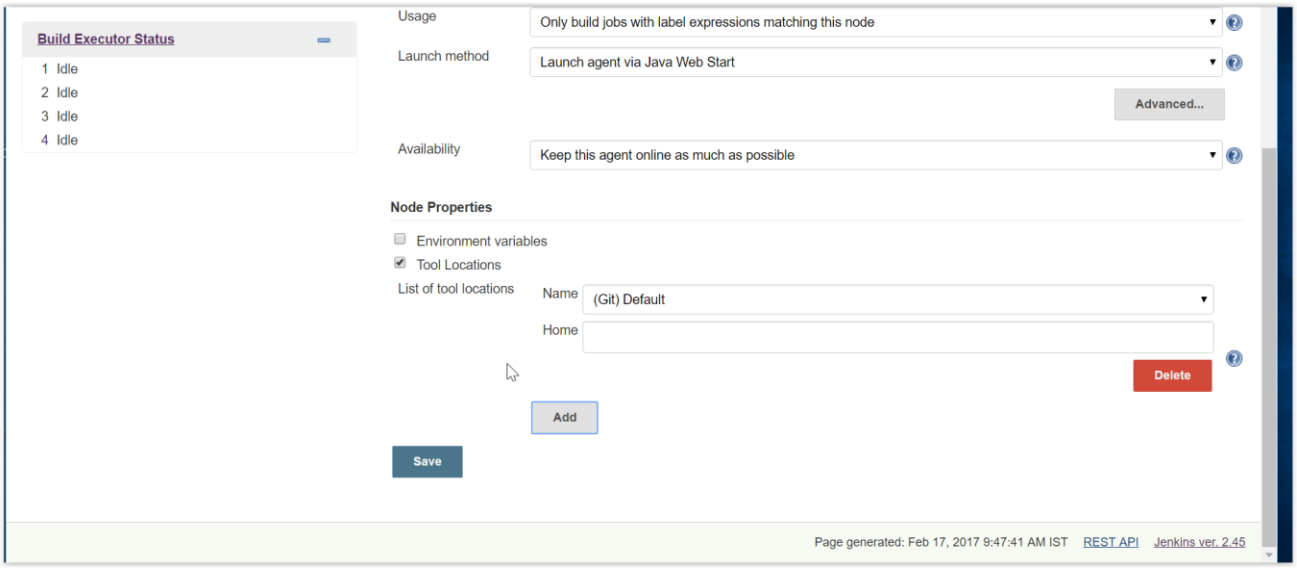
11. Click new node
12. Give it a name and mark permanent agent



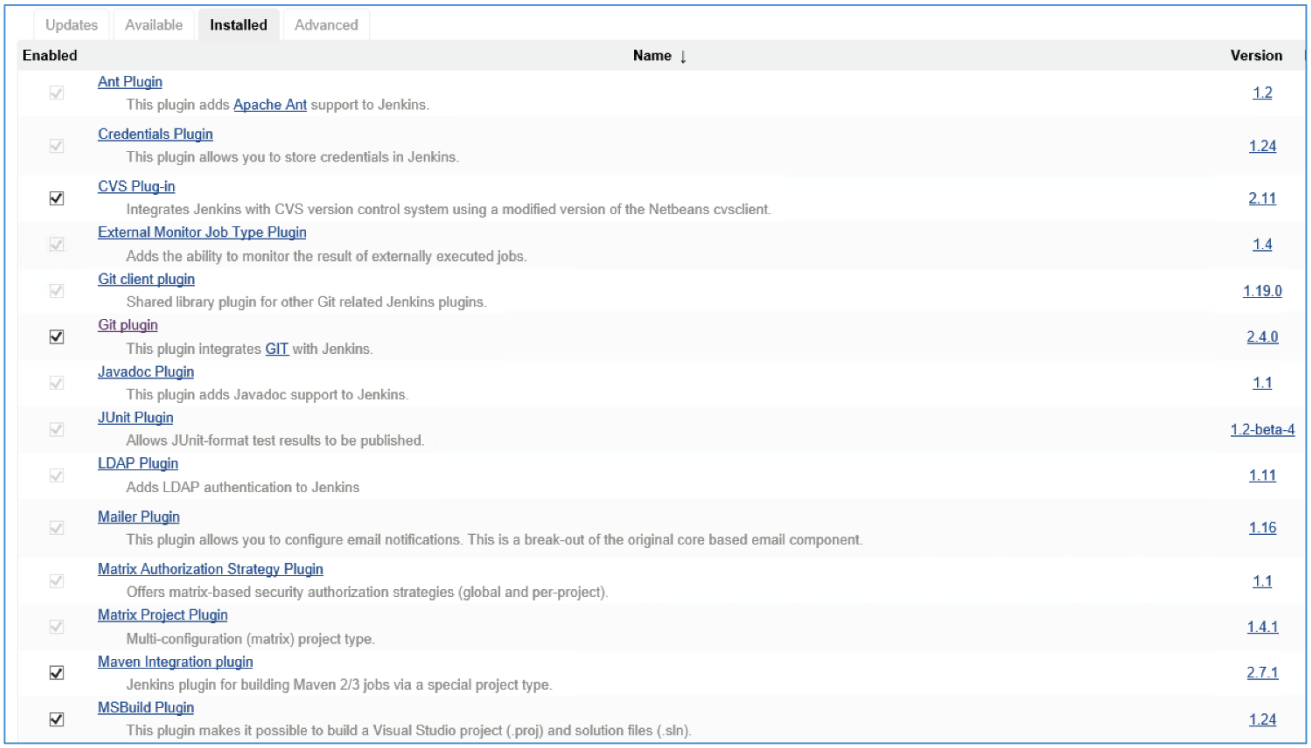
13. Make sure the settings as follow:



14. Insert the path to git bin folder



15. And click save
16. Now click on the new node and launch agent
17. Now in the small window where it says connected: go file and click install as a service
18. Open Jenkins in your browser
19. Navigate to “Manage Jenkins”
20. Navigate to “Manage Plugins”
21. Install the MSBuild Plugin for Jenkins:



22. Jenkins now understands MSBuild files

23. but we still need to configure the MSBuild Plugin with a path to the msbuild.exe 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. Click “Manage Jenkins”
2. Click “Configure System”
3. Scroll down the list until you find “MSBuild”
4. Click the “MSBuild installations…” button
5. Click “Add MSBuild”
6. Give the new MSBuild configuration a name like “MSBuild-default”.
7. In the path field, insert the fully qualified path to msbuild.exe
   On my server the path is: C:Program Files (x86)MSBuild14.0Binmsbuild.exe, but this can be different on your system.
8. 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.

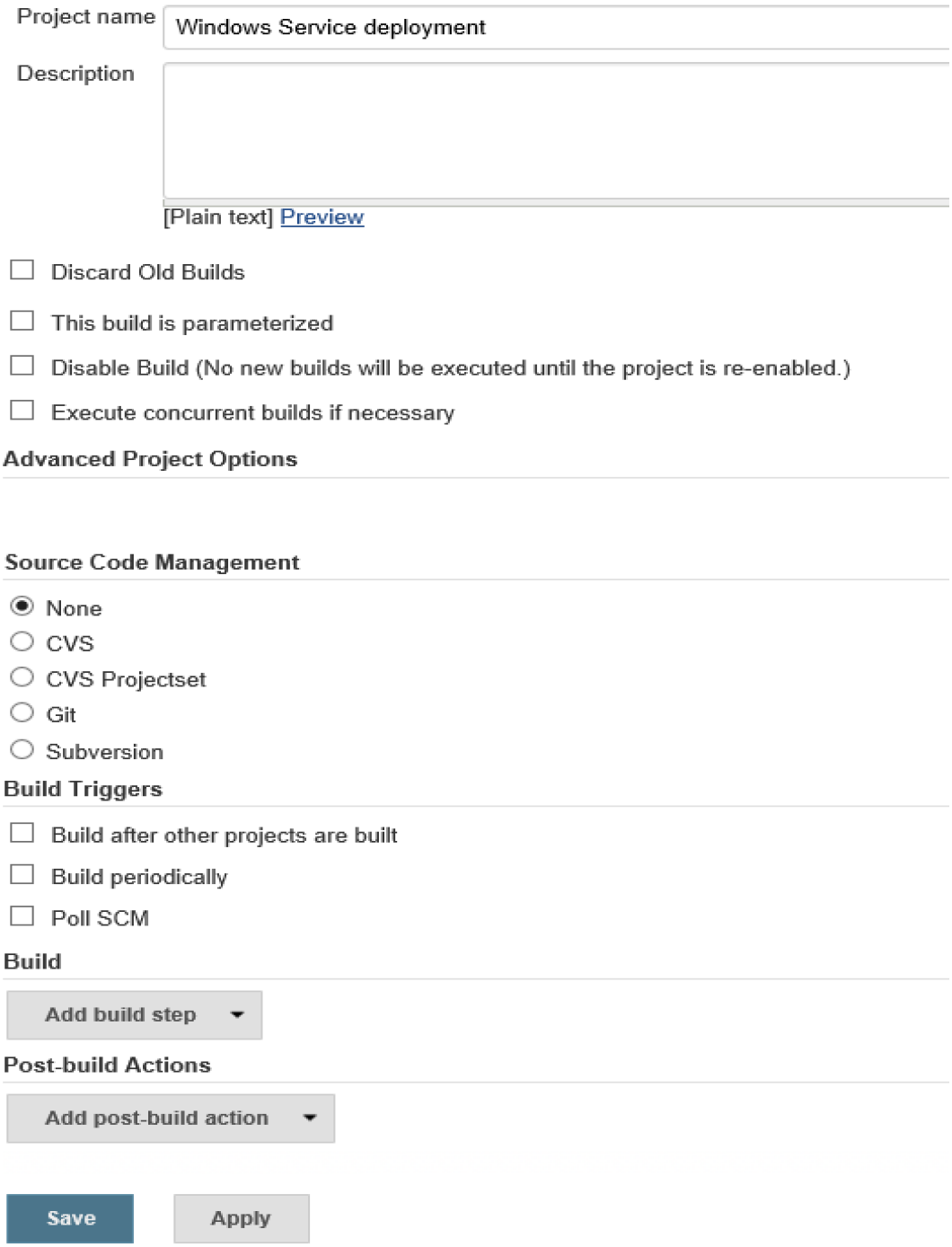
To accommodate this Microsoft has released a separate package called: "Microsoft Build Tools 2015" that contains all you need for using MSBuild.

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



1. Select “new” job
2. Select “Freestyle project”.

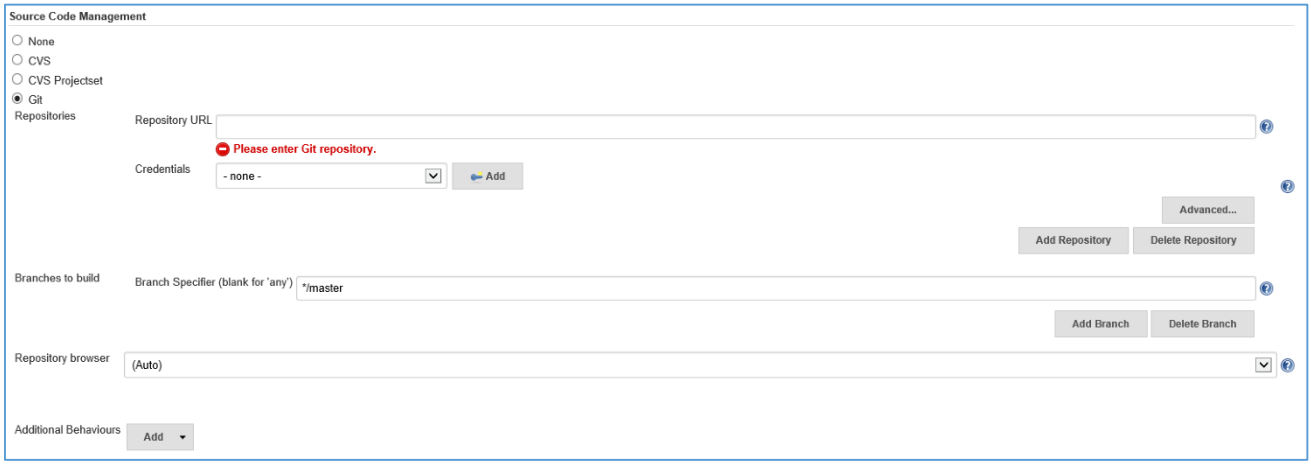
In the advanced project option click on run only on specific node label : windows (or your windows agent slave name label)

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

Fork my github repo:

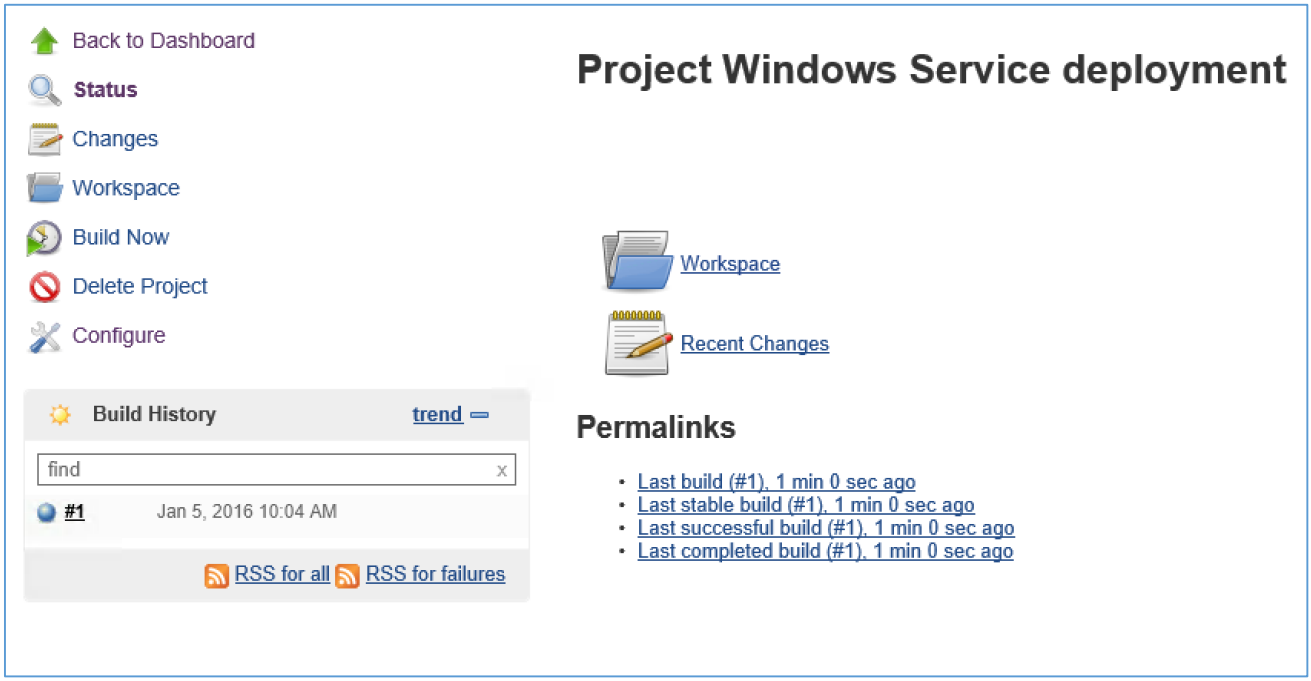
https://github.com/Lidorlg/jenkins-ci-template.git

and use your own url in git:



Testing the Git configuration

1. Navigate to the “Project” page
2. Click the “Build Now” to start a “Build”.



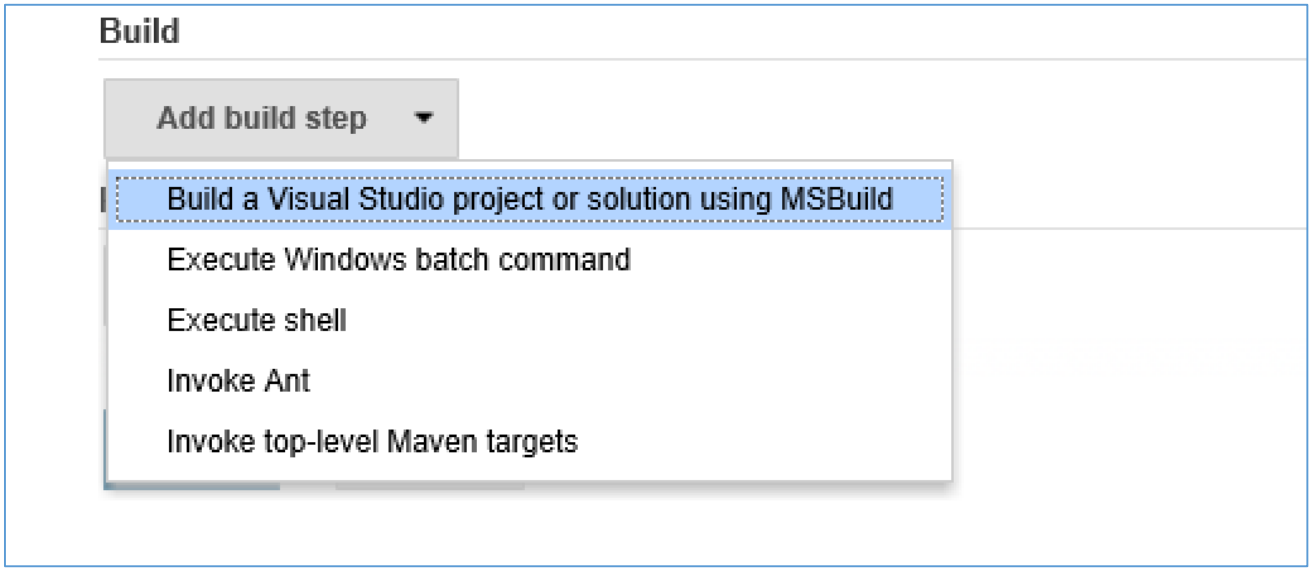
3. In the “Build History” region you should now see a build in progress with the name “#1”.

4. If every completes as expected “SUCCESS” then the bubble maker stays blue, if not it goes red.

Building the source

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

1. Navigate to the “project” page
2. Select “Configure”
3. Find the “Add build step”

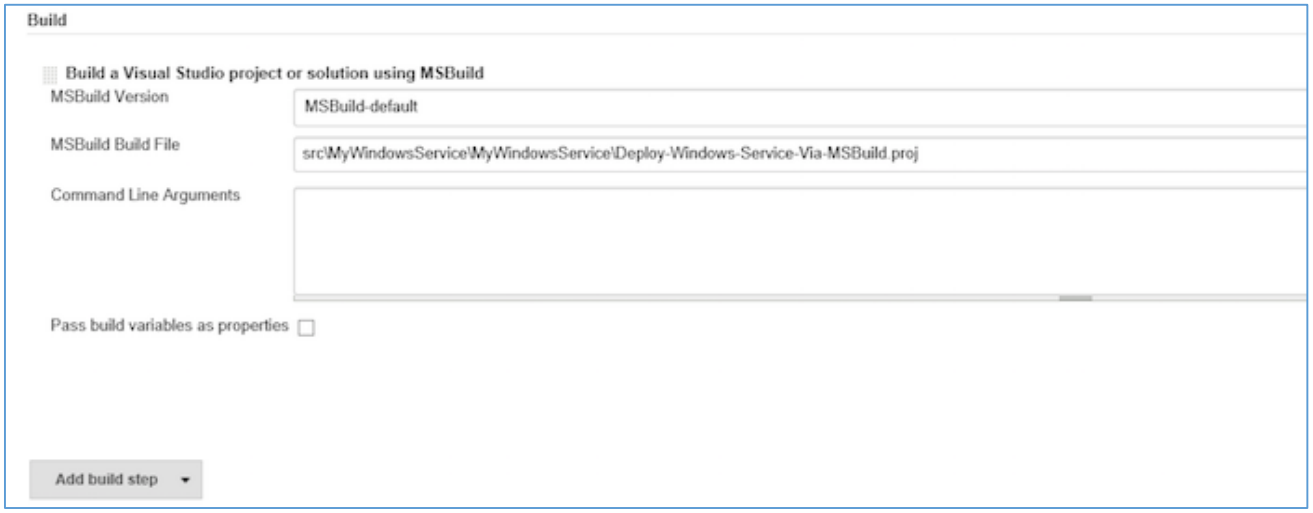


Select the “Build a Visual Studio project or solution using MSBuild”.

We need to configure a few values here:

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

   For the pre-cooked repository the path is: srcMyWindowsServiceMyWindowsServiceDeploy-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.*
   If this step not working -> point to sln file

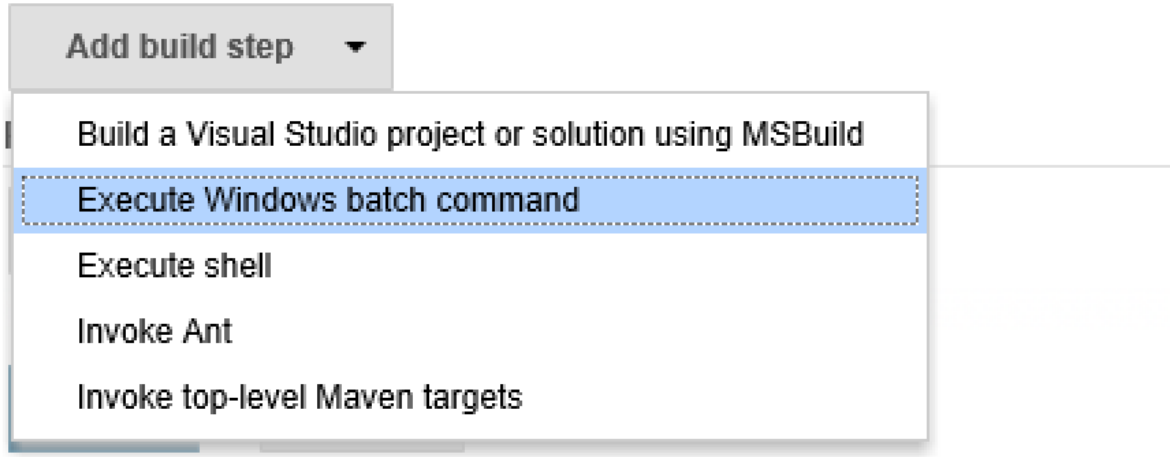


NuGet package restore

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.

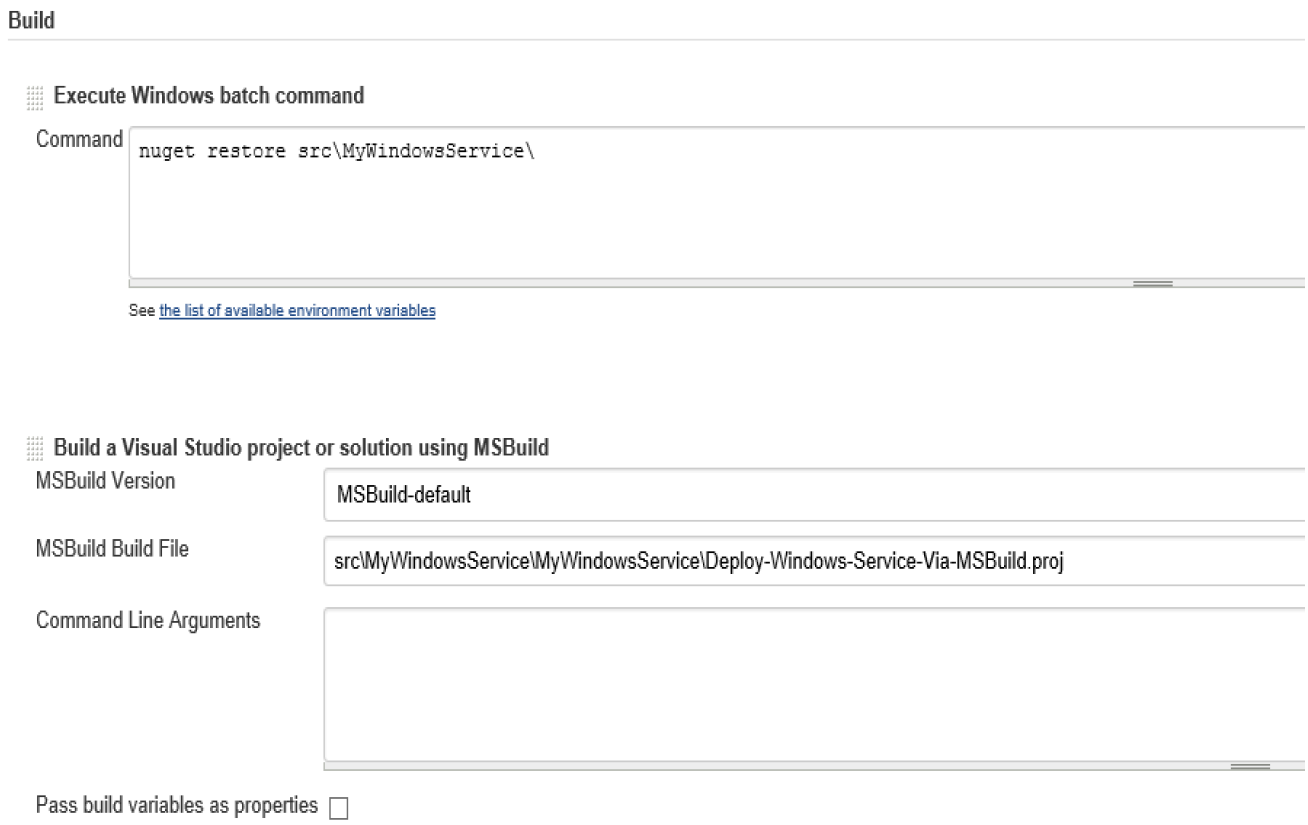
nuget.exe makes this task very easy, we simply need to fire this command on the solution file:
nuget restore "path to *.sln" file
No surprise Jenkins can handle multiple build steps, so let's add a build step to enable NuGet restore.

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



5. Select “Execute Windows batch command”.

6. Re-arrange the build order by dragging the new build step to the top.



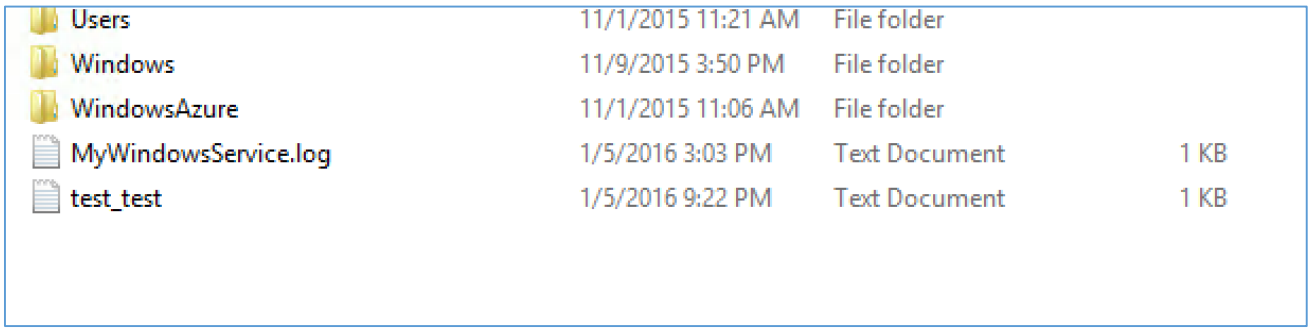
7. In the Command field insert:

   nuget restore src\MyWindowsService
8. Click “Save”.

We are now ready to build the project!

The final test! Primetime!

1. Navigate to the project page
2. Click the “Build now” link
3. Wait for the build to complete
4. Open “Windows Explorer” and navigate to C:
5. Confirm that the file “MyWindowsService.log” exists



Users	11/1/2015 11:21 AM	File folder	
Windows	11/9/2015 3:50 PM	File folder	
WindowsAzure	11/1/2015 11:06 AM	File folder	
MyWindowsService.log	1/5/2016 3:03 PM	Text Document	1 KB
test_test	1/5/2016 9:22 PM	Text Document	1 KB

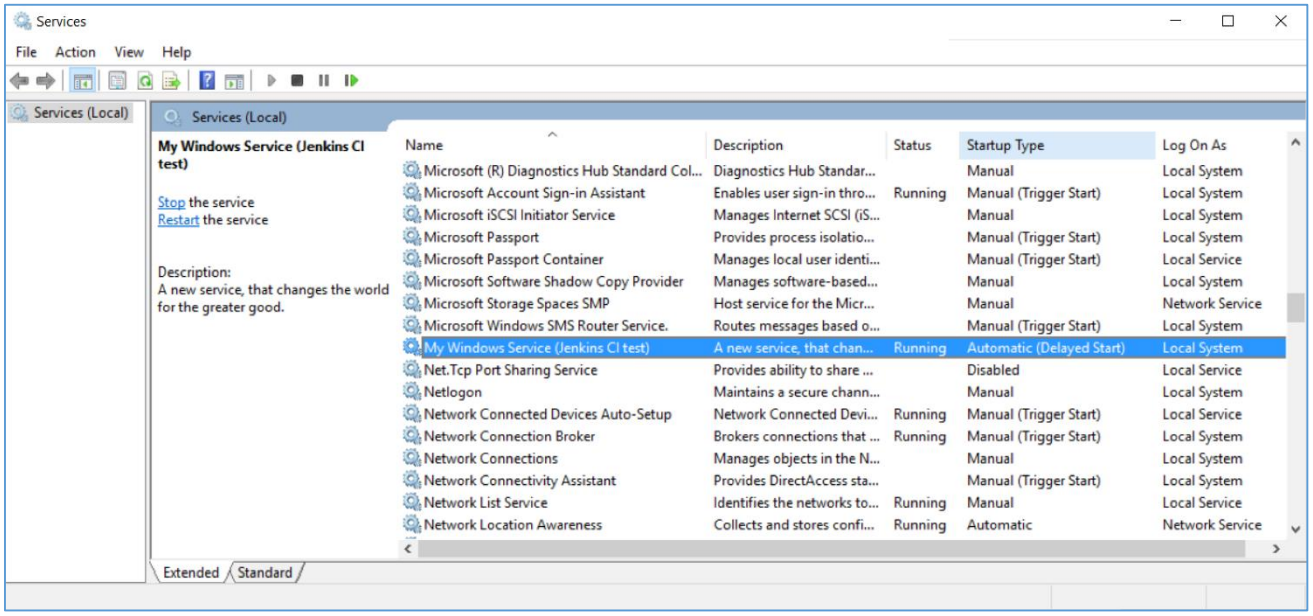
6. Open the file and read the log content.

   This file was create by our newly installed Windows Service!

7. Check that the Service is installed and running:

   A. On Windows open “Services” management window

   B. Scroll down to you find the service “My Windows Service (…)”



Congratulations! **You have now successfully configured Jenkins to download, compile and deploy a .NET Windows Service! Automation ROCKS!**