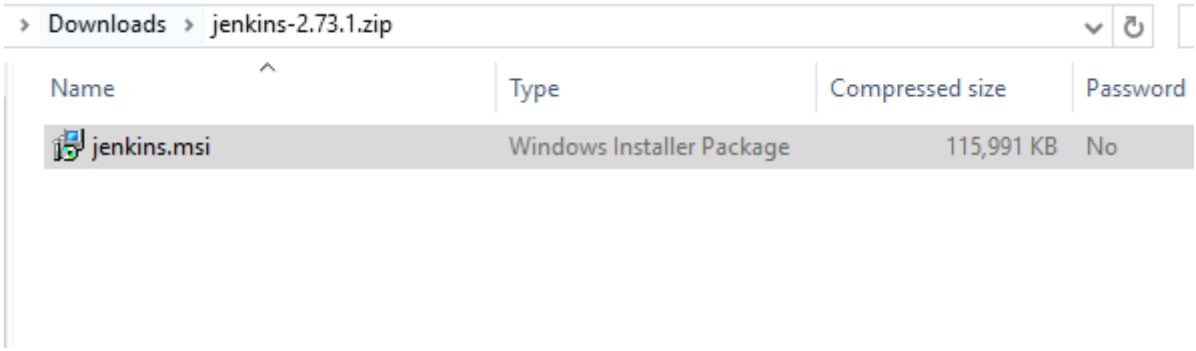
.Net 项目 Jenkins 持续集成实践

title: Continuous integration Guide

目录

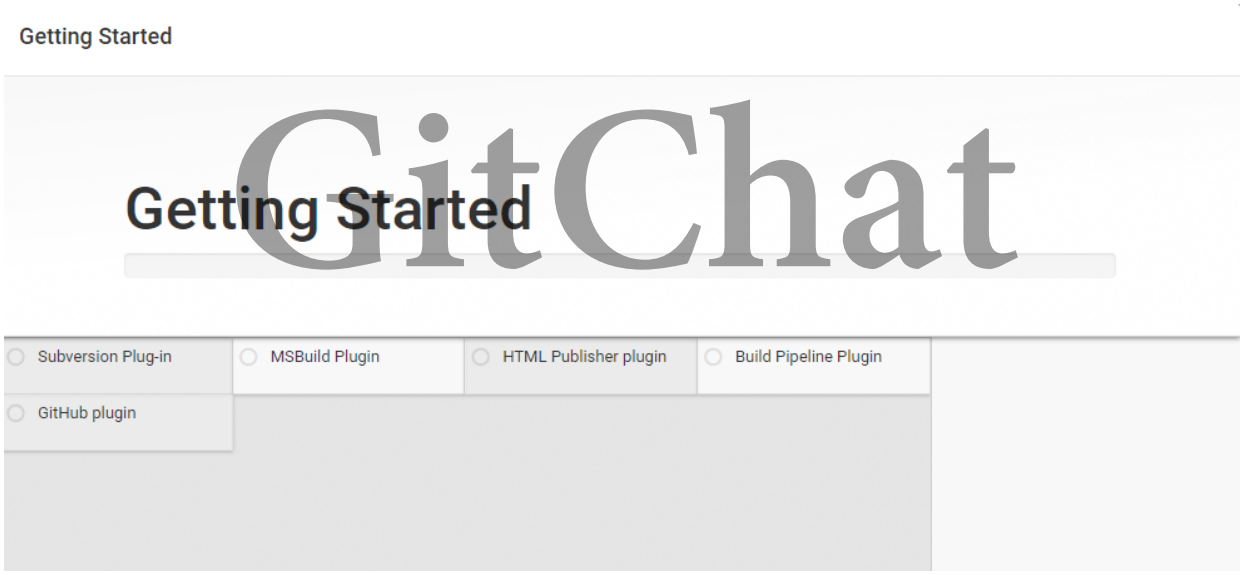
Download **Jenkins 2.73.1 for windows** from https://jenkins.io will automatically install Jenkins as window service.



当你解压下载的zip后，里面是一个名为jenkins.msi的可执行文件，双击安装，一路默认选项到最终完成。

此时浏览器会自动打开http://localhost:8080/。

会有一些交互的页面需要你去执行一些初始化的操作，特别的是在定制化插件的步骤，会让你自主选择需要安装的**部分插件**。勾选上一些.NET 项目所需的插件。



Note that the full list of plugins is not shown here. Additional plugins can be installed in the

- HTML Publisher plugin

Create Admin

上述部分插件安装完成后，界面会提示创建Admin账号。示例信息如下：

Username: admin

Password: admin

Fullname: admin

Email: liuning0820\@outlook.com

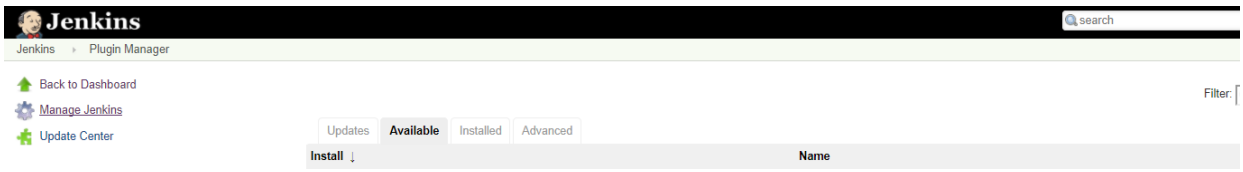
Install Git Tool

Download the git SCM tool from https://git-scm.com/download/win and install it on the Jenkins server.

插件管理

安装 MSTest Plug-In

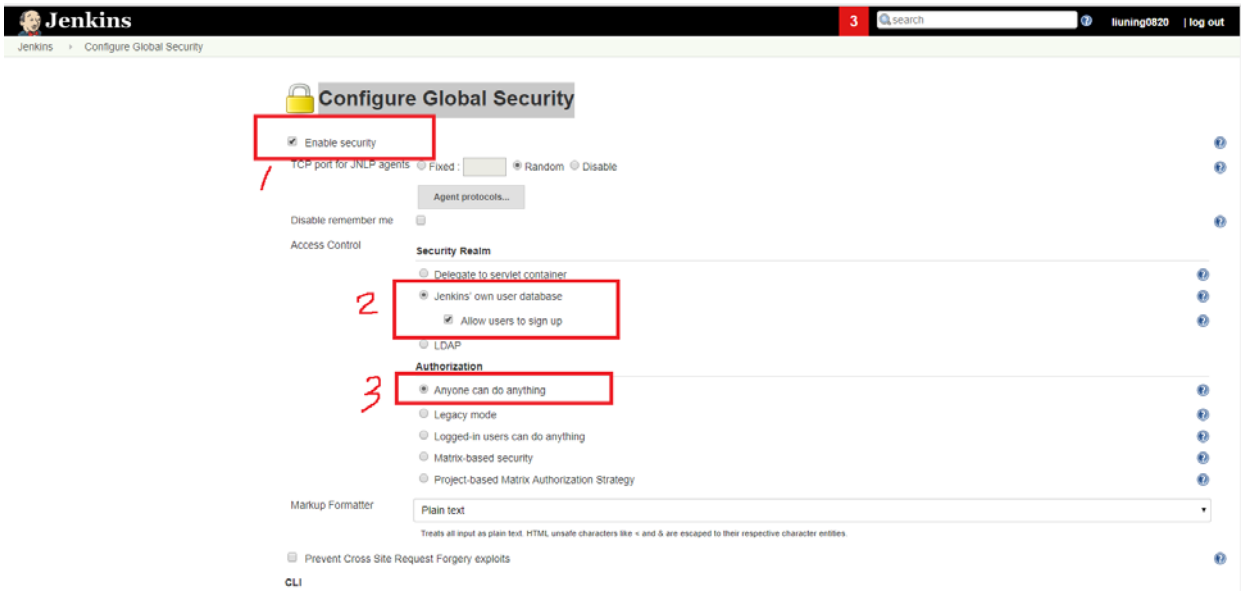
Click “Manage Jenkins” link, click “Manage Plugins” link, Click “Available” tab and type the name of the desired plugin in the filter text box. Install the “MSBuild Plugin” – This plugin converts MSTest TRX test reports into html.



权限管理SecurityManagement

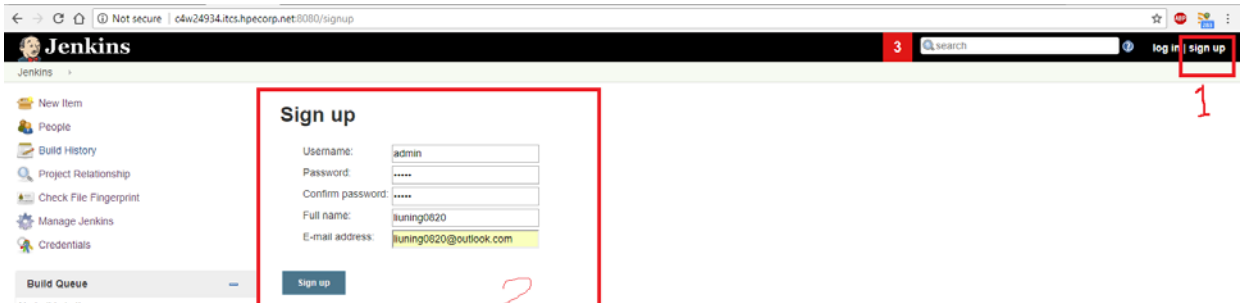
Enable Security

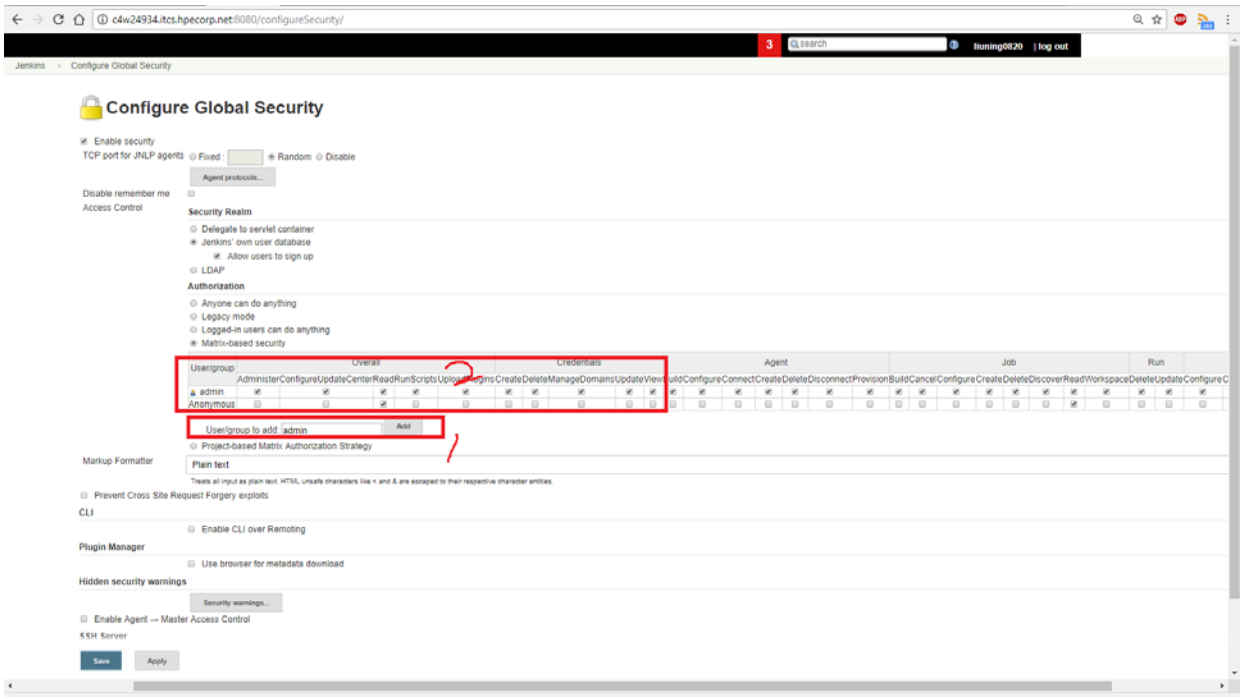
Manage Jenkins-> Configure Global Security



Note: In this step, we firstly set the authorization to “Anyone can do anything”. 因为到这一步，你还没有创建任何用户，需要去创建一些用户后，然后选择Matrix-based security。

Signup the “admin” user





持续集成步骤分解

Integrate Jenkins with GitHub

Install “GitHub plugin” if it is not installed yet:

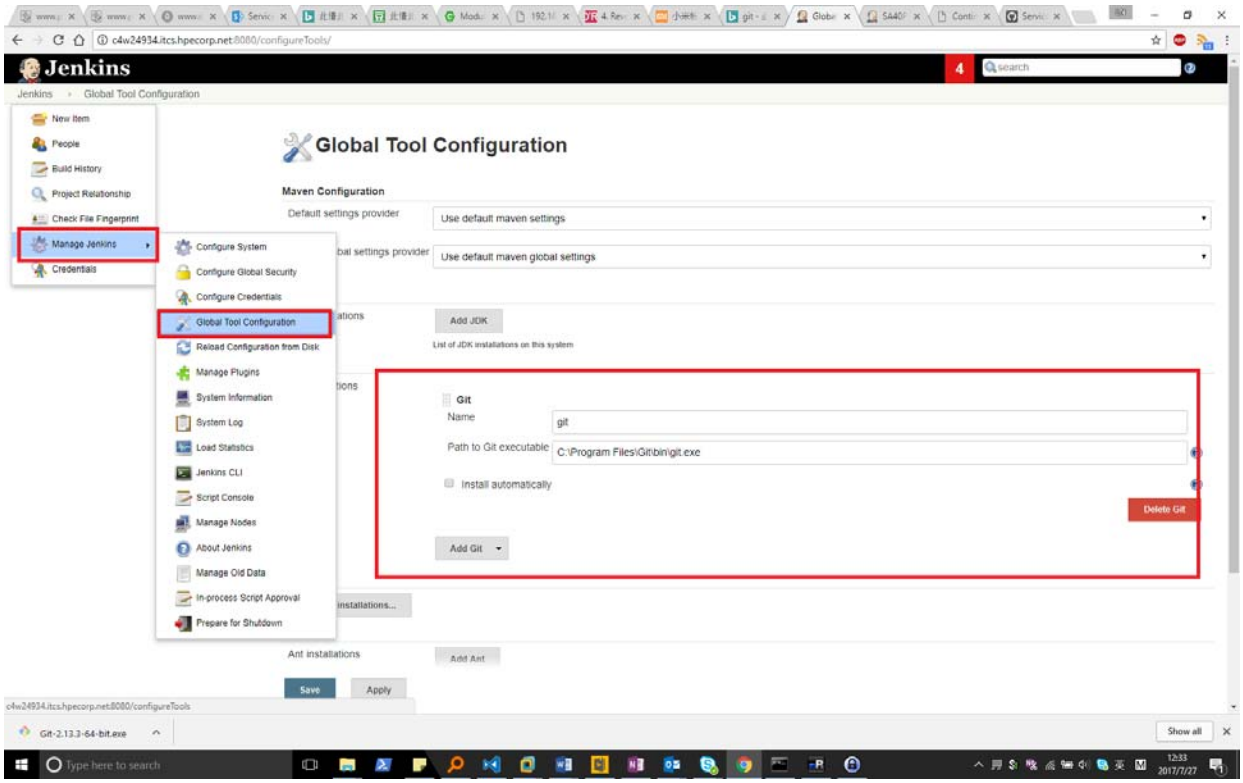
Open the Plugin Manager page.

Search “GitHub plugin” in the “Available” Tab and Install it.

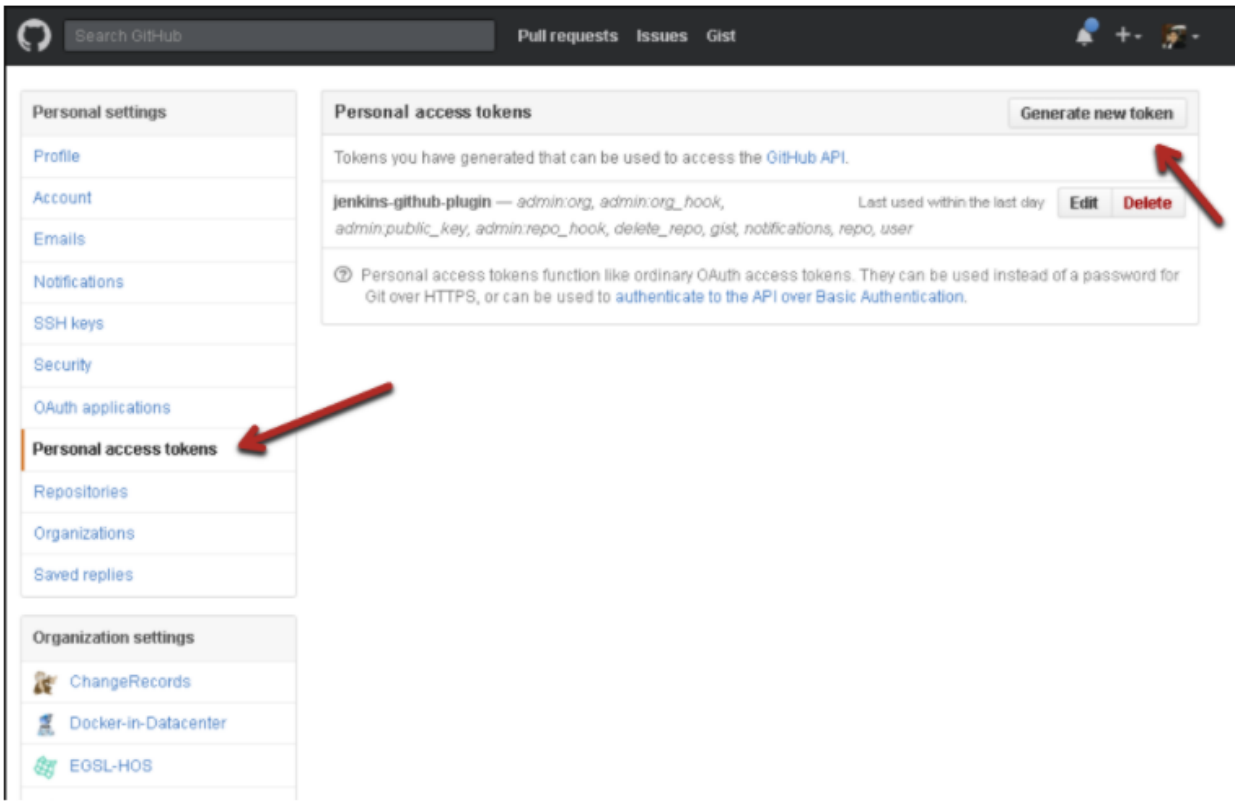
Install git scm tool in the Jenkin server:

Download git https://git-scm.com/ and install it in the Jenkins server, remember the installation path of git.exe. By default it is “C:\Program Files\Git\bin\git.exe”

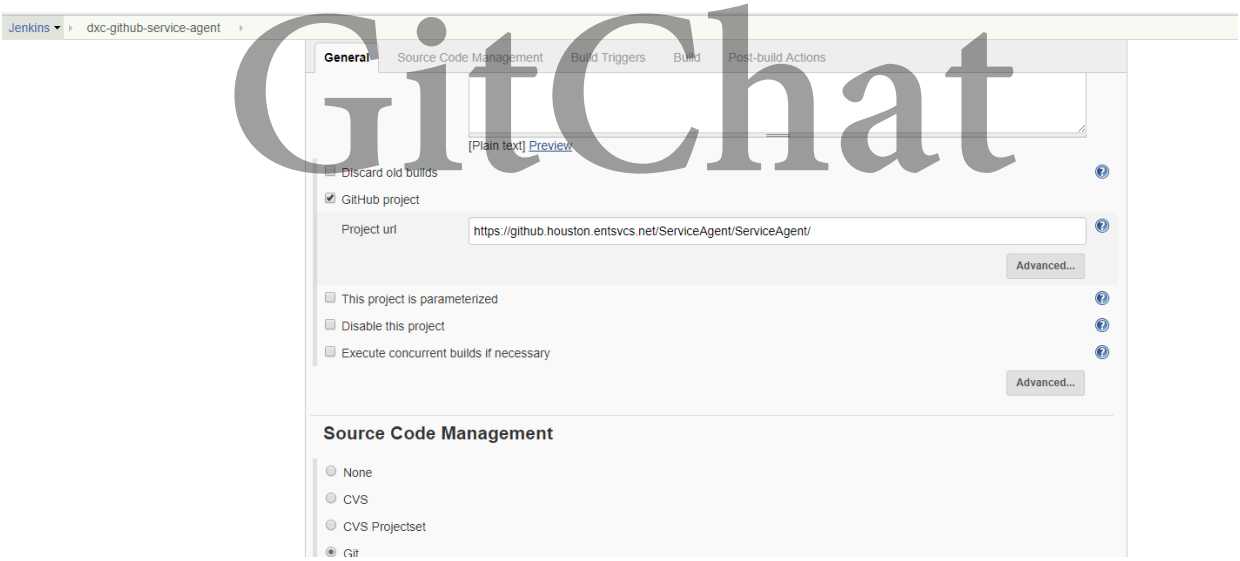
Then configure the git Path on Jenkins below



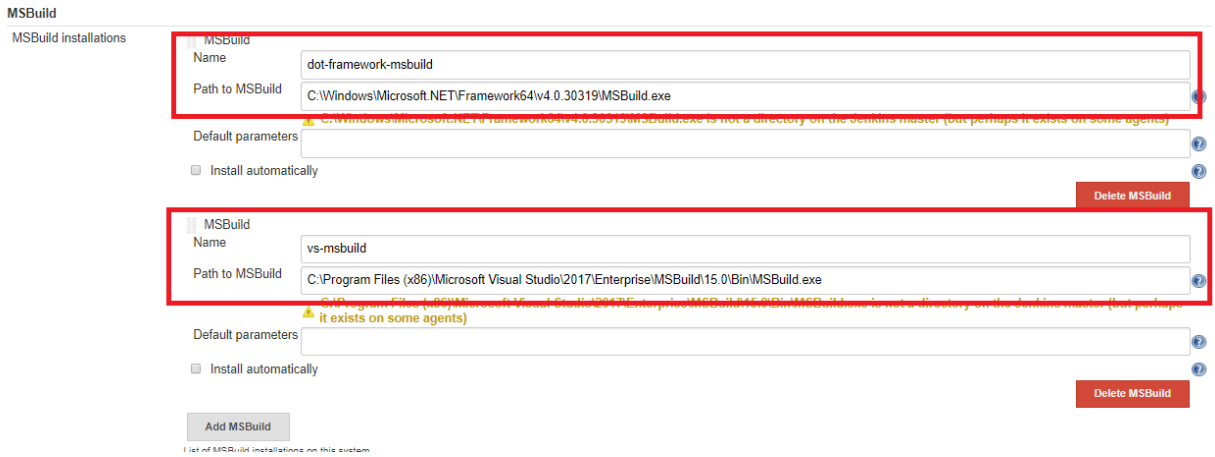
Add Credentials to Access GitHub



Then create a job the fetch the source code.



And configure the path of the MSBuild.



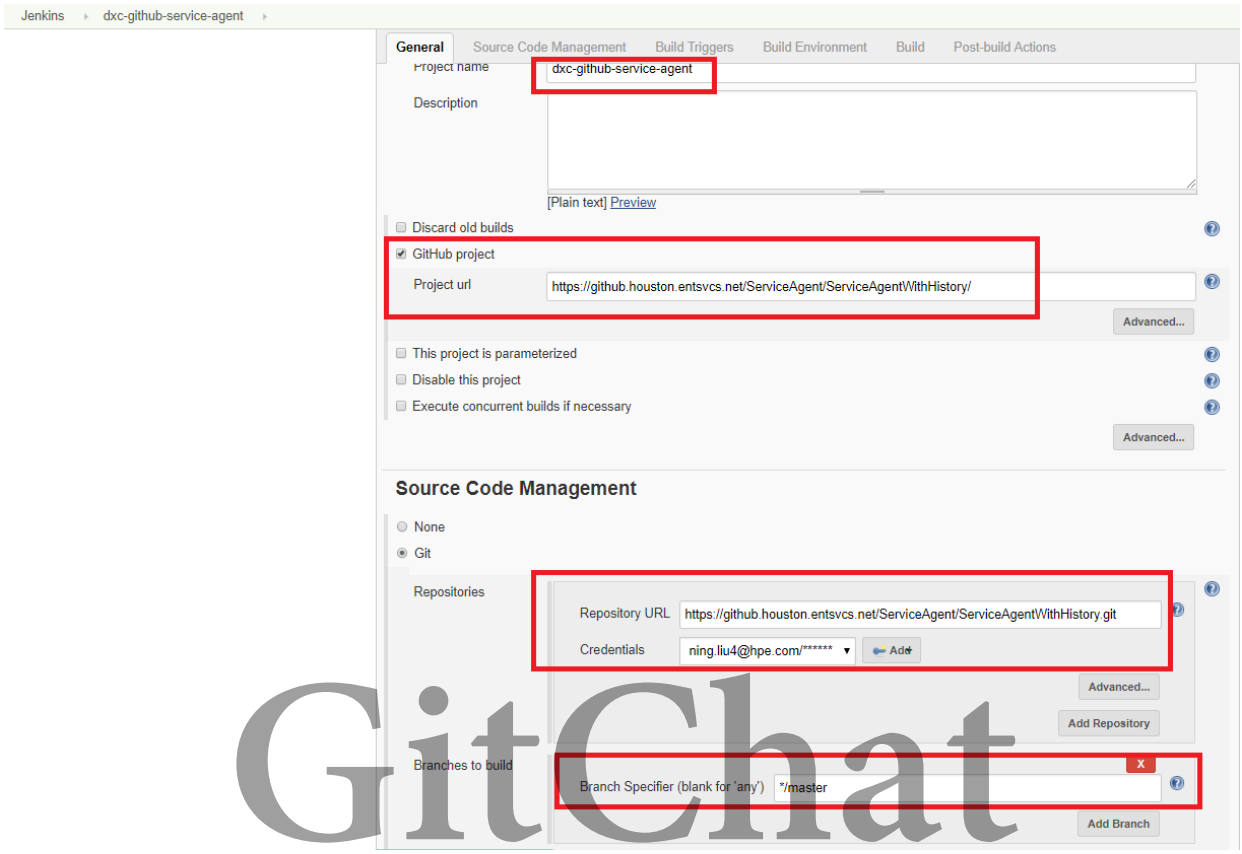
Configure Email Notification

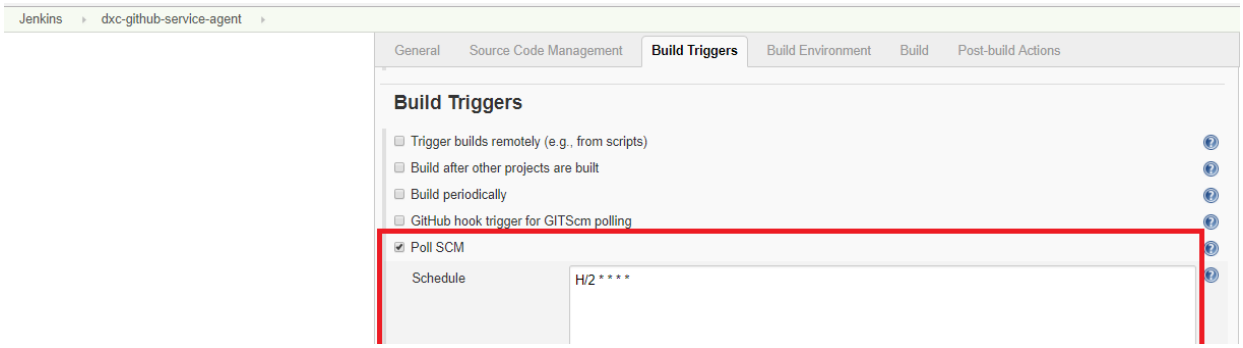
- Browse to the Jenkins homepage e.g. http://d4w0335g.houston.hp.com:8080/ (8080 is the default port)
- Navigate to Manage Jenkins\Configure System
- In the E-mail Notification section, Enter:

端到端持续集成演练

Create a job to build ASP.NET MVC Project

The job in below screenshot will pull source code from GitHub every two minutes and do the source
code compile using msbuild.

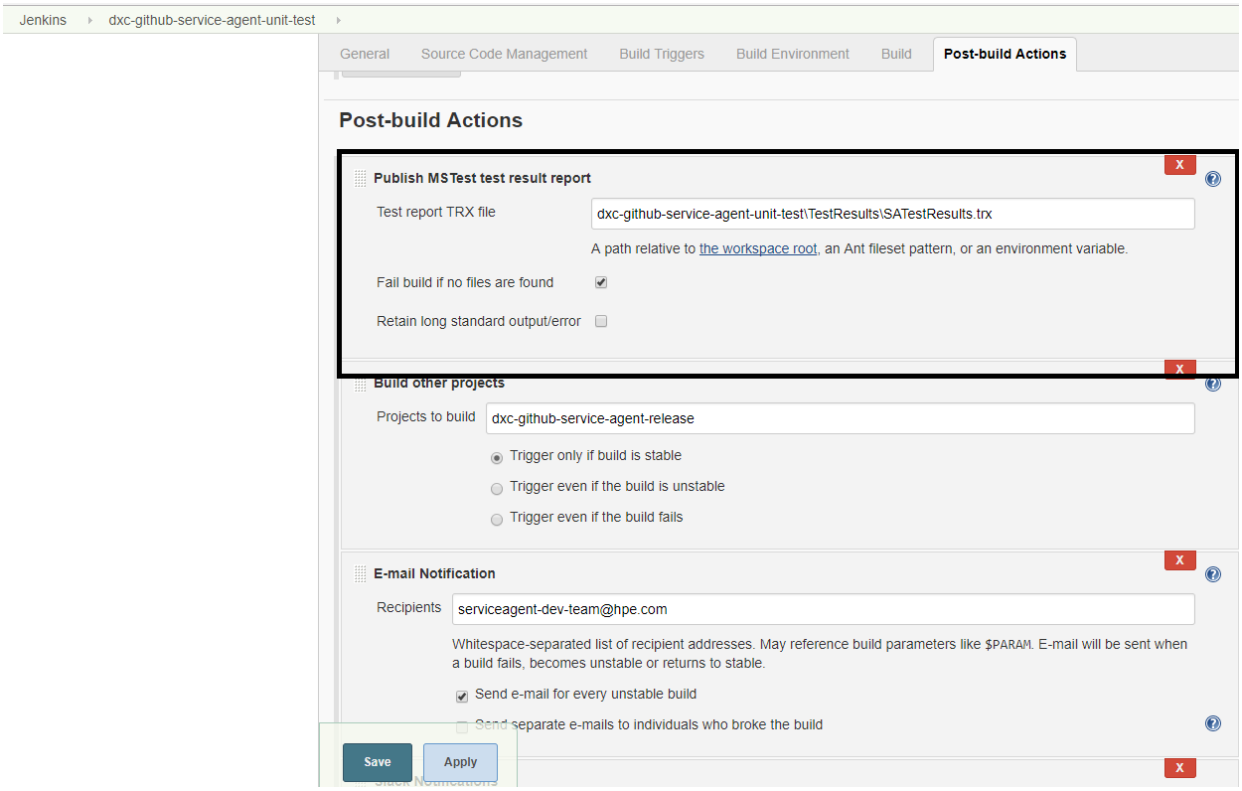




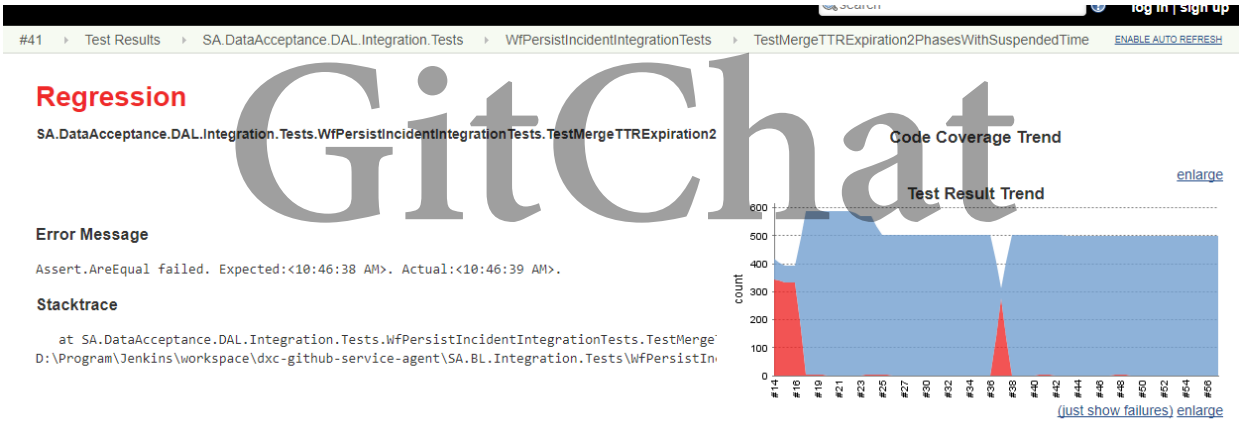
In the configuration page of the job, Build->add build step -> Execute windows batch command...

Below is a sample example for our project.

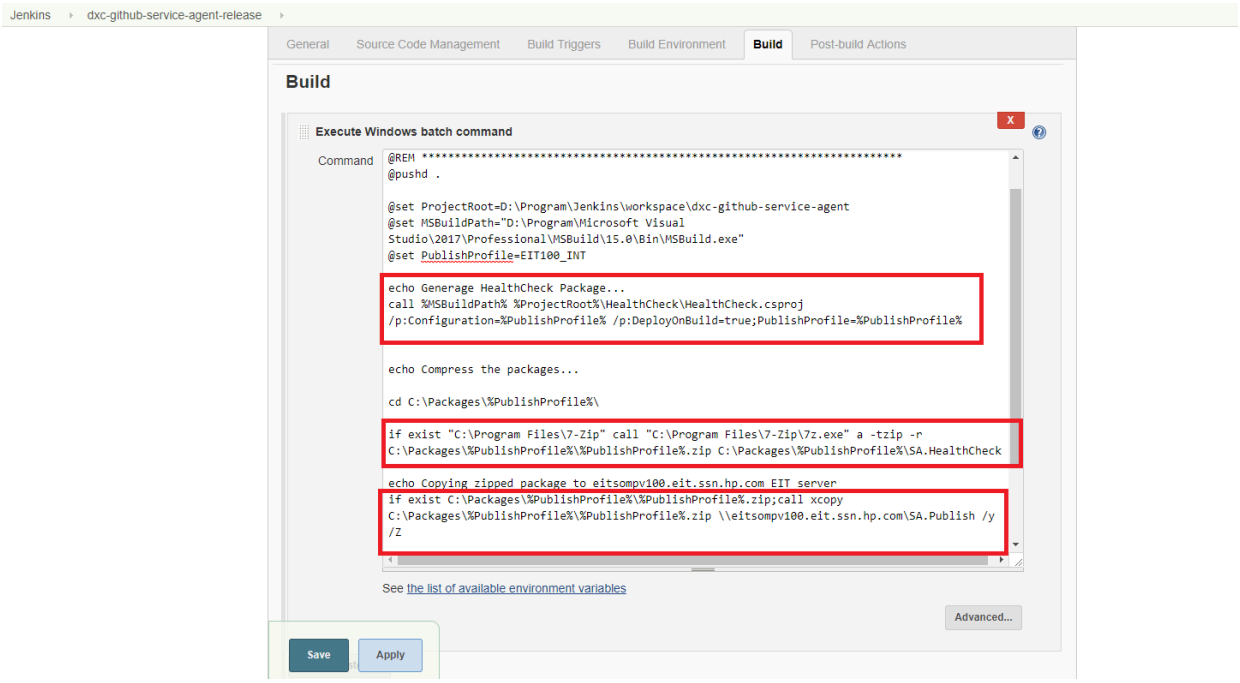
```
@echo "Run Unit Test......"

@set
MySourceCodeWorkSpace=D:\\Program\\Jenkins\\workspace\\dxc-github-
service-agent

@set
MyUnitTestJobWorkSpace=D:\\Program\\Jenkins\\workspace\\dxc-
github-service-agent-unit-test\\

if exist %MyUnitTestJobWorkSpace%\\TestResults == false

md %MyUnitTestJobWorkSpace%\\TestResults

rd /s /q %MyUnitTestJobWorkSpace%\\TestResults\\

md %MyUnitTestJobWorkSpace%\\TestResults

cd "D:\\Program\\Microsoft Visual
Studio\\2017\\Professional\\Common7\\IDE\\"

.\\MSTest.exe

/testcontainer:%MySourceCodeWorkSpace%\\SA.Tests\\bin\\Debug\\SA.T
ests.dll
/testcontainer:%MySourceCodeWorkSpace%\\SA.PdM.BL.Tests\\bin\\Debu
g\\SA.PdM.BL.Tests.dll
/testcontainer:%MySourceCodeWorkSpace%\\CommonUtils.Tests\\bin\\De
bug\\CommonUtils.Tests.dll

/testcontainer:%MySourceCodeWorkSpace%\\SA.Service.Tests\\bin\\Deb
ug\\SA.Service.Tests.dll
```



Below are sample result of the unit test job:

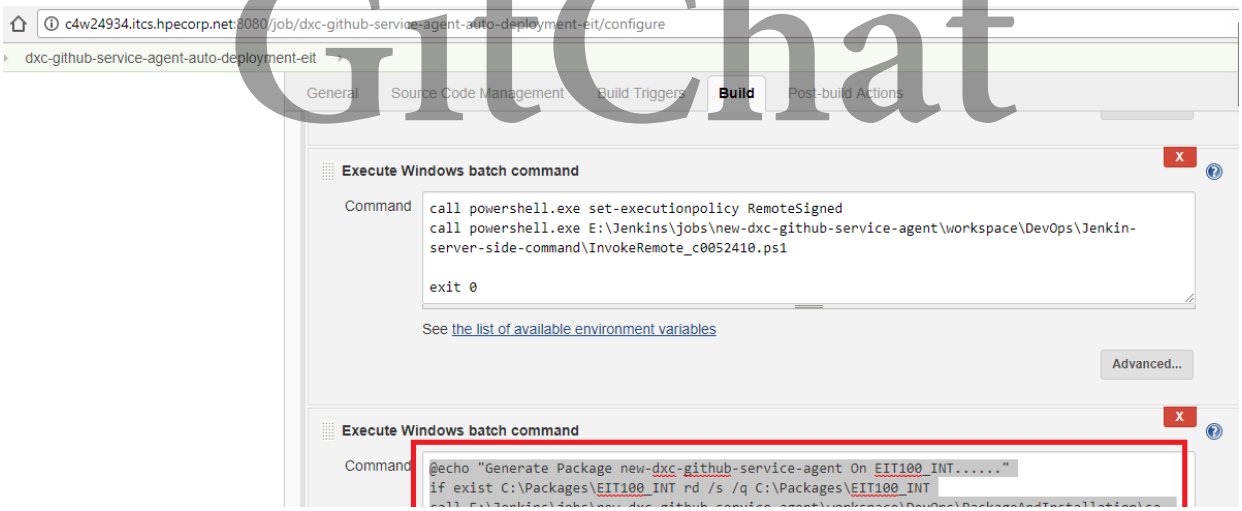


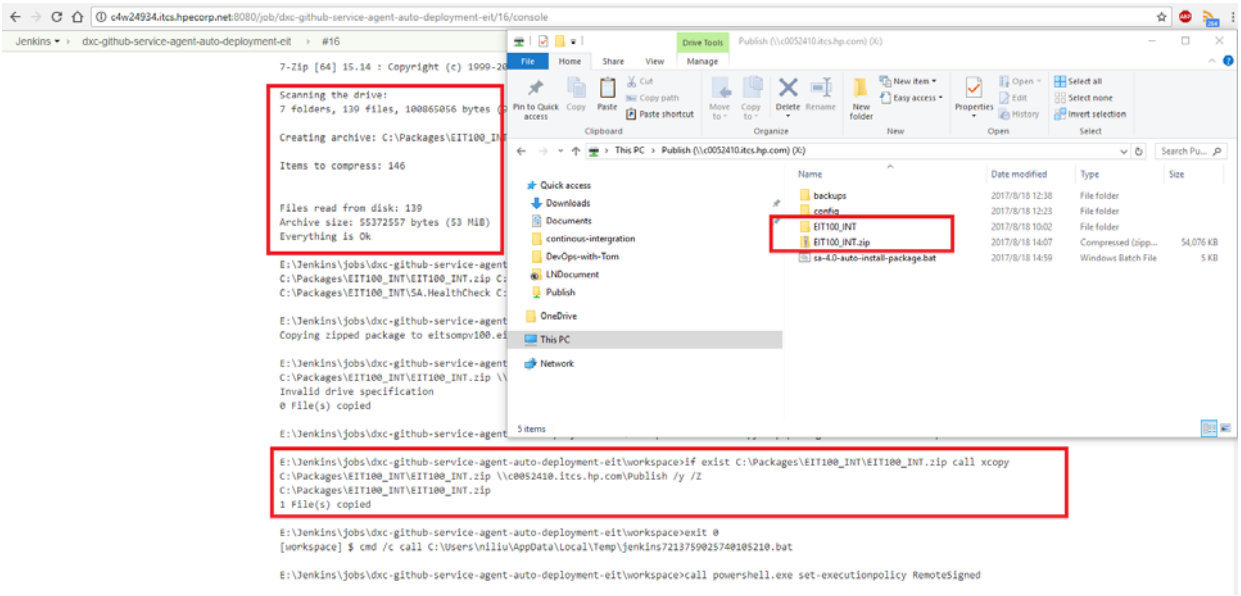
Create a job to generate and release package



For the batch script above, you can also manage it as git source code, maintained directly as code.

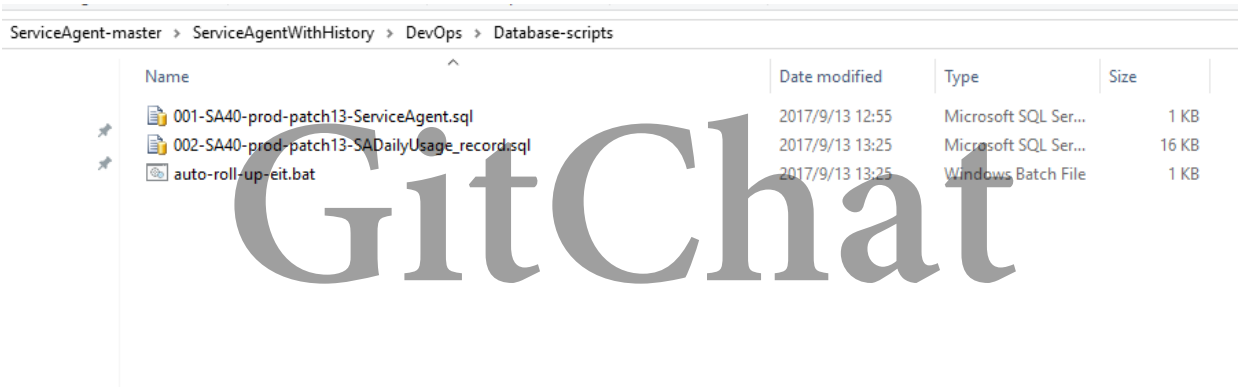
Then configuration will similar like below:





Create a job to install package and db script upgrade

DB scripts is managed as GitHub source code.

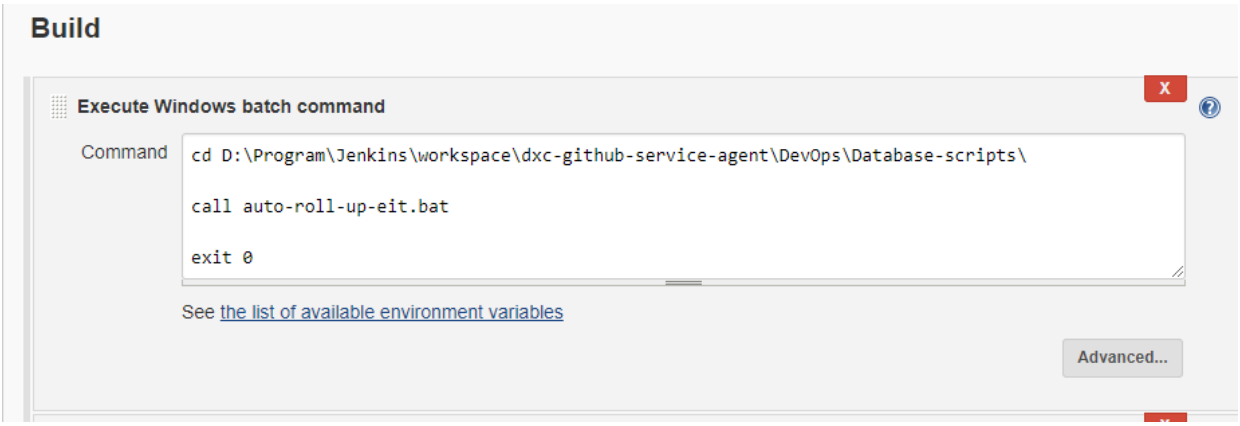


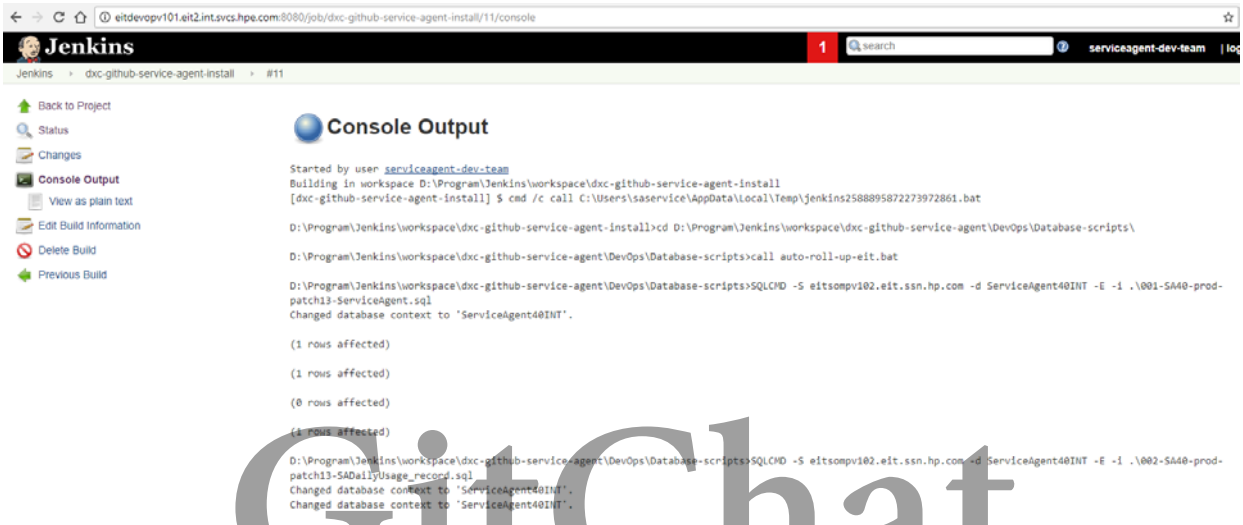
Every time, you add a new script file, update the auto-roll-up.bat.

To include the new added script file.

Then commit the new added script file and updated auto-roll-up.bat.

Below configuration of the installation job is in charge of the db roll upgrade.



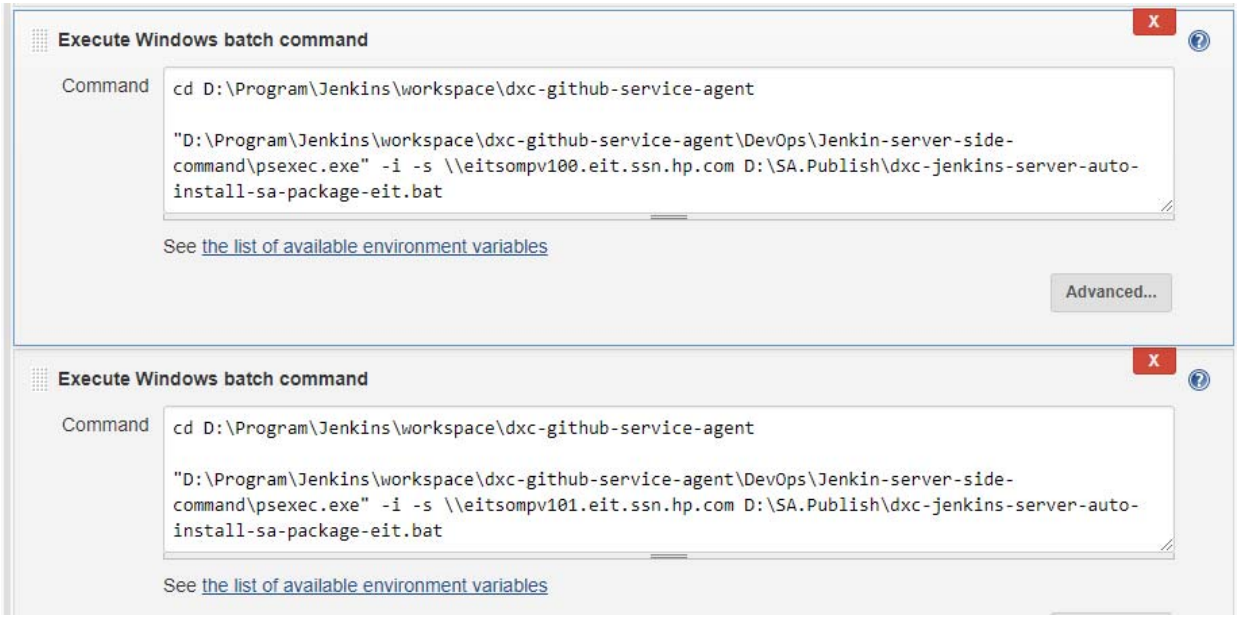


Below configuration of the installation job is in charge of the remotely application installation.

Jenkins server run remote script from Jenkins servers using tool psexec.exe.

So now Jenkins can do continuously delivery to the app server when there is a code change.

(Two others ways I implemented to do remote installation, once is using powershell, another is through ansible, but will not cover in this doc)



Create a job to verify the app using specflow

Create a specflow test project to auto detect the healthy status of the application. Integrate the specflow project as a Jenkins job for verifying if the build if good enough.

For more info about specflow, please check SpecFlow.

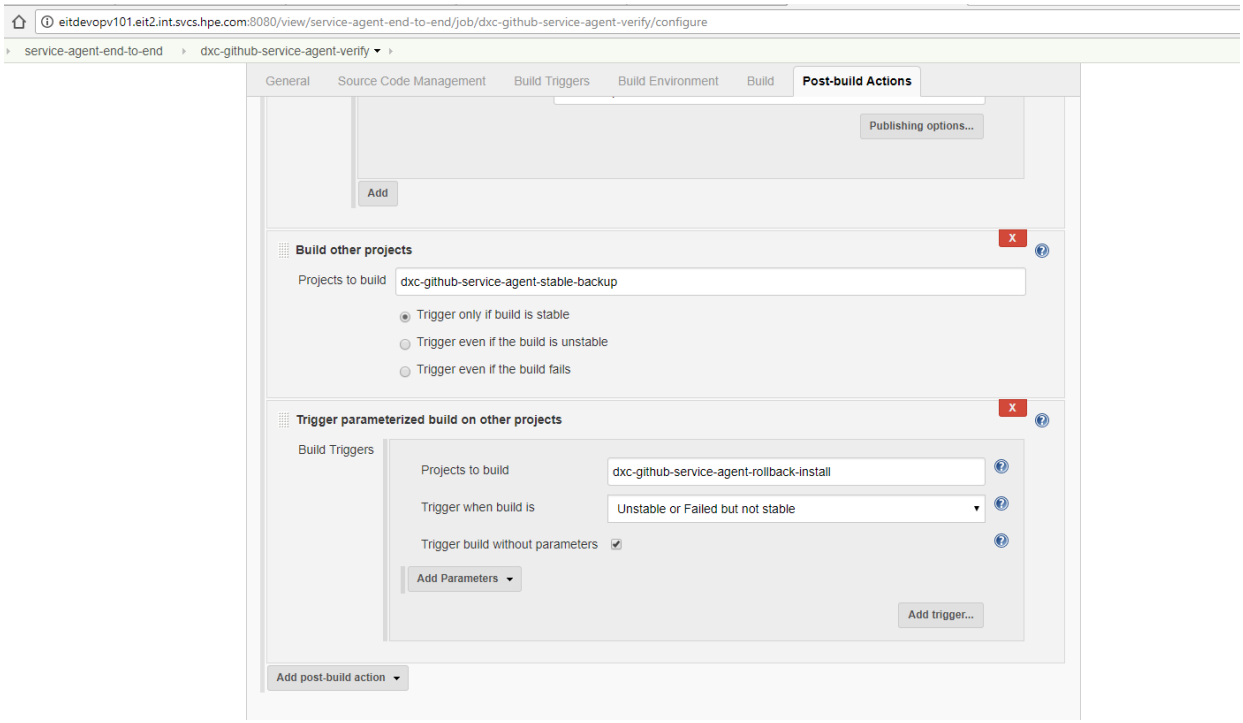
在specflow 项目里，你可以定义一些验证性测试场景用例，它会模拟人工去做smoke test和acceptance test 来决定这次的部署是不是足够stable.

下面是我们项目中一个示例验证步骤场景：

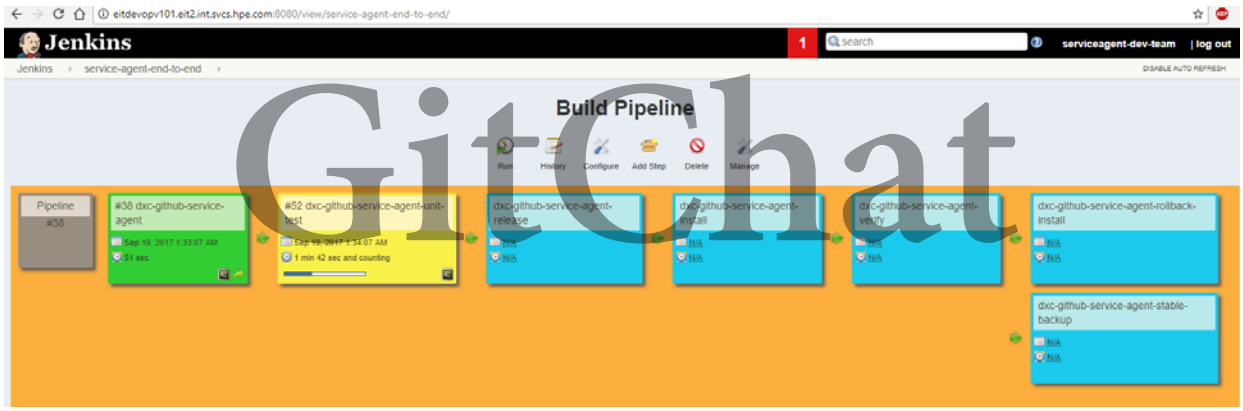
Create a job to auto backup and rollback

If the verification job of specflow test project pass, then back up the app packages as a latest stable version for future use.

If the SpecFlow verification job failed, then rollback the app using the last stable version.



Final Pipeline



Demo the normal install and db roll upgrade

Check Points:

Demo the unstable install and roll back

Code change to simulate failure.

Change .\healthcheck\controllers\homecontroller.cs line 436 and commit to github.

```
if (ctx.SALoadStatus.Count() \< 0)
```

to simulate a healthcheck status checkpoint failure.

Check Points:

1. rollback script applied

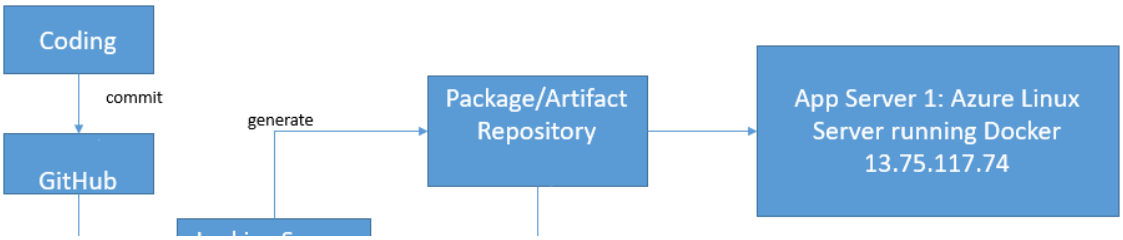
```
select \* from [dbo].[SAApplicationSettings] where Name =
'MassUpdateTicketsInParallel'
```

2. Before rollback start, health check page eit100 with red

3. After rollback finished, health check page eit100 rollback to green

自动部署with-docker-ansible

Below is a flowchart of deployment with docker and ansible



演示录像

链接: https://pan.baidu.com/s/1i5eQshn 密码: 2upi

一些项目实践经验

Best Practices

Normal for single project, we create 4 Jenkins jobs to help the development.

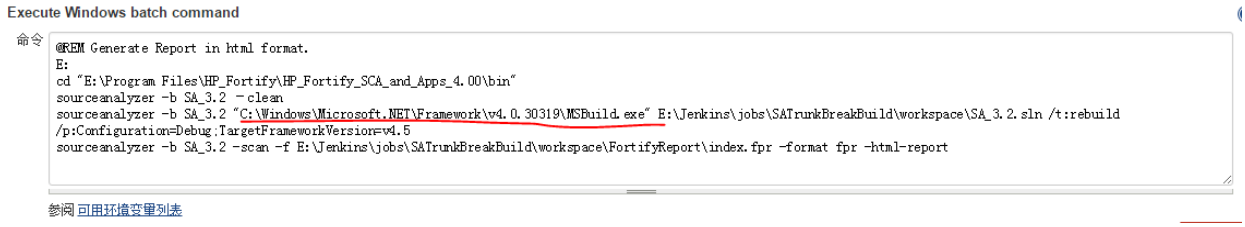
1. Quick Break Build Check— every 5mins, if change made into SVN, then auto trigger the build.
2. Daily Job– two times every day 8:00 and 17:00, get SCM code, build code, unit testing, code quality analyze.
3. Full job – one time every Friday, get SVN code, build code, unit test, code quality analyze, source code API documentation, deployment.
4. Temporary job – target branches code for investigation purpose .(optional)

经验教训 lesson learn

Windows server 2008 R2 上面不能安装visual studio 2015 或更新版本，所以如果你想在Jenkins server 上直接安装visual studio, 至少Jenkins server版本要Window server 2012

Troubleshooting

```
sourceanalyzer -b SA_3.2
"C:\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\MSBuild.exe"
E:\\Jenkins\\jobs\\SATrunkBreakBuild\\workspace\\SA_3.2.sln
/t:rebuild
/p:Configuration=Debug;TargetFrameworkVersion=v4.5
```



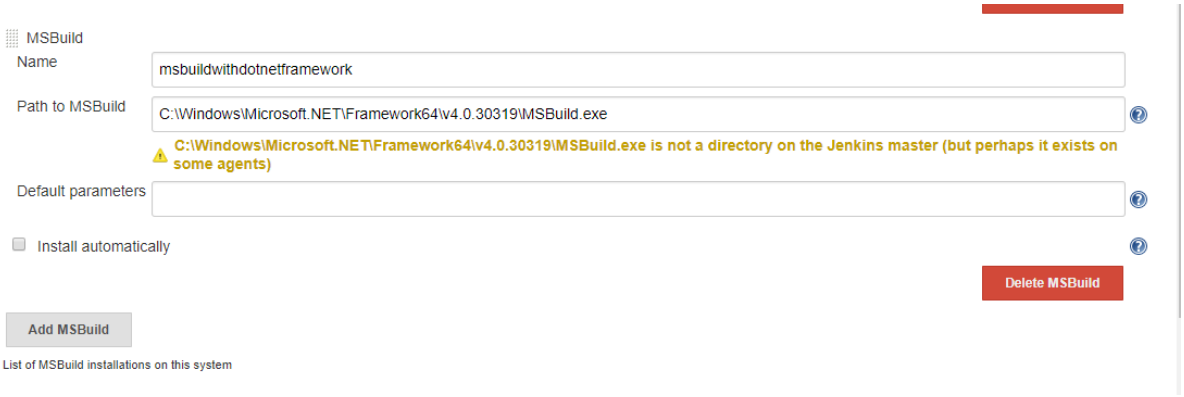
MSBuild Error：MSB4175: The task factory “CodeTaskFactory” could not be loaded from the assembly

```
Details error:  MSB4175: The task factory "CodeTaskFactory" could
not be loaded
from the assembly "C:\\Program Files
(x86)\\MSBuild\\12.0\\bin\\Microsoft.Build.Tasks.v4.0.dll". Could
not load file
or assembly 'file:///C:\\Program Files
(x86)\\MSBuild\\12.0\\bin\\Microsoft.Build.Tasks.v4.0.dll' or one
of its
dependencies. The system cannot find the file specified.
```


Root Cause：

Somebody when create the project, the tools version target to 12.0, which means using the MSBuild

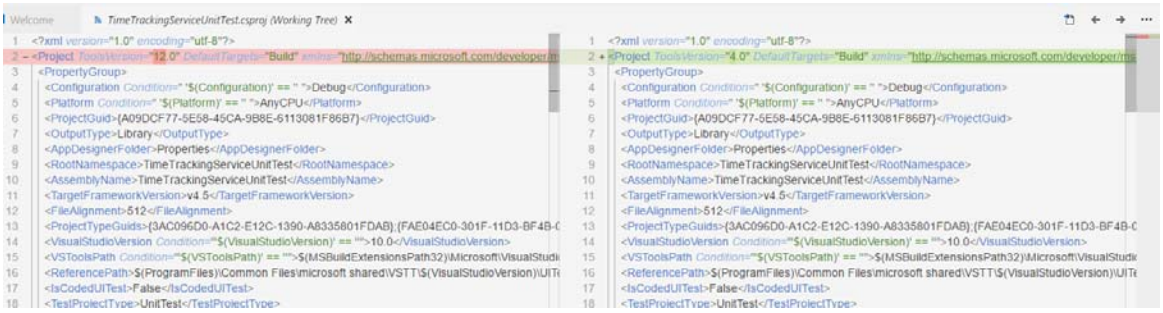
Along with visual studio 2013 tool which located at C:\Program Files(x86)\MSBuild\12.0\bin



Solution:

1. If in the Jenkins server, you have the visual studio 2013 installed, or you have standalone Microsoft Build Tools 2013 from https://www.microsoft.com/en-us/download/details.aspx?id=40760 installed, you can update the msbuild path

2. If you want to depend on dotnet framework only, you can update *.csproj file which report the error to set the ToolsVersion from 12.0 to 4.0



12

NuGet can use local settings for it's behavior which can be unpredictable if you're not 100% sure how the server is configured.

I prefer putting the NuGet settings inside the `<sln root>/.nuget/NuGet.targets` file which is version controlled and at a single location. I got this working with 3 quick edits to `<sln root>/.nuget/NuGet.targets`, they should look as below after editting:

Change 1:

```
<!-- Enable the restore command to run before builds -->
<RestorePackages Condition="  '$(RestorePackages)' == '' ">true</RestorePackages>
```

Change 2:

```
<!-- Determines if package restore consent is required to restore packages -->
<RequireRestoreConsent Condition=" '$(RequireRestoreConsent)' != 'false' ">false</RequireRes
```

My comment: Awkward logic but think of "*requires consent not equal to false must be true*" (original) as "*requires consent equal to true must be true*" (translated) and it makes sense to change the last part to "*false*" (the edit)

Change 3 : I also added/uncommented the `<PackageSource ... >` tag to to remove any dependencies on the

```
<ItemGroup Condition=" '$(PackageSources)' == '' ">
    <PackageSource Include="https://nuget.org/api/v2/" />
</ItemGroup>
```

https://stackoverflow.com/questions/12788521/nuget-package-restore-not-working-on-build-server

Issue : Specflow report Error: WatiN.Core.Exceptions.ElementNotFoundException: Could not find INPUT (hidden) or INPUT (password)

Root Cause: IIS failed to run with error below:

```
[SecurityException: Request for the permission of type
'System.Security.Permissions.SecurityPermission, mscorlib,
Version=4.0.0.0,
```

```
\<trust level="Full" /\>

\</system.web\>
```

Solution: Jenkins running a user don't have the db access.

Run below script in sql.

CREATE LOGIN [americas\$sadev001] FROM WINDOWS

Issue : Specflow report Error: WatiN.Core.Exceptions.RunScriptException: RunScript failed —> System.UnauthorizedAccessException

Solution: Adding the test site to the IE's trusted sites list and it has resolved the error message.

Issue : Specflow issue: the 'validation' attribute must be one of these values: SHA1

```
Parser Error Message: When using \<machineKey
compatibilityMode="Framework45"
/\> or the MachineKey.Protect and MachineKey.Unprotect APIs, the
'validation'
attribute must be one of these values: SHA1, HMACSHA256,
HMACSHA384, HMACSHA512,
or alg:[KeyedHashAlgorithm].
```


Solution:

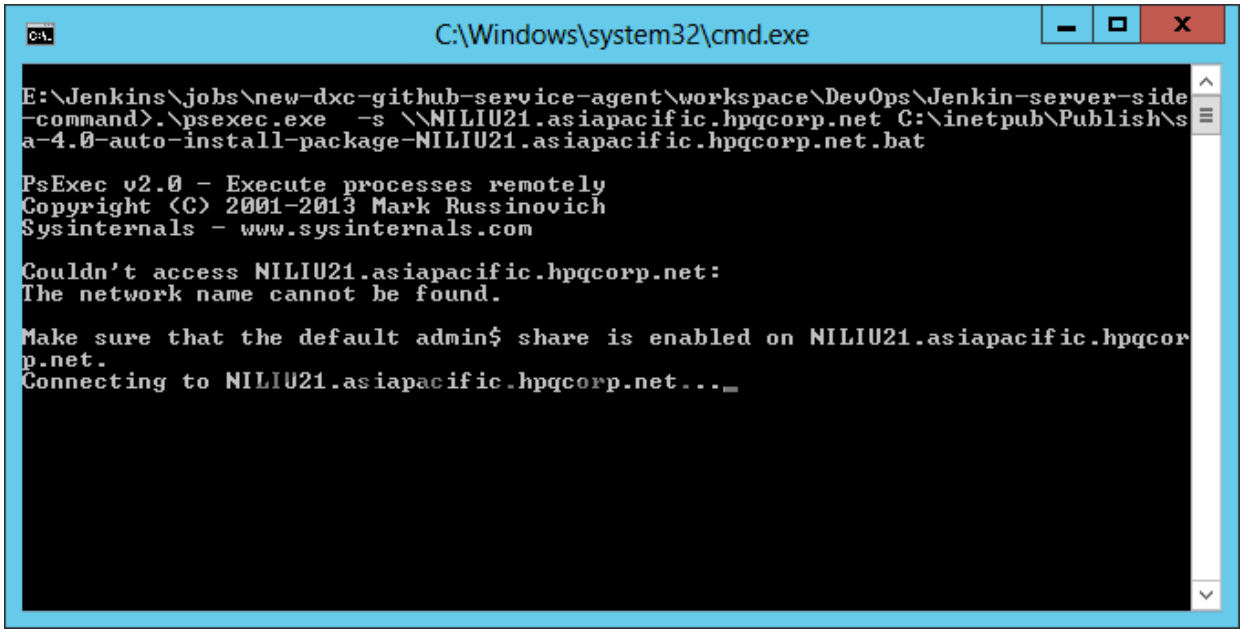
```
C:\\Windows\\Microsoft.NET\\Framework64\\v4.0.30319\\Config\\web.
config line 400
```

Add the **-i** options highlighted below.

```
"E:\\Jenkins\\jobs\\new-dxc-github-service-
agent\\workspace\\DevOps\\Jenkin-server-side-command\\psexec.exe"
-i -h -s \\\\c0052410.itcs.hp.com D:\\Publish\\sa-4.0-auto-
install-package.bat
```

Issue : Psexec return error code 6 The handle is invalid.

Details: when run in the commandline, will see the details error, caused by admin$ share not set,



Follow:

https://stackoverflow.com/questions/18388381/make-sure-that-the-default-admin-share-is-enable-on-servername

Solution:

http://aangelov.com/2014/08/30/getting-started-jenkins-msbuild-nuget-git/

www.martinfowler.com/articles/continuousIntegration.html

http://redsolo.blogspot.com/2008/04/guide-to-building-net-projects-using.html

http://martinfowler.com/articles/continuousIntegration.html

http://jenkins-ci.org/

https://jenkins.io

Jenkins distributed build master/slave node