

Jenkins Job Configure for Sitcore

|  |  |
| --- | --- |
| **Author** |  |
| **Approved By** |  |
| **Application Version** |  |
| **Date of Issue** |  |
| **Release Note Number** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document Control Section** | | | | |
|  | | | | |
| **Revision History:** | | | | |
| **Document version** | **Pages** | **Action** | **Date of revision** | **By** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | | | | |

Table of Content

[1.0 Confidentiality / Legal notification 3](#_Toc331583772)

[2.0 Introduction 5](#_Toc331583773)

[2.1 Jenkins Job CONFIGURATION for JOB CONFIGURATION **5**](#_Toc331583774)

1. Confidentiality / Legal notification

Confidential. Copyright © AI Enterprise., 2018. All rights reserved. This document, including the information contained

herein, is restricted, confidential and proprietary to AI Enterprise, and is to be used only by and disclosed only to those

within AI Enterprise with a need-to-know. DO NOT COPY OR FORWARD INTERNALLY OR RELEASE outside AI Enterprise

without authorization in writing by a Sr. Vice President, Principal, or Director-level manager or a direct designee

thereof who has responsibility for the information contained herein.

1. Introduction

This document describes about configure Jenkins job for Sitecore including build and deployment on Windows Instance.

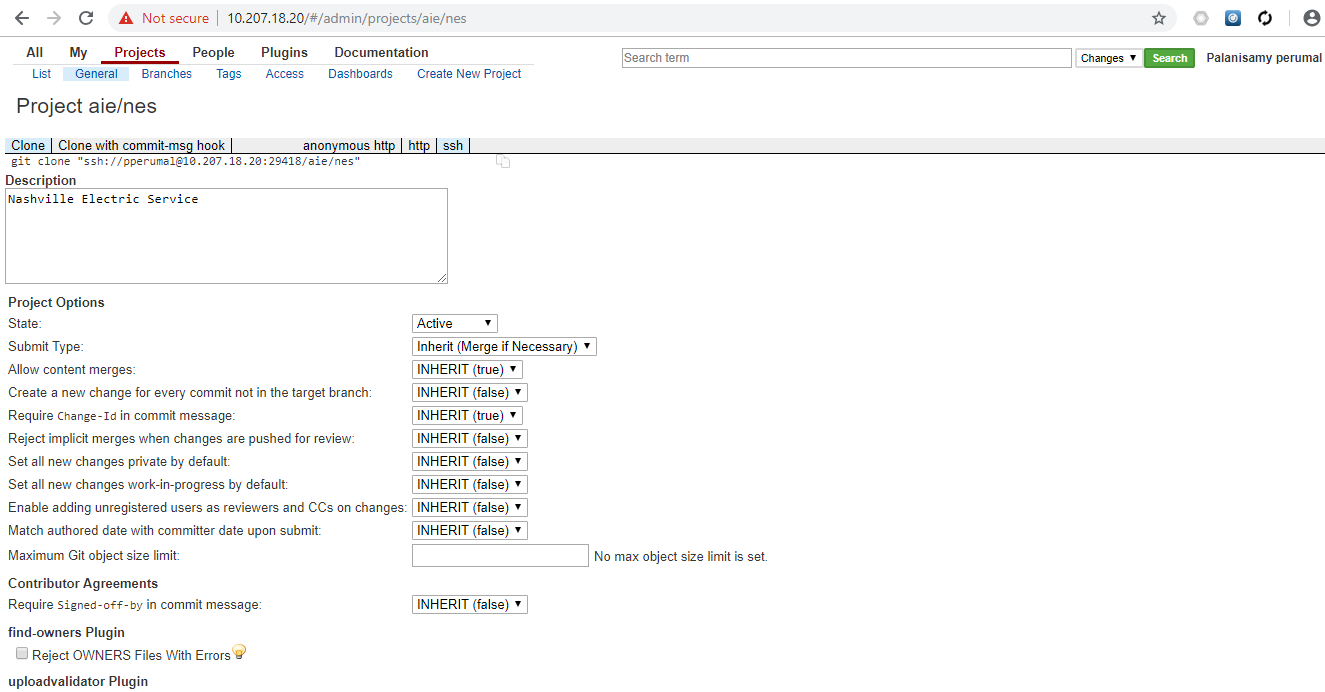
Sitecore is one of the leading enterprise-level content management systems built on ASP.NET. Sitecore is enabling web content editors and marketers to have full control over all aspects of their website from social integration and blog posts to advanced personalisation, ecommerce and more.

Source code is store in Gerrit server and job is configured in our Jenkins which is hosted on Google cloud platform.

* 1. Jenkins job Configuration

Source code

You can download the complete source code from AIE gerrit repository. Please refer the sample pasted snippet of repo path.

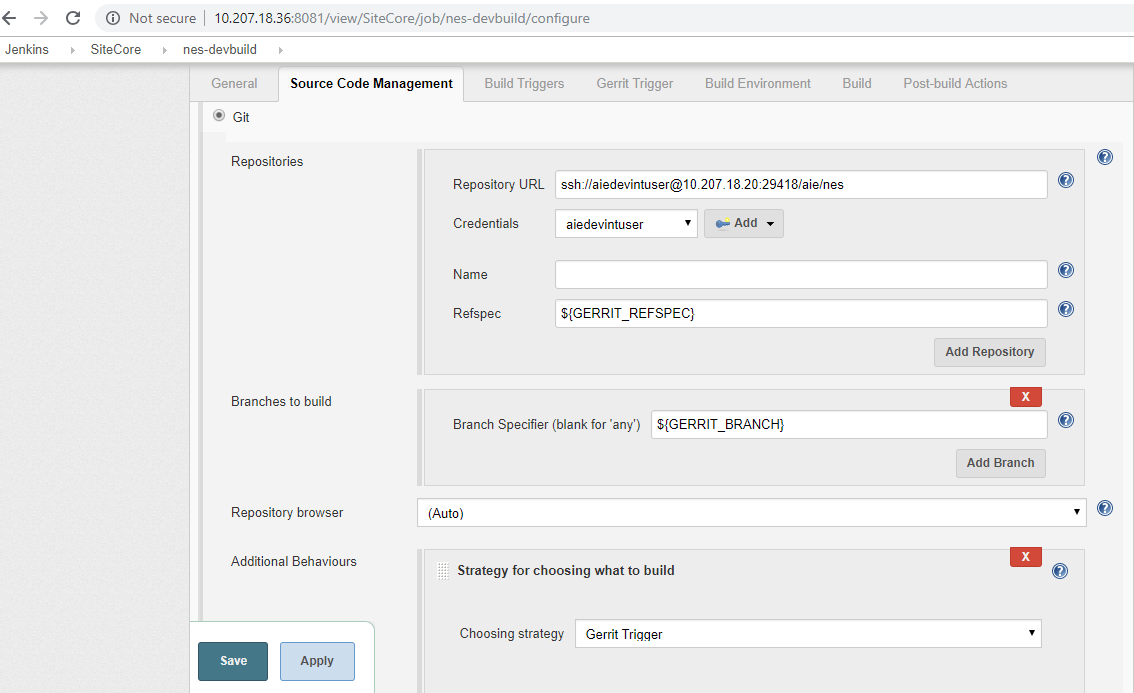


We’ve created a separate project in the Jenkins view and mapped this site core job. We’ll detail look into this in below screenshots.

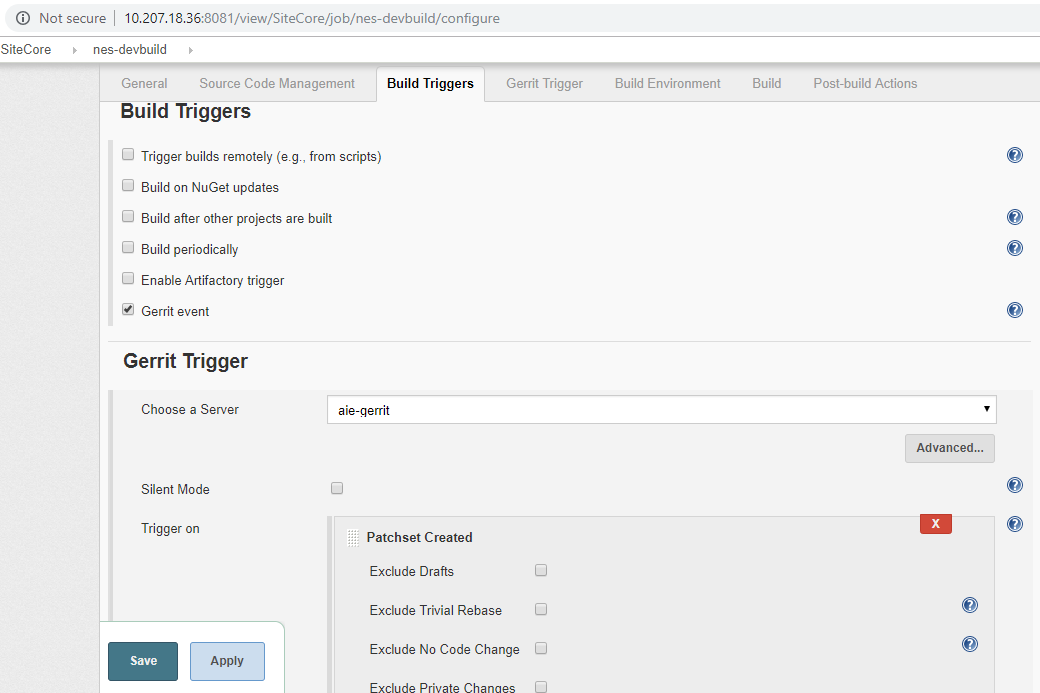
http://10.207.18.36:8081/view/SiteCore/job/nes-devbuild/configure



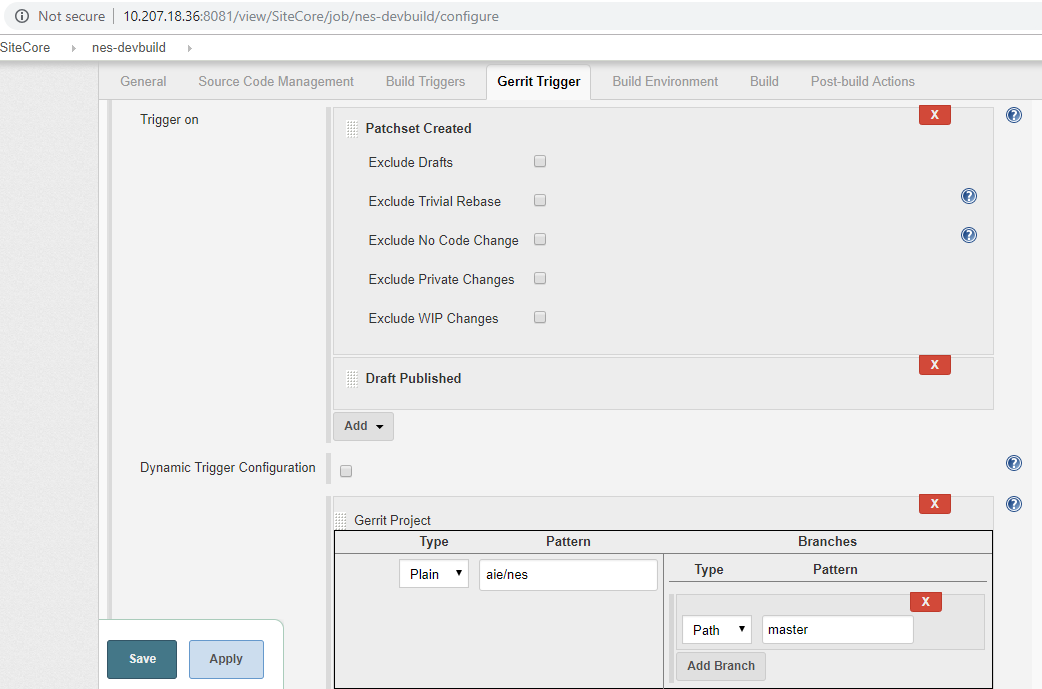
We’ve configured windows server (10.207.18.37) as Jenkins slave. Need to select the label while job is schedule from Jenkins master.



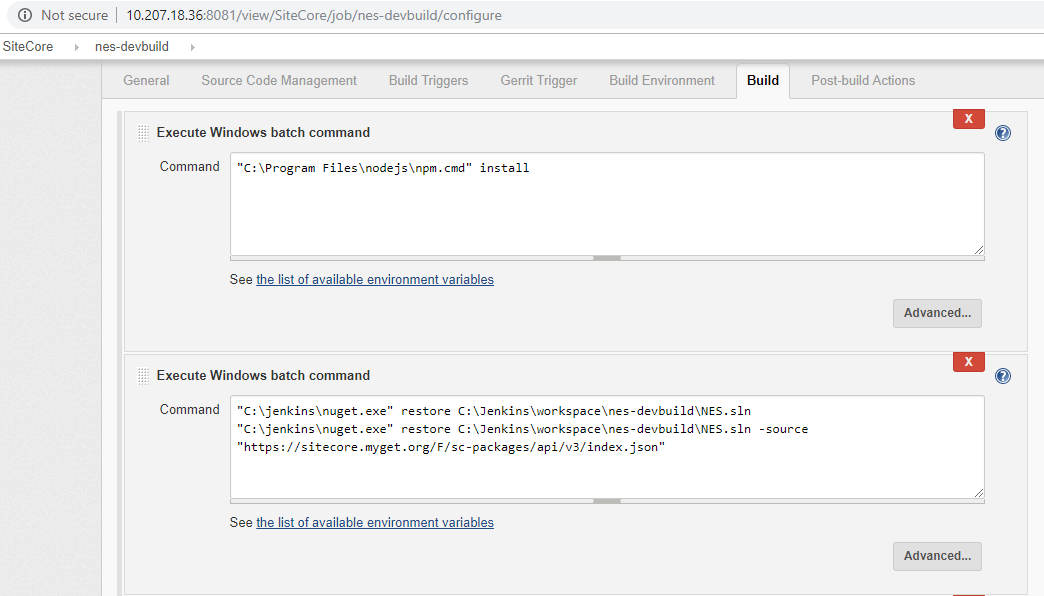
Source code management select as Git and provide the source code repository path as specified in above screenshot.



At least one build trigger needs to select, so that when developer push codes to Gerrit, Jenkins jobs will automatically trigger through webhooks. In order to achieve this, required plugins already installed in Jenkins and configurations enabled in Gerrit as well.



As per Dev team request, Job needs to trigger and build while push codes to master branch. This can be achieved by selecting path in the Dynamic trigger configuration section.



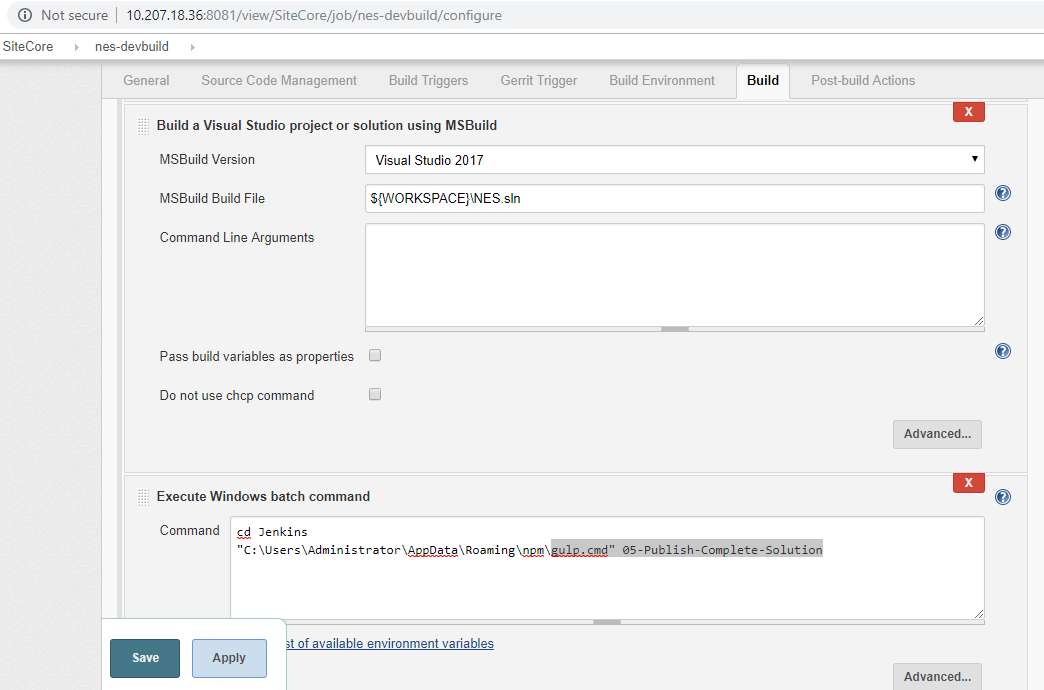
We’re using following commands for build NES codes from solution files. It’s formally called .sln file.

npm install

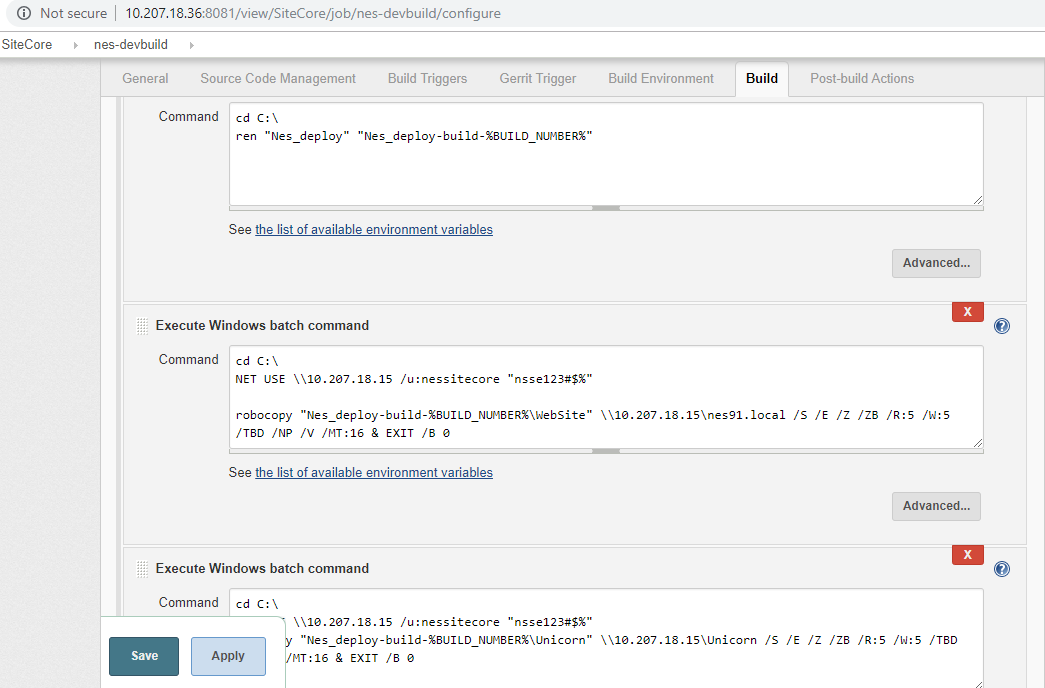
nuget restore NES.sln

nuget restore NES.sln -source <https://sitecore.myget.org/F/sc-packages/api/v3/index.json>

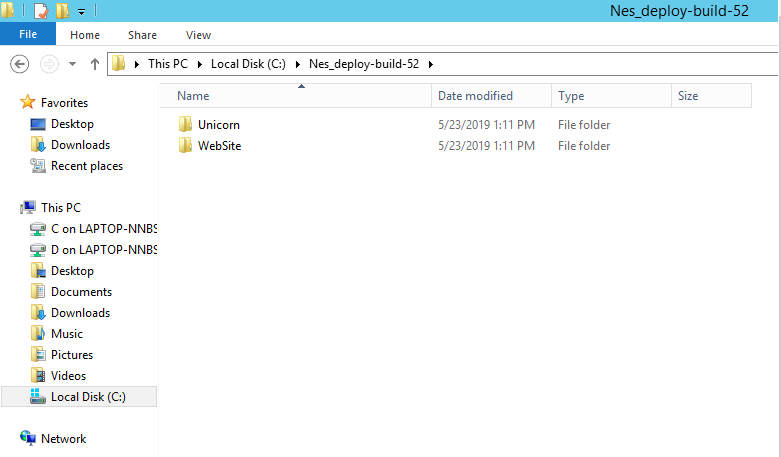
gulp 05-Publish-Complete-Solution



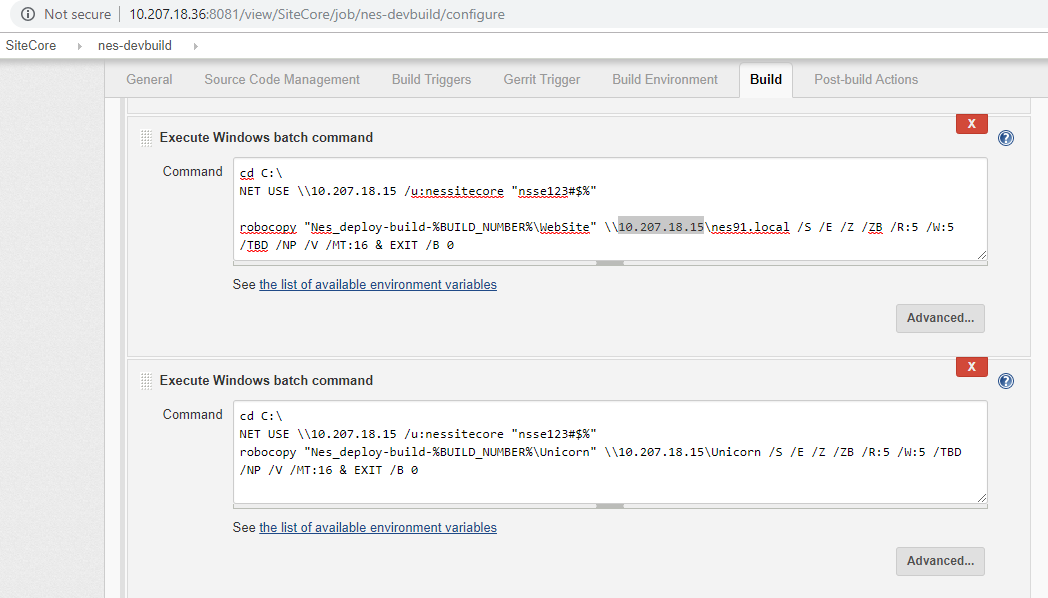
If you’re configuring new jobs from vanilla Jenkins then make sure installed MSBuild plugins before configure this jobs.



Once build is completed, output files will be stored in Jenkins slave Instance (10.207.28.37) at C:\Nes\_deploy-build-Jenkinsbuildnumber directory.



This build output files are copying to windows instance (10.207.18.15) using windows robocopy.



The new code level changes are able to see immediately while hitting the url dev.nes91.local