Jenkins

**What is Jenkins?**

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

With Jenkins, organizations can accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis and much more.

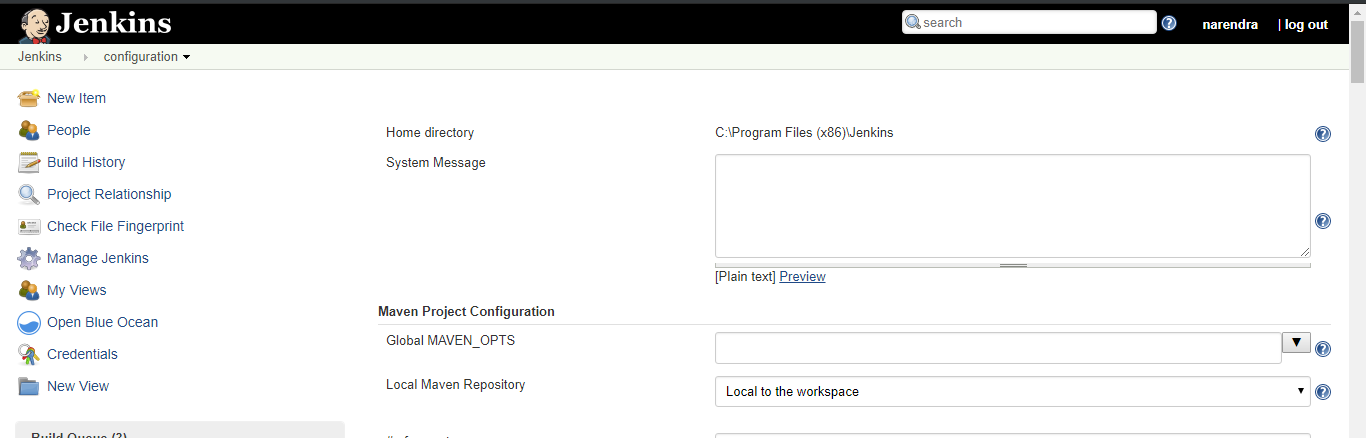
Jenkins achieves Continuous Integration with the help of plugins. Plugins allows the integration of Various DevOps stages. If you want to integrate a particular tool, you need to install the plugins for that tool. For example: Git, Maven 2 project, Amazon EC2, HTML publisher etc.

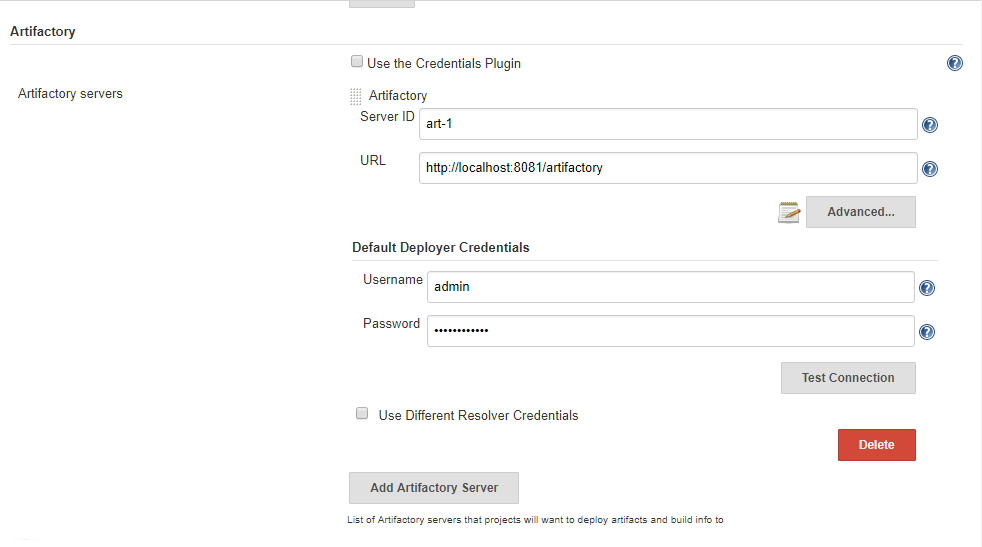
Installation and Configuration

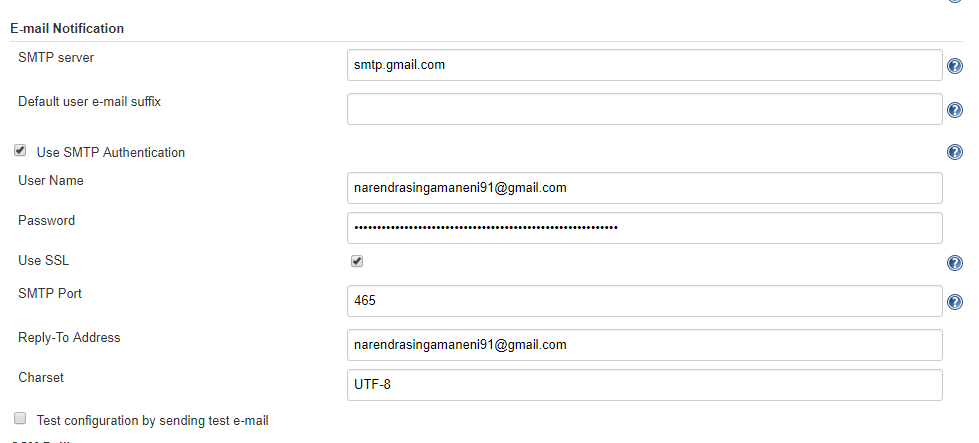
<https://dzone.com/articles/how-to-install-jenkins-on-windows>

Manage Jenkins

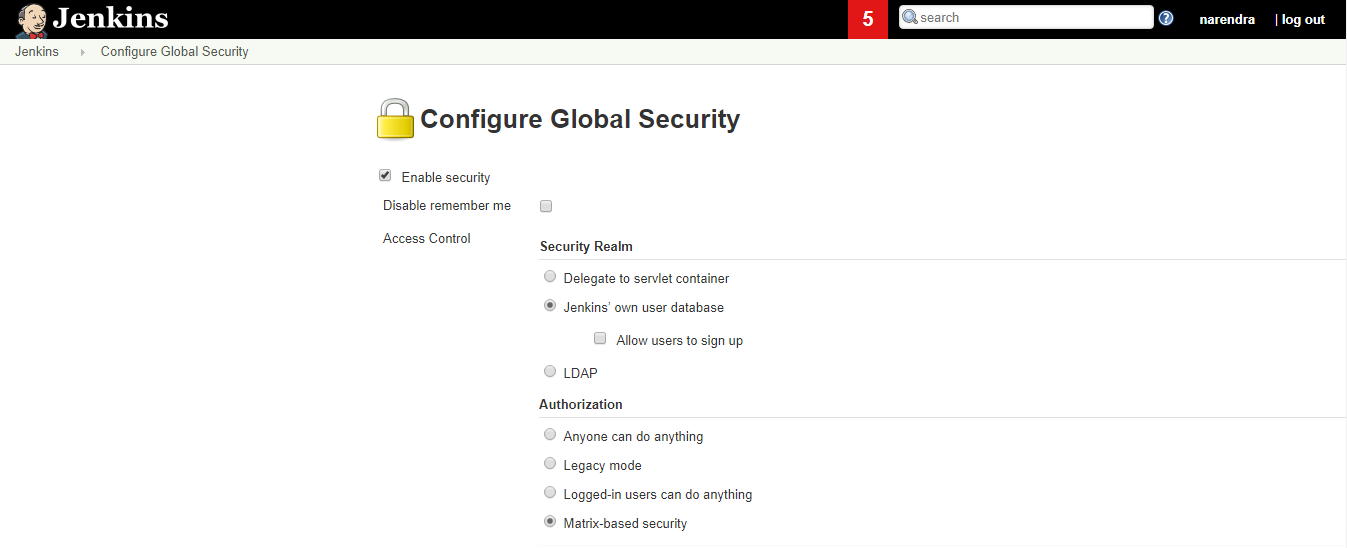
Configure Systems



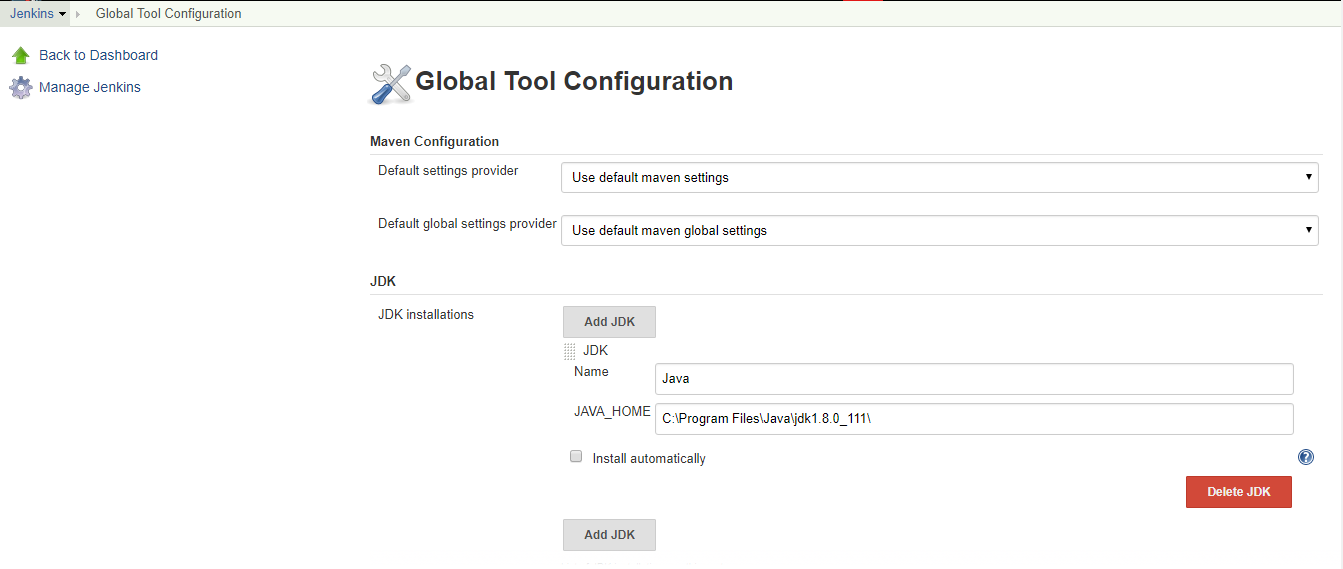


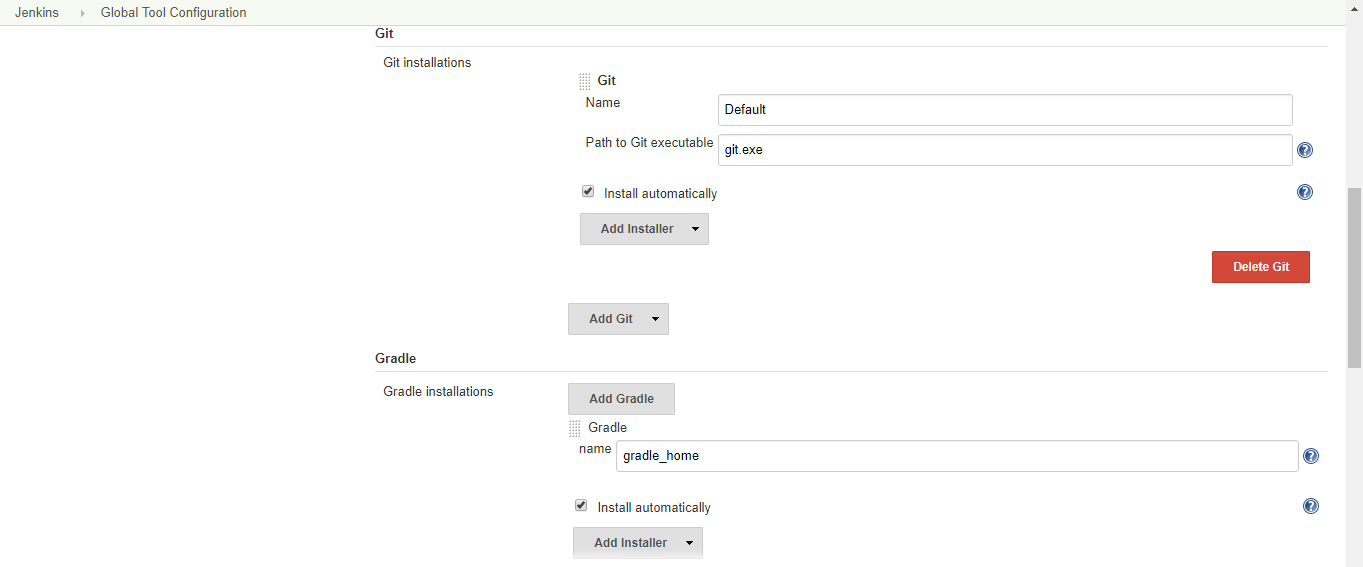


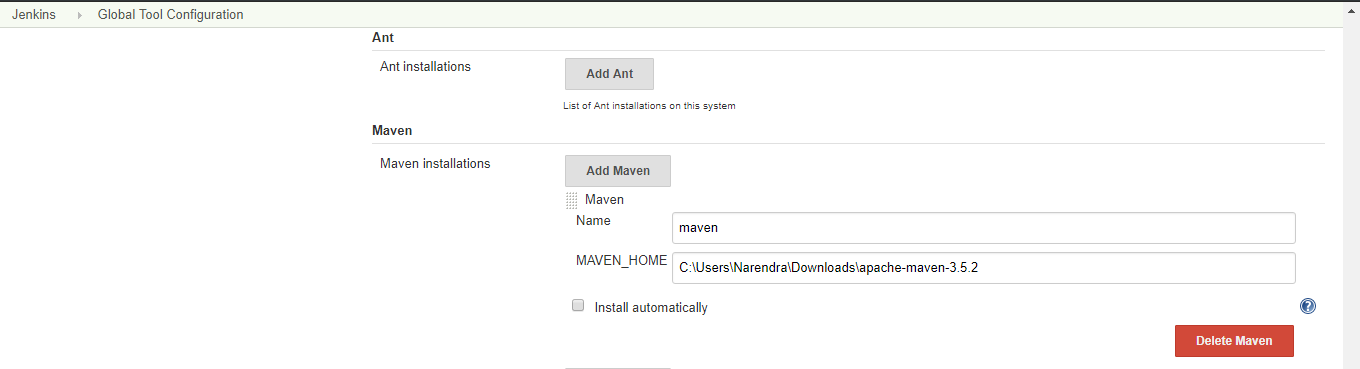
Global



Global Tool Config

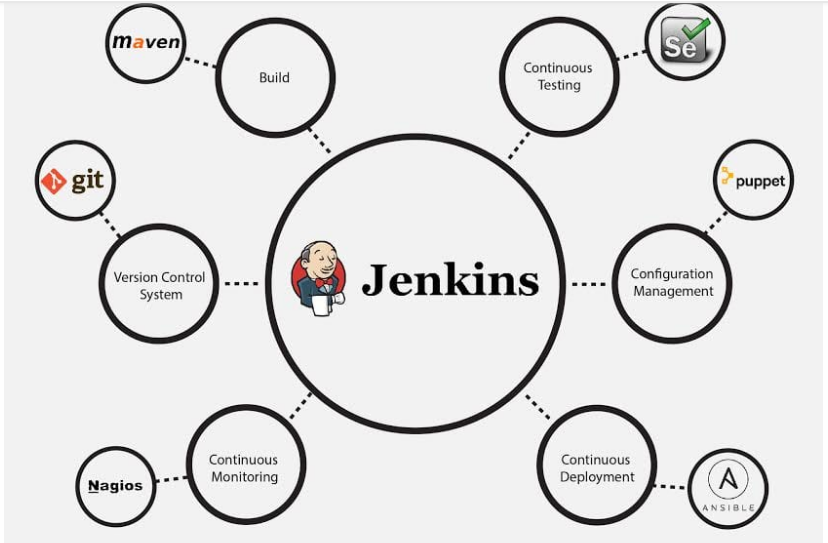






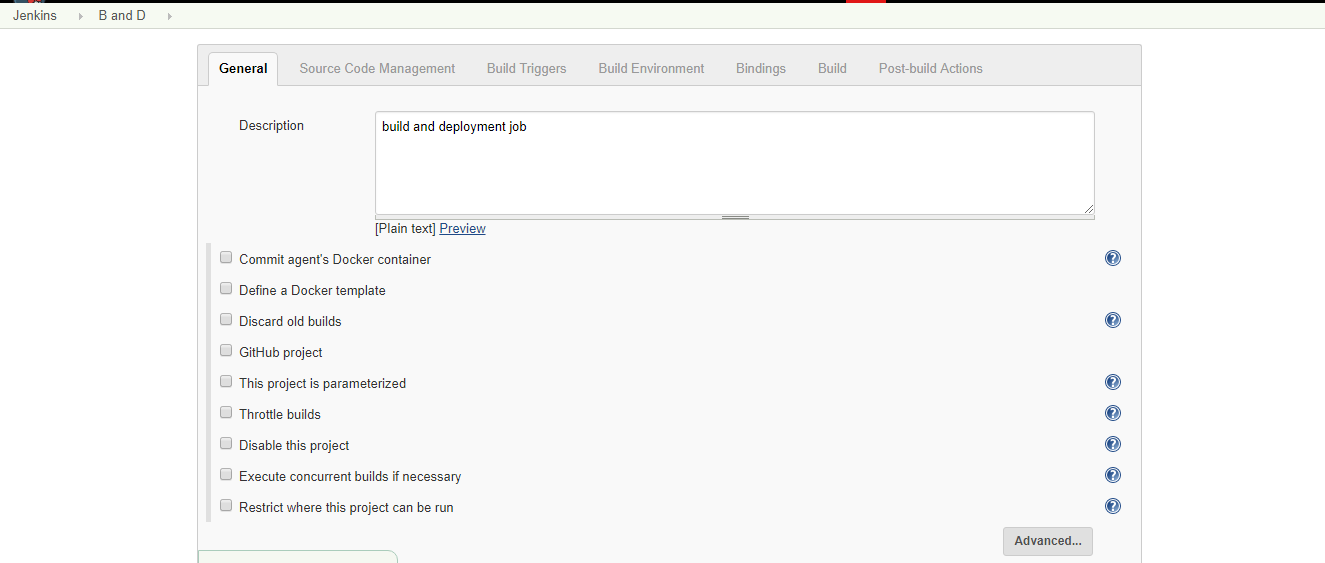
## What is Continuous Integration?

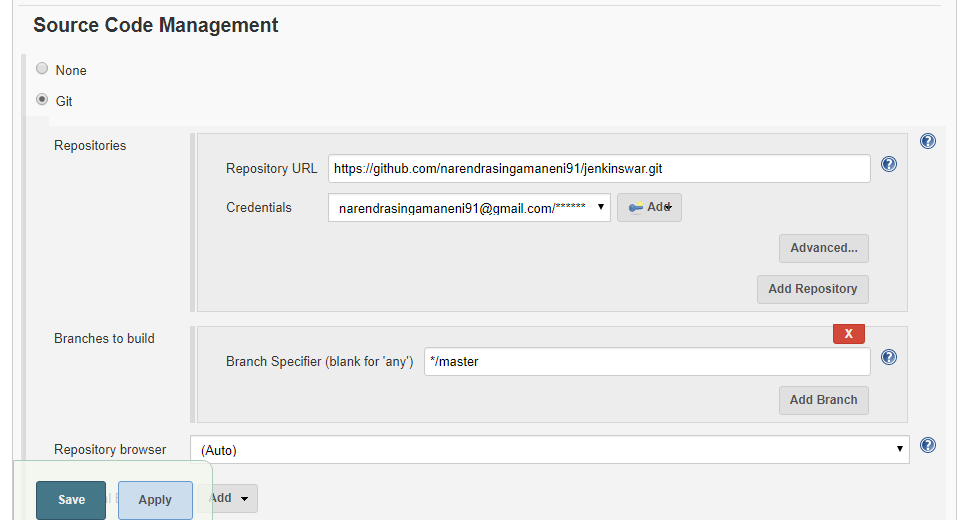
Continuous Integration is a development practice in which the developers are required to commit changes to the source code in a shared repository several times a day or more frequently. Every commit made in the repository is then built. This allows the teams to detect the problems early. Apart from this, depending on the Continuous Integration tool, there are several other functions like deploying the build application on the test server, providing the concerned teams with the build and test results etc.

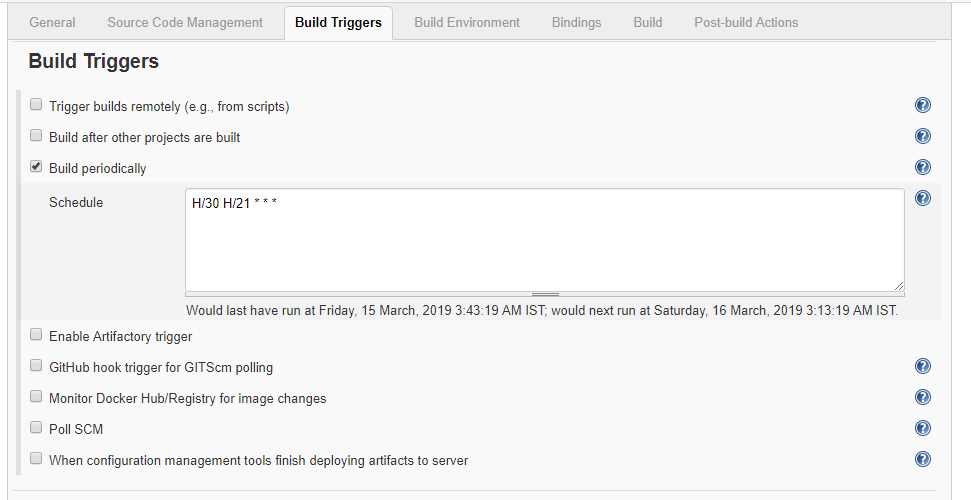


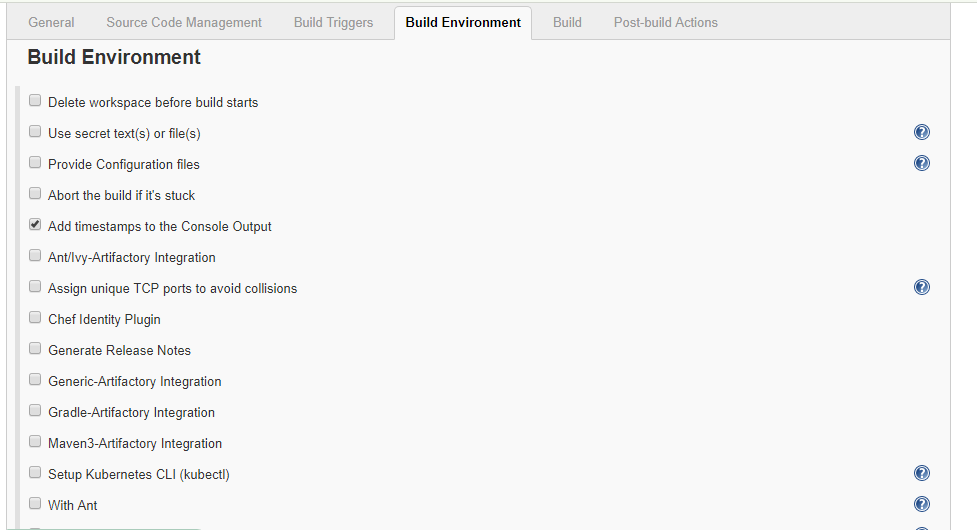
Free Style Creation

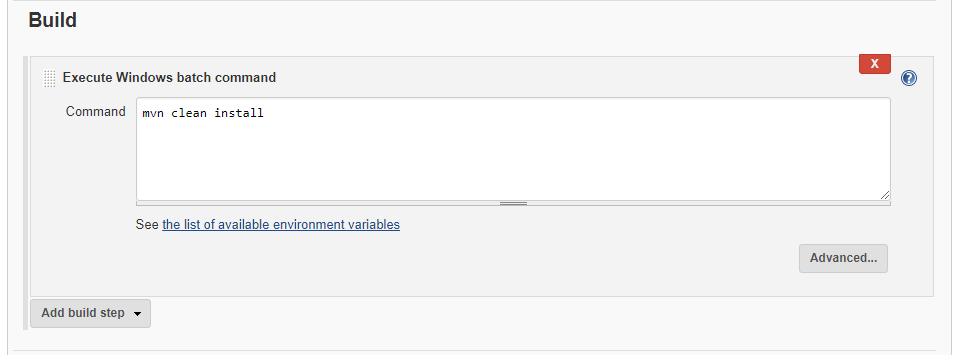
<https://www.guru99.com/create-builds-jenkins-freestyle-project.html>

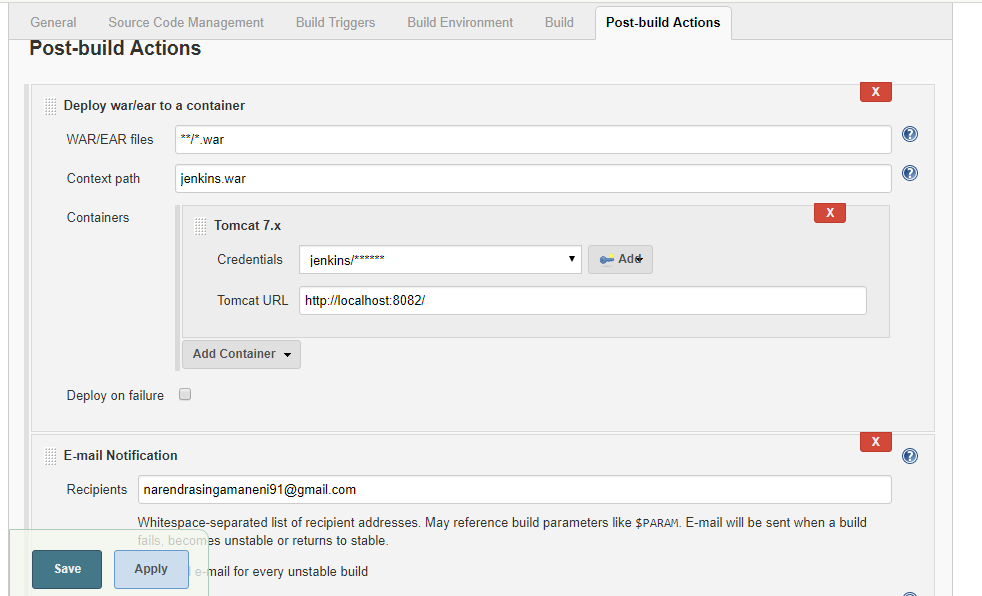






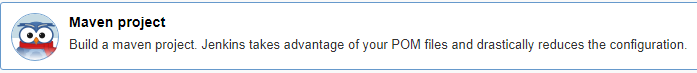




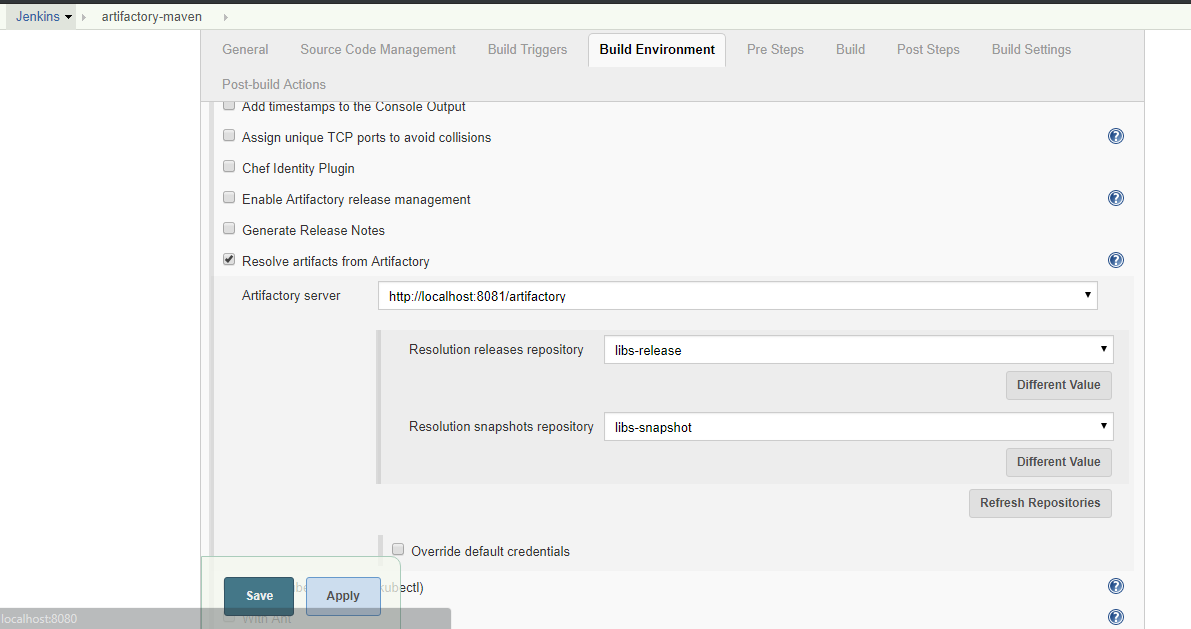
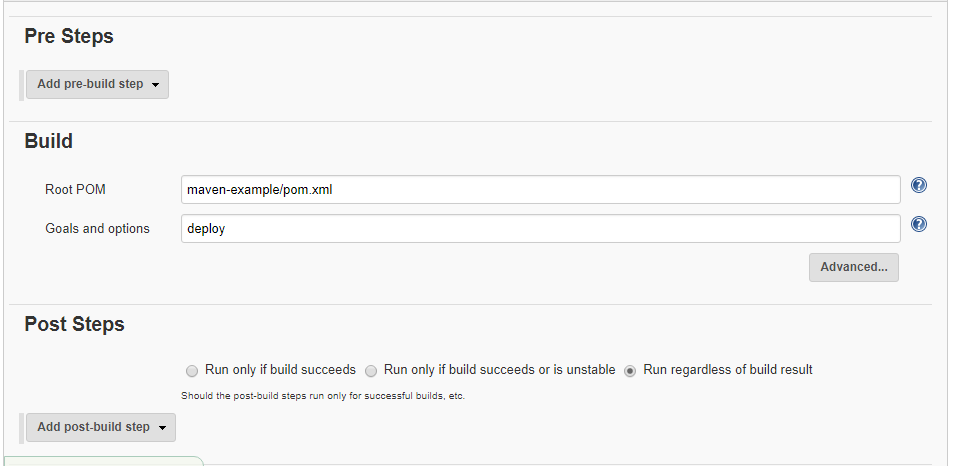


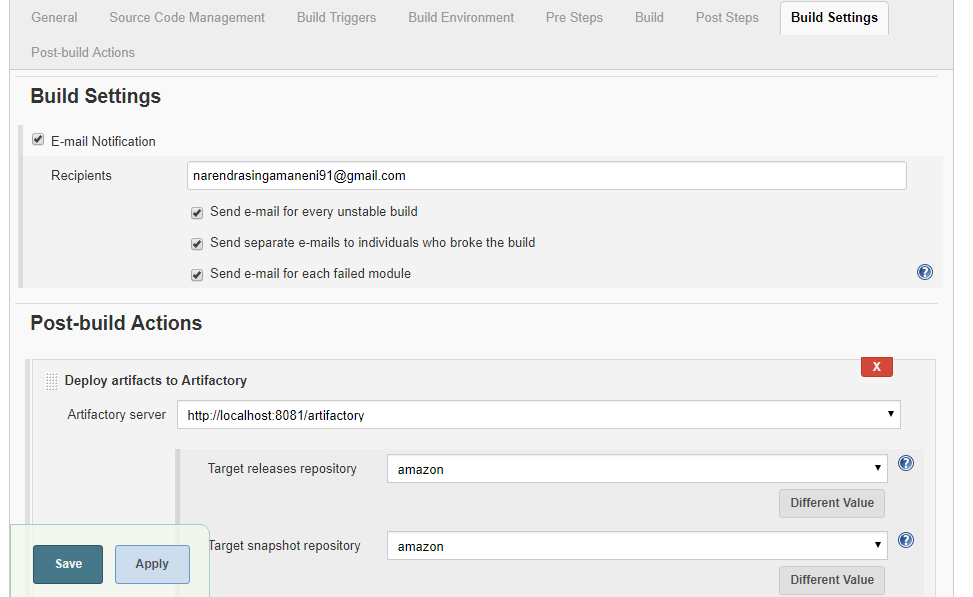
Maven Job

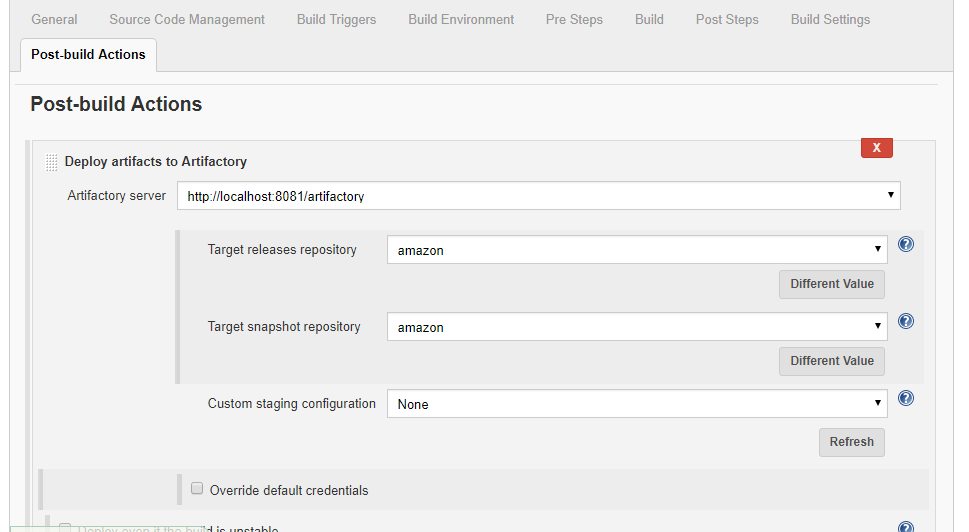
General and SCM and triggers is same for maven job also



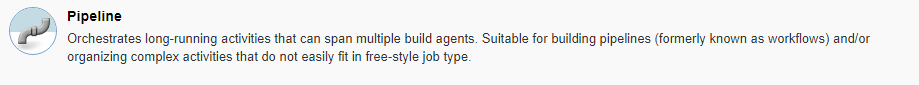
Build env:





Pipeline Job

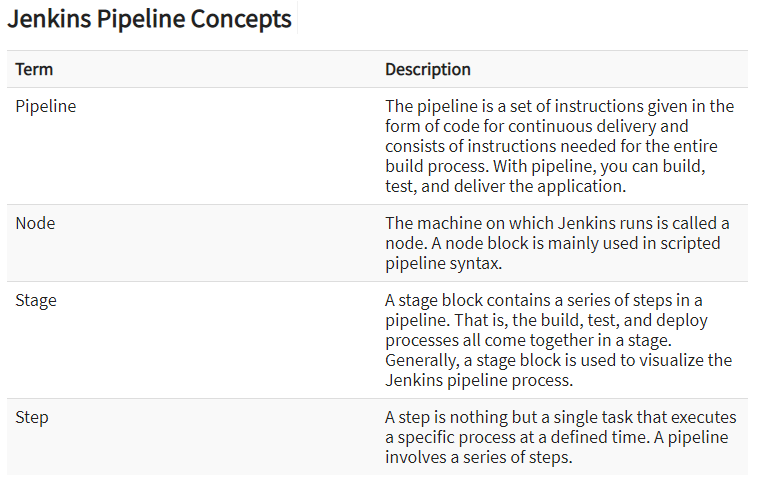


Why Use Jenkin's Pipeline?

Jenkins is an open continuous integration server which has the ability to support the automation of software development processes. You can create multiple automation jobs with the help of use cases and run them as a Jenkins pipeline.

Here are the reasons why you use should use Jenkins pipeline:

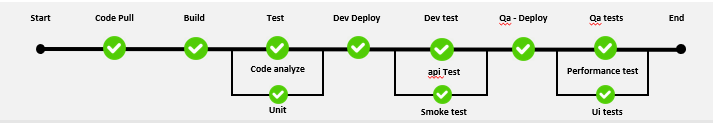
* Jenkins pipeline is implemented as a code which allows multiple users to edit and execute the pipeline process.
* Pipelines are robust. So, if your server undergoes an unforeseen restart, the pipeline will be automatically resumed.
* You can pause the pipeline process and make it wait to resume until there is an input from the user.
* Jenkins Pipelines support big projects. You can run multiple jobs, and even use pipelines in a loop.



Reference URL:

<https://jenkins.io/doc/book/pipeline/getting-started/>

If you want to create pipelines by writing the Jenkinsfile then we need to write Jenkinsfile in Groovy or by using the Blue Ocean plugin



For this Pipeline we need to download blue ocean plugin and configure the Pipeline.

<https://jenkins.io/doc/tutorials/create-a-pipeline-in-blue-ocean/>