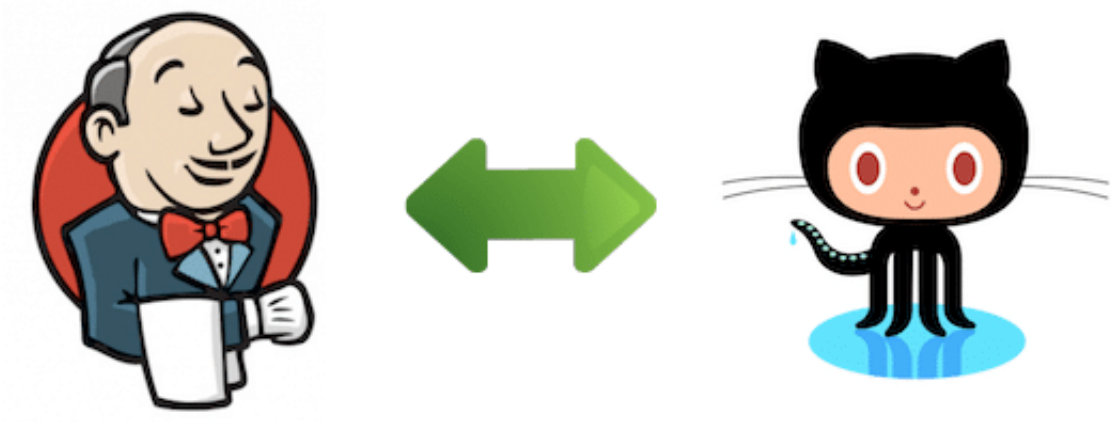


Automatically triggering a Jenkins Build on every Code Push Event

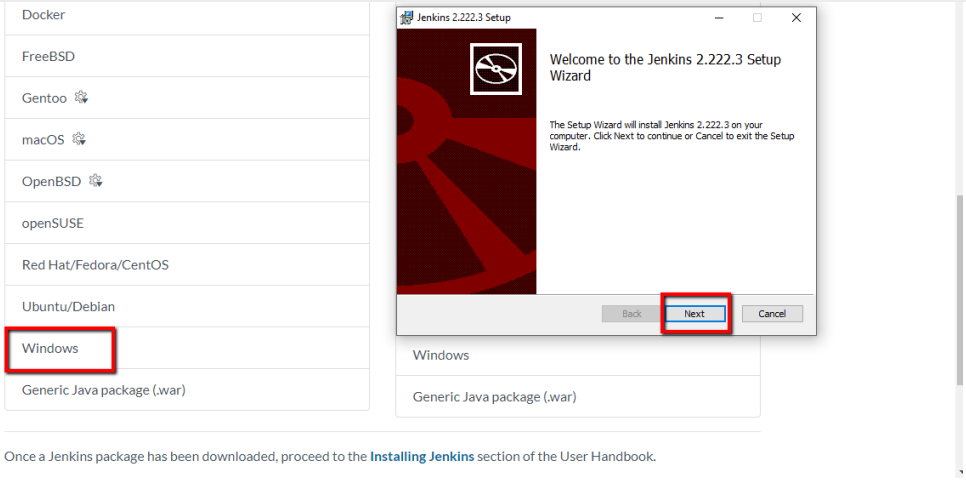
Published on May 2, 2020

Srikant Pandey
Senior Quality Engineer at TO THE NEW
3 articles + Follow

Objective of writing this article is to give you idea how we can automatically run the build on every code push. But to understand this concept first we need to understand the integration of a CI/CD tool i.e Jenkins with Source Control Management tool. Here I am using GitHub as source control management tool.

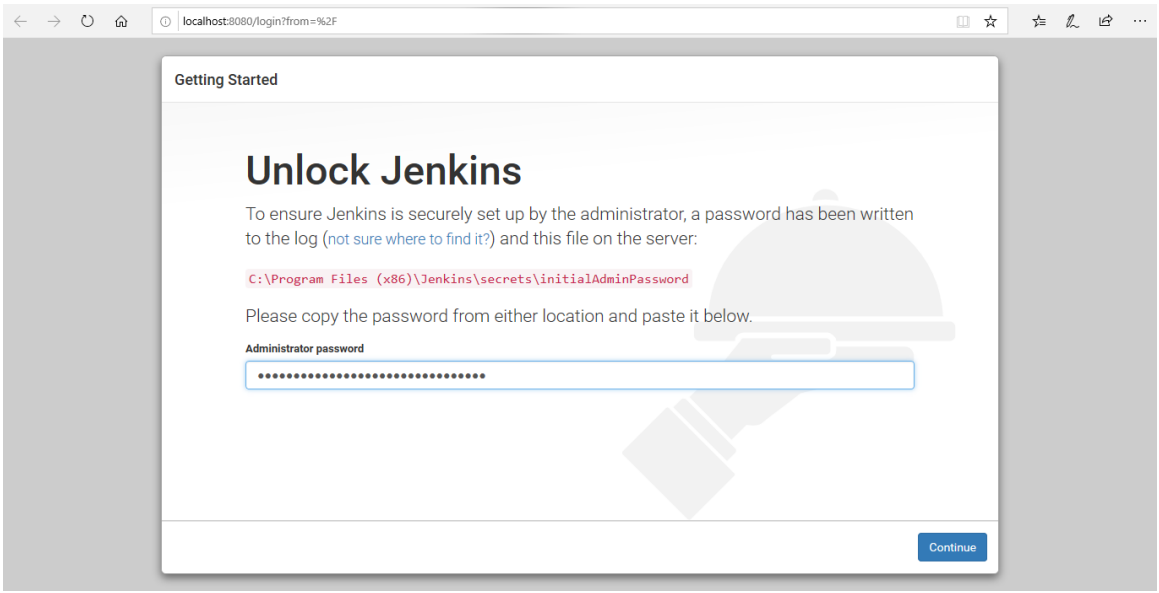
Step 1: Jenkins Installation

- Navigate to **https://www.jenkins.io/download/** and download the **jenkins.msi** as per your platform.
- Unzip **jenkins.msi** and then install Jenkins where you want to have the Jenkins instance.



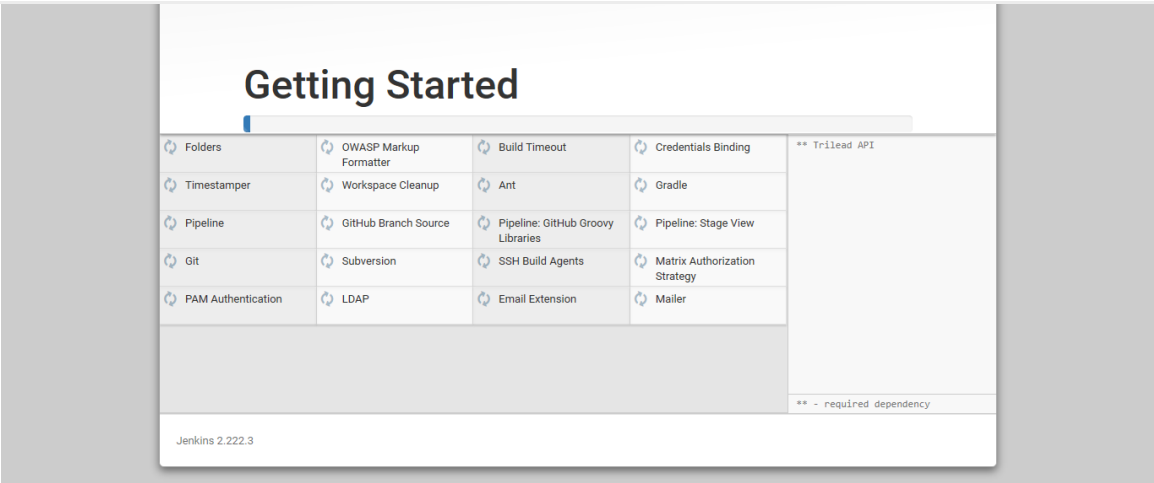
Step 2: Configure Jenkins

Once the installation process is completed then a browser tab will pop-up asking for the initial Administrator password.

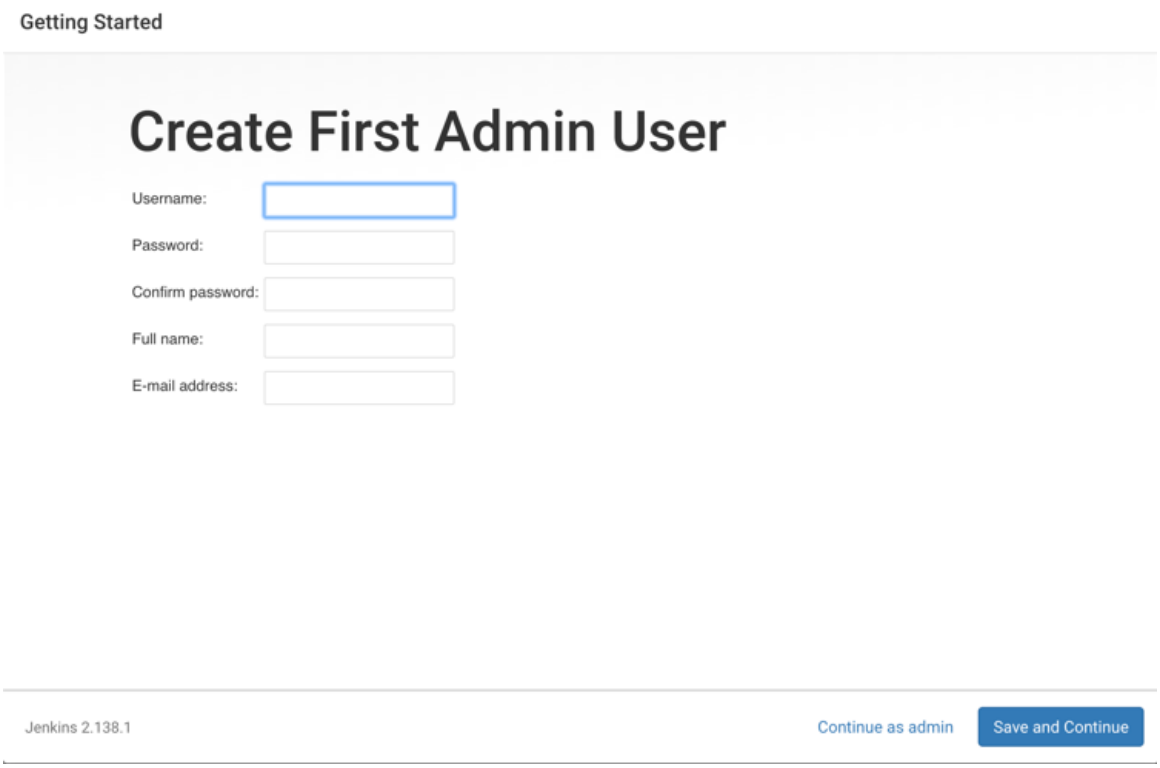


Copy the password from the **initialAdminPassword** file then paste password into browser's pop-up tab to unlock the Jenkins.

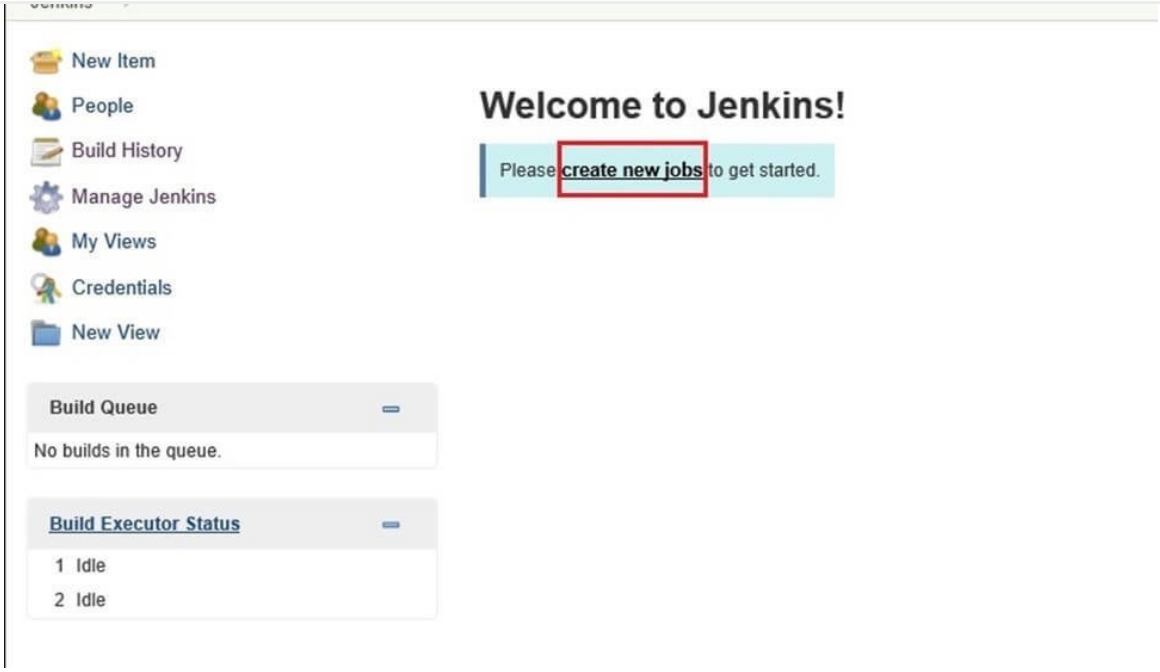
After that click on the **Install suggested plugins button** so Jenkins will retrieve and install the essential plugins needed to create new Jenkins Jobs.



Once all suggested plugins will be installed, the "Create First Admin User" panel will show up. Fill all the fields with desired account details and hit the "**Save and Continue**" button.

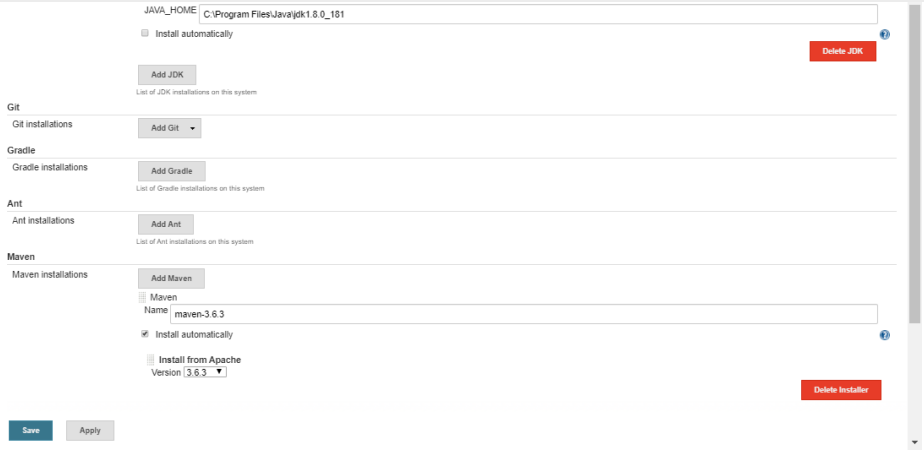


Once the Jenkins instance is up and running you can check it by navigating to **http://localhost:8080/**



Step 3: Configure JDK, Maven & Email Notification in Jenkins

- Go to Jenkins Dashboard >>Manage Jenkins >>Manage plugins >>Available >>Install **Maven Integration Plugin**.
- Go to Manage Jenkins>>Global tool configuration>>Maven >>Add **MAVEN_HOME** variable value (i.e. path of the maven file on your system).
- Once you will add **MAVEN_HOME** then in a similar way add **JAVA_HOME** variable value.



- Go to Jenkins Dashboard >>Manage Jenkins >>Manage plugins >>Available >>Install **Email Extension Plugin**

- Go to Manage Jenkins>>Configure System>>Set the SMTP settings in **Extended E-mail Notification** and **E-mail Notification** and apply the changes.

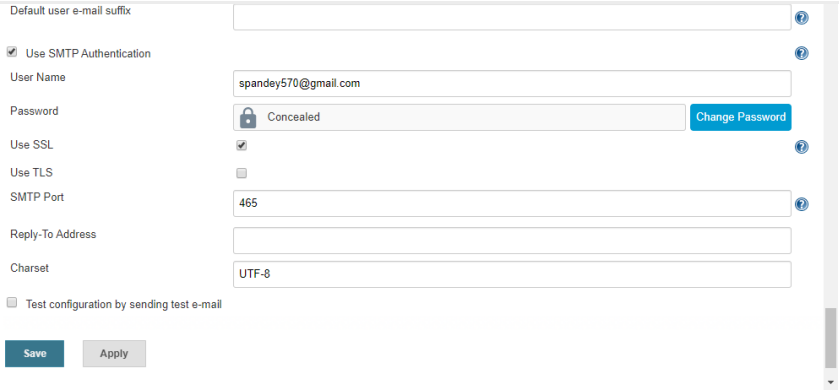
SMTP Server Name : smtp.gmail.com

Username: test_email_id@gmail.com

Password: test_password

Use SSL : Checked

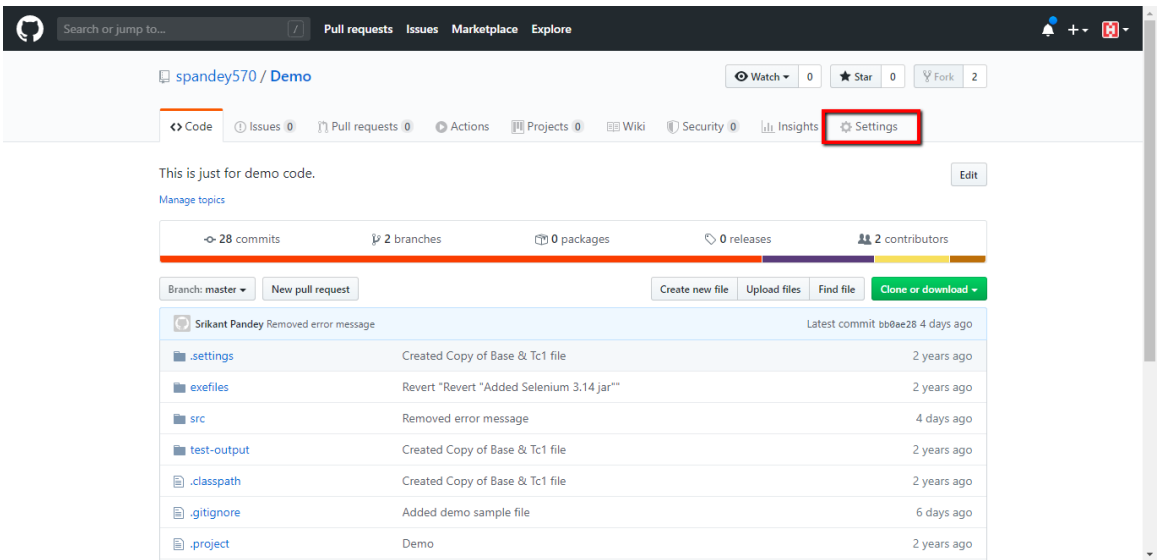
SMTP Port: 456

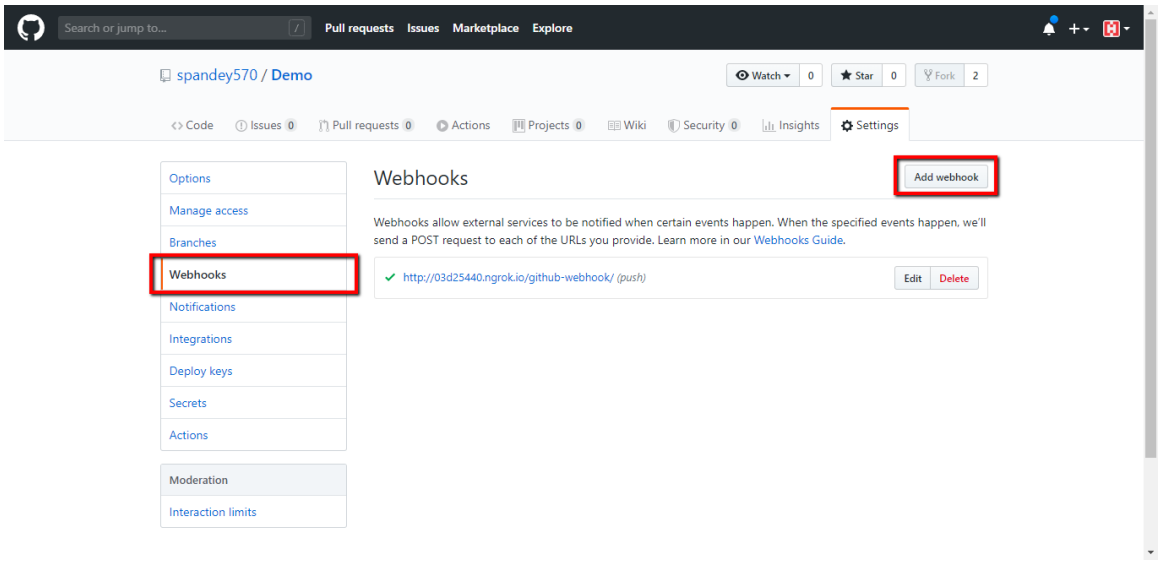


Step 4: Configure GitHub

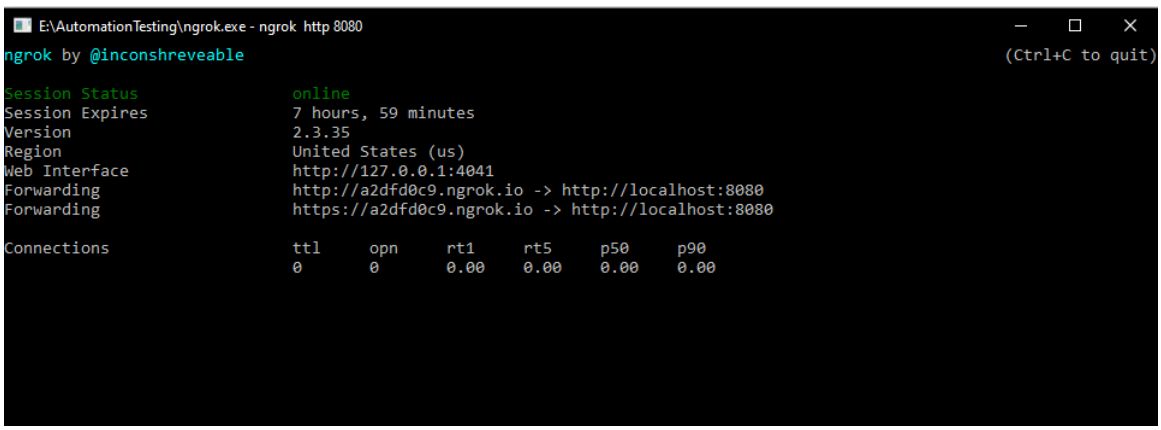
Once the Jenkins is ready to use then configure GitHub by adding GitHub webhook in Jenkins. But before that we need to understand the concept of webhook. A webhook is an HTTP callback, an HTTP POST that occurs when something happens through a simple event-notification via HTTP POST. Webhook will send a POST request to the requested Payload URL below with details of subscribed events. GitHub webhooks in Jenkins are used to trigger the build whenever a developer commits something to the master branch. Here I am using my Demo repository to demonstrate the entire process.

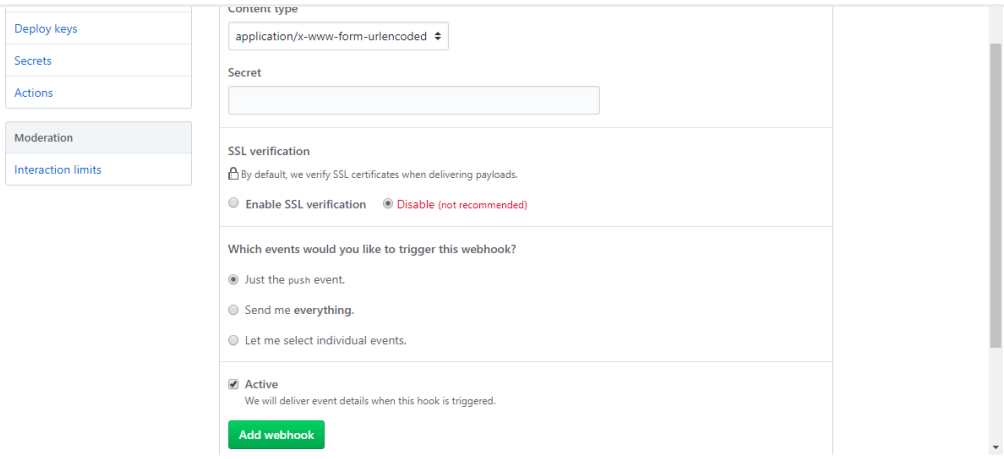
- Go to your GitHub repository and click on **'Settings'**





- In the **Payload URL** field, paste your Jenkins environment URL and at the end of the URL add /github-webhook/. But as your Jenkins is running on localhost then writing https://localhost:8080/github-webhook/ will not work because Webhooks can only work when they are exposed to the internet. So if you want to make your localhost:8080 expose to the internet then we can use **https://ngrok.com/download** & then it will looks like: https://a2dfd0c9.ngrok.io/github-webhook/

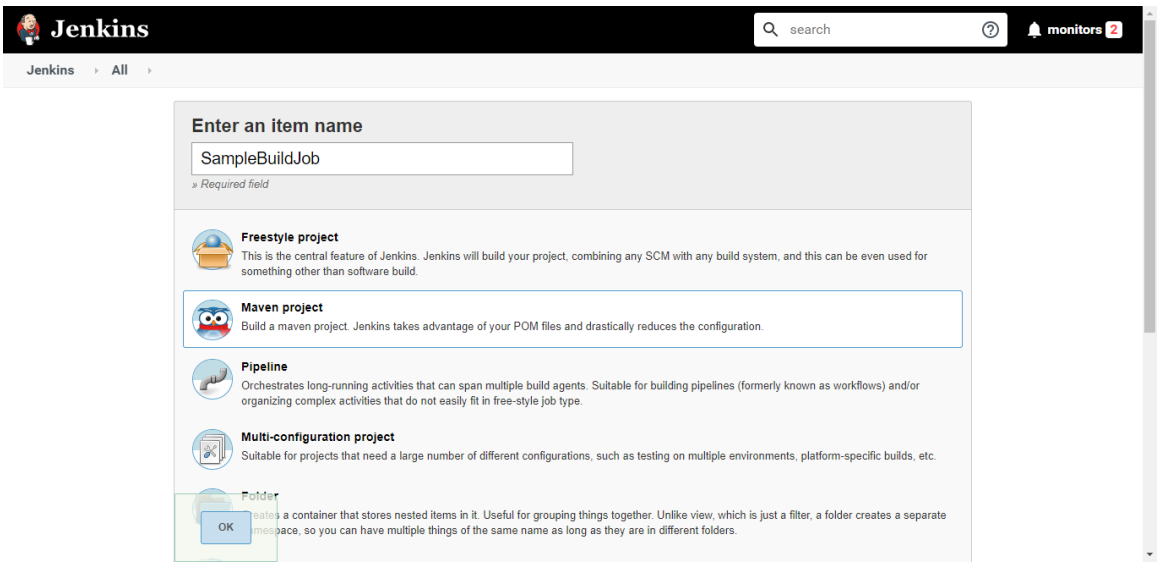




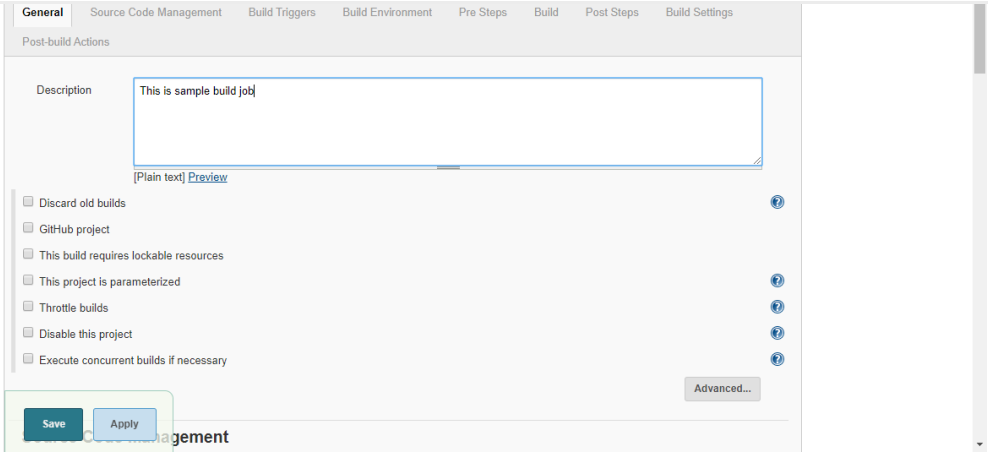
Then select option **Just the push event** as the trigger point for this webhook and check **Active** option so that event details will be delivered when this hook is triggered.

Step 5: Configure a sample build job in Jenkins

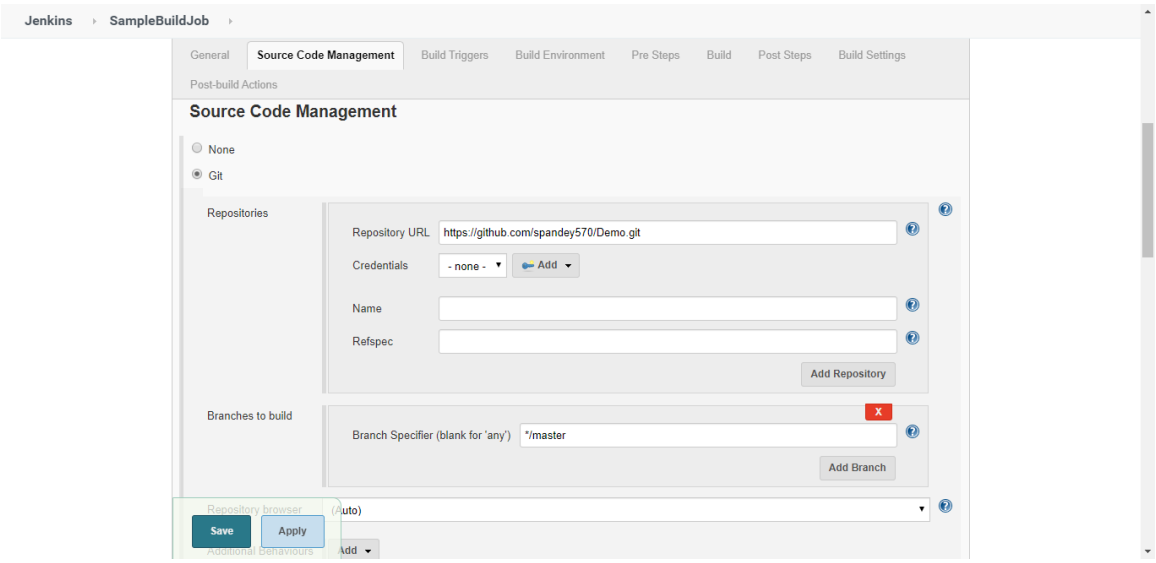
- Go to Jenkins Dashboard >> New Item >> Choose name for the Maven Project for example **SampleBuildJob**



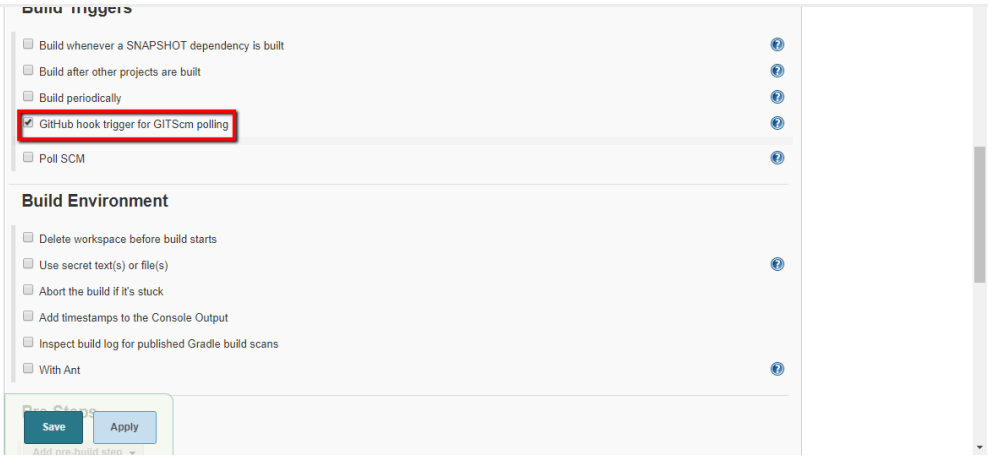
- Set the general description for the job.



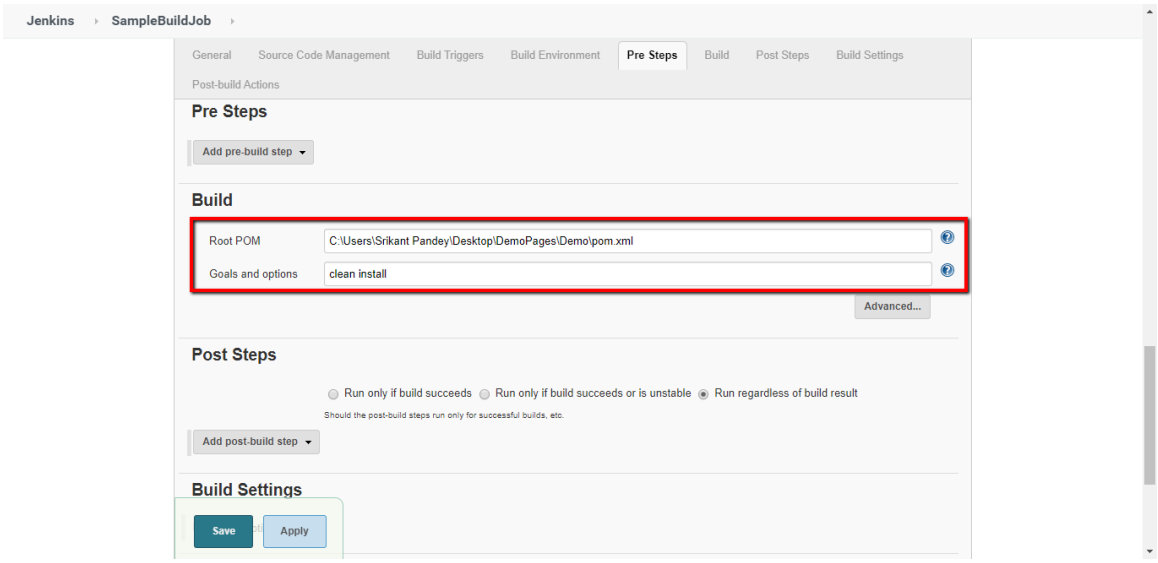
- Select the preferred source code management tool, here I am using Git as Source Code management tool, So provide the required details like repo url, branch to build.



- Select **GitHub hook trigger for GITScm polling** option under Build Triggers as we want to trigger the job on every code push event.



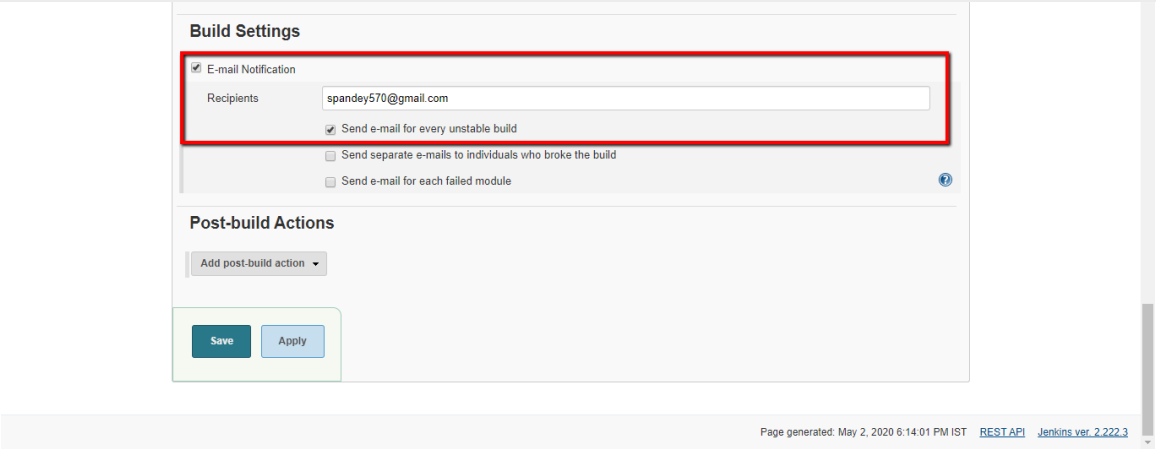
- Provide the pom.xml path & maven command to execute the build.



- Select **Run regardless of build result** option for Post build step as we want to receive email after every job.

- Configure recipients email under build settings options so that post job execution email notification will be sent to the configured recipients.

- Click on **Apply** button to save the details.



Now update the code in your git repository then we will see how Jenkins ran the script after every code push event and send the email notification after the job execution to the requested recipients email id.

Published by

Srikant Pandey
Senior Quality Engineer at TO THE NEW
Published • 1y

Hello Connections,
This time I am sharing article on how we can automatically run the build on every code push. This article will help you to understand the one of the basic steps of implementing CI/CD Pipeline. Please go through it for the reference.
#qaengineer #testautomation #framework #jenkins #github #integrations #cicd #keeplearning

Add a comment...

Aymen Ferjaoui • 3rd+ 1y
DevOps engineer at VERMEG for Banking & Insurance Software

Well done Srikant, but the title is automating the build. Here you just configuring jenkins.
A complete build automation process must include several steps : compilation, running tests and generating reports for developers, quality assurance , generating executable...

Like · 1 | Reply · 1 Reply

Srikant Pandey • 3rd+ 1y
Senior Quality Engineer at TO THE NEW

Thanks Aymen Ferjaoui for the inputs.

Like | Reply

Maureen Ononiwu • 3rd+ 4mo
Chemical Engineer(in view)||Growing in tech (Django x devops)||Content creator @NschE-Futo chapter (Linkedin page)

Thanks sir

Like · 1 | Reply

Srikant Pandey
Senior Quality Engineer at TO THE NEW

+ Follow

More from Srikant Pandey

Cucumber Implementation Guide

Srikant Pandey on LinkedIn

Hurdles faced by experienced candidates nowadays while switching their job

Srikant Pandey on LinkedIn