



Home



My Network



Jobs



Automatically triggering a Jenkins Build on every Code Push Event

Published on May 2, 2020



Srikant Pandey

Senior Quality Engineer at TO THE NEW

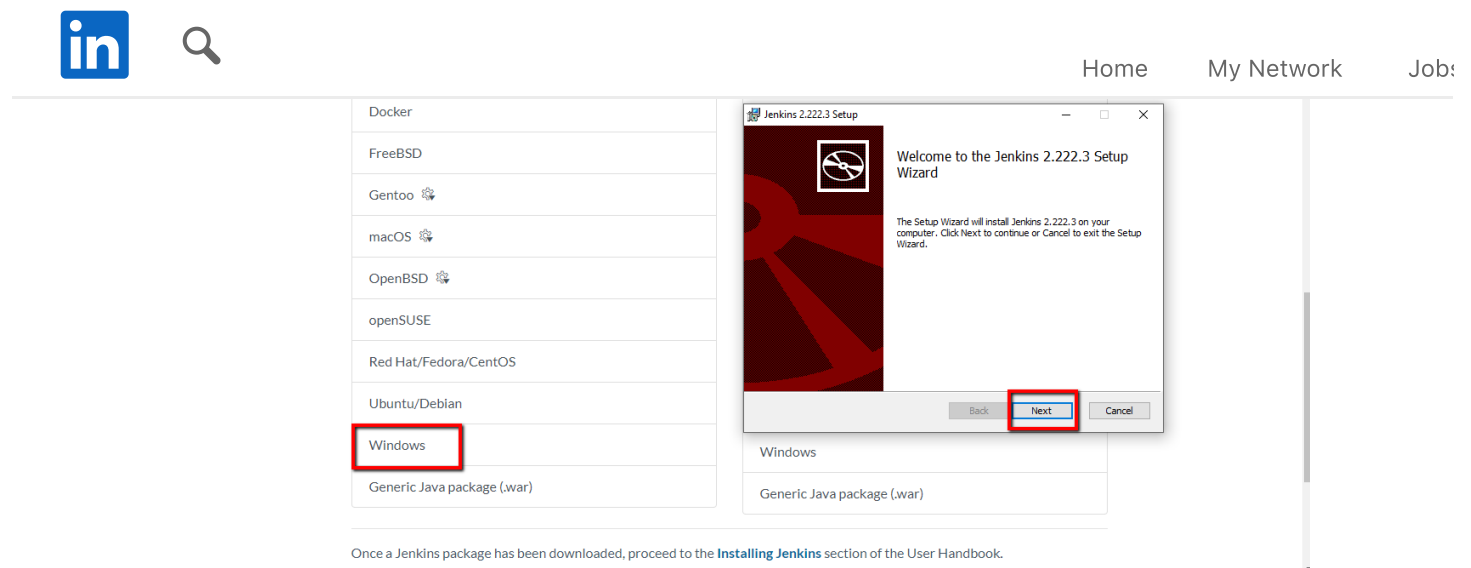
3 articles

+ 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.

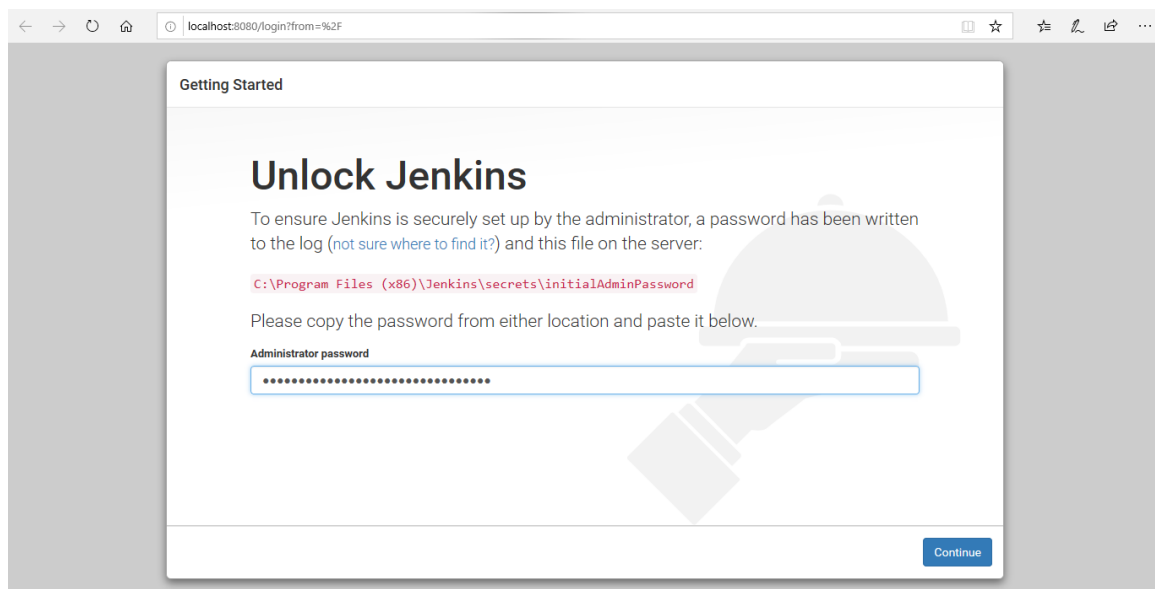
Step 1: Jenkins Installation

- 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.



Getting Started

Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding	** Trilead API
Timestamper	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	Subversion	SSH Build Agents	Matrix Authorization Strategy	
PAM Authentication	LDAP	Email Extension	Mailer	
				<small>** - required dependency</small>

Jenkins 2.222.3

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.

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

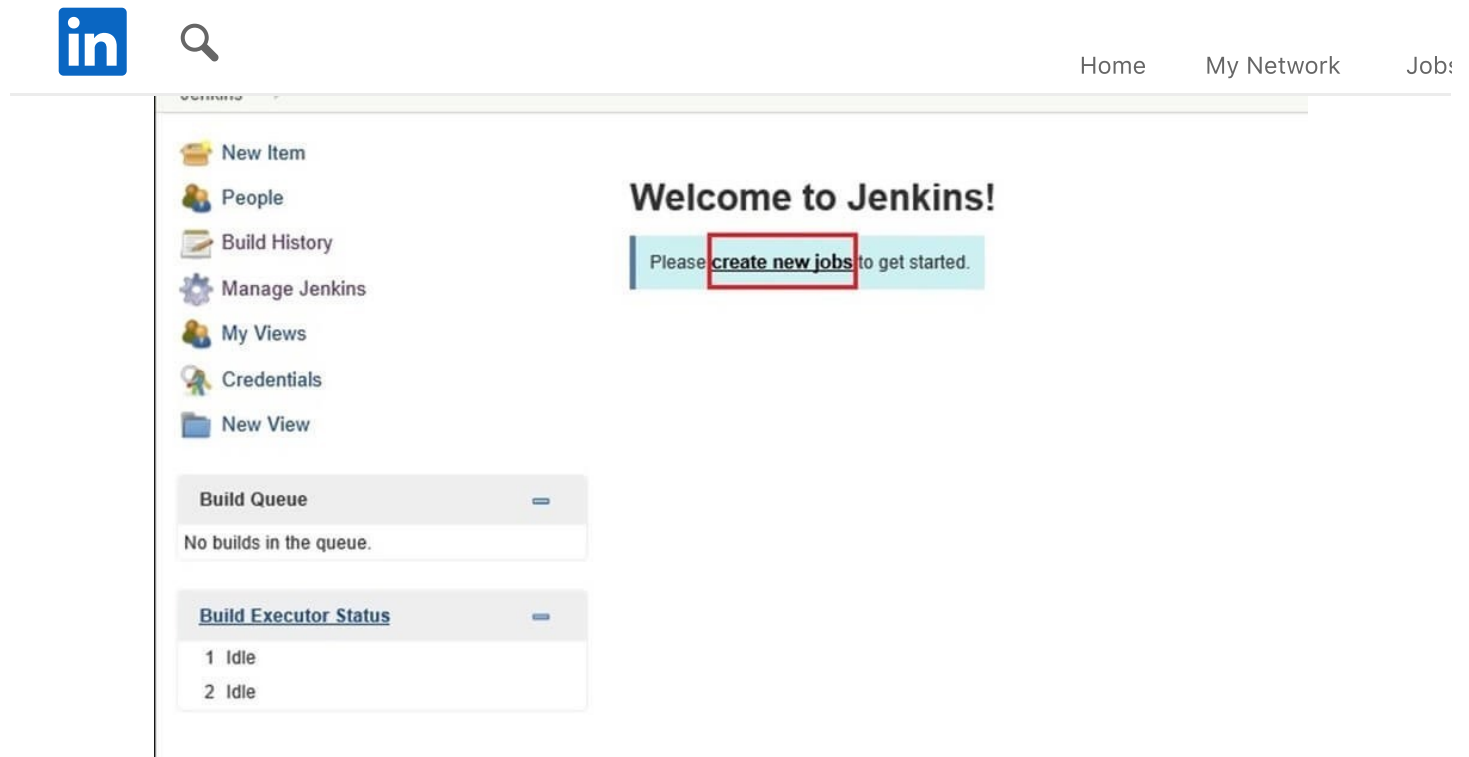
Full name:

E-mail address:

Jenkins 2.138.1

[Continue as admin](#)[Save and Continue](#)

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.

in 🔍

Home My Network Jobs

JAVA_HOME C:\Program Files\Java\jdk1.8.0_181
☐ Install automatically Delete JDK

Add JDK
List of JDK installations on this system

Git
Git installations Add Git

Gradle
Gradle installations Add Gradle
List of Gradle installations on this system

Ant
Ant installations Add Ant
List of Ant installations on this system

Maven
Maven installations Add Maven

Name
maven-3.6.3

☒ Install automatically Install from Apache
Version 3.6.3 Delete Installer

Save Apply

- 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



Home

My Network

Jobs

Default user e-mail suffix

☒ Use SMTP Authentication

User Name: spandey570@gmail.com

Password: Concealed [Change Password](#)

☒ Use SSL

☐ Use TLS

SMTP Port: 465

Reply-To Address:

Charset: UTF-8

☐ Test configuration by sending test e-mail

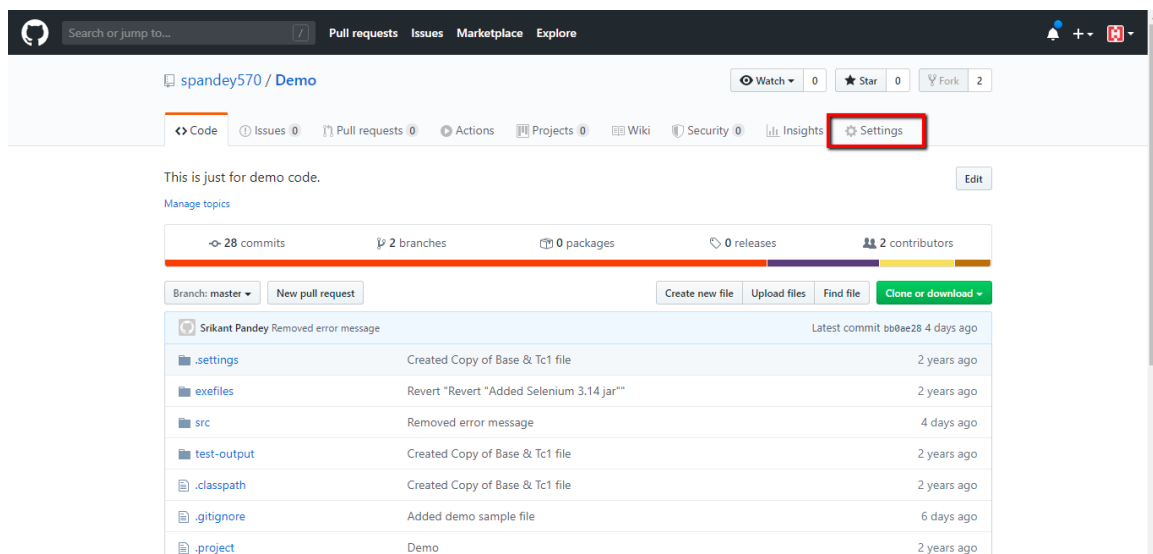
[Save](#) [Apply](#)

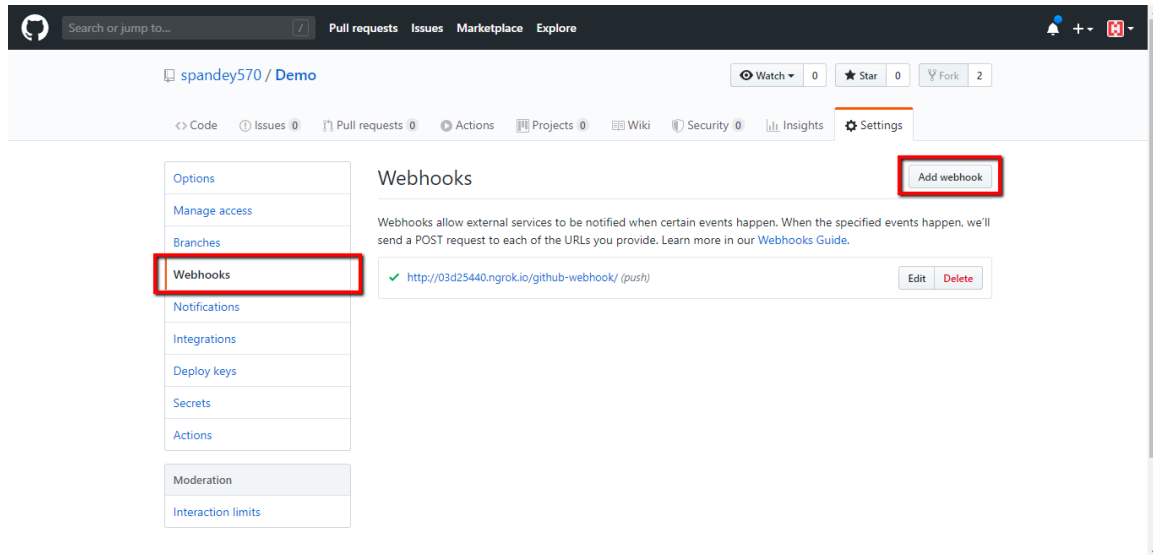
Step 4: Configure GitHub

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 look like: `https://a2dfd0c9.ngrok.io/github-webhook/`


```

E:\AutomationTesting\ngrok.exe - ngrok http 8080
ngrok by @inconshreveable

Session Status      online
Session Expires    7 hours, 59 minutes
Version            2.3.35
Region             United States (us)
Web Interface       http://127.0.0.1:4041
Forwarding          http://a2dfd0c9.ngrok.io -> http://localhost:8080
                   https://a2dfd0c9.ngrok.io -> http://localhost:8080

Connections
  ttl    opn    rt1    rt5    p50    p90
   0      0      0.00   0.00   0.00   0.00

```



Deploy keys

Secrets

Actions

Moderation

Interaction limits

Content type

application/x-www-form-urlencoded

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☐ Enable SSL verification
 ☒ Disable (not recommended)

Which events would you like to trigger this webhook?

☒ Just the push event.
 ☐ Send me everything.
 ☐ Let me select individual events.

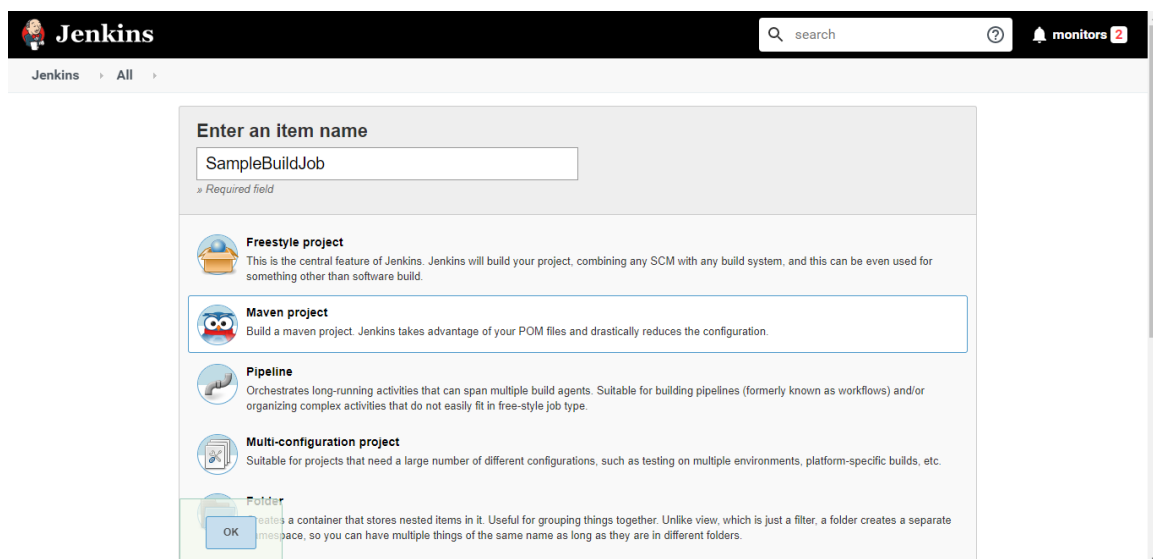
☒ Active
 We will deliver event details when this hook is triggered.

Add 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**



Jenkins

search

monitors 2

Jenkins > All >

Enter an item name

SampleBuildJob

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate mespace, so you can have multiple things of the same name as long as they are in different folders.

OK

- Set the general description for the job.



Home

My Network

Jobs

The image shows the 'General' tab of a Jenkins job configuration. The 'Description' field contains the text 'This is sample build job'. Below the description, there are several checkboxes: 'Discard old builds', 'GitHub project', 'This build requires lockable resources', 'This project is parameterized', 'Throttle builds', 'Disable this project', and 'Execute concurrent builds if necessary'. At the bottom, there are 'Save' and 'Apply' buttons, and an 'Advanced...' link.

- 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.

The image shows the 'Source Code Management' tab of a Jenkins job configuration. The 'None' radio button is selected, and the 'Git' radio button is also selected. Under 'Repositories', the 'Repository URL' is set to 'https://github.com/spandey570/Demo.git'. The 'Credentials' dropdown is set to 'none'. The 'Name' and 'Refspec' fields are empty. The 'Branches to build' section shows a 'Branch Specifier (blank for \'any\')' set to '*/master'. At the bottom, there are 'Save' and 'Apply' buttons, and a 'Repository browser' dropdown set to '(Auto)'.

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



Home

My Network

Jobs

Build Triggers

- ☐ Build whenever a SNAPSHOT dependency is built
- ☐ Build after other projects are built
- ☐ Build periodically
- ☒ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published Gradle build scans
- ☐ With Ant

Save Apply

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

Jenkins > SampleBuildJob >

General Source Code Management Build Triggers Build Environment **Pre Steps** Build Post Steps Build Settings

Post-build Actions

Pre Steps

Add pre-build step

Build

Root POM	C:\Users\Srikant Pandey\Desktop\DemoPages\Demo\pom.xml
Goals and options	clean install

Advanced...

Post Steps

☐ Run only if build succeeds
 ☐ Run only if build succeeds or is unstable
 ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

Add post-build step

Build Settings

Save Apply

- 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.



Home

My Network

Jobs

Build Settings

☒ E-mail Notification

Recipients

spandey570@gmail.com

☒ Send e-mail for every unstable build☐ Send separate e-mails to individuals who broke the build☐ Send e-mail for each failed module

Post-build Actions

Add post-build action ▼

Save

Apply

Page generated: May 2, 2020 6:14:01 PM IST [REST API](#) [Jenkins ver. 2.222.3](#)

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.

[Report this](#)

Published by

**Srikant Pandey**

Senior Quality Engineer at TO THE NEW

Published • 1y

3 articles

[+ Follow](#)

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](#)

Like Comment Share

35 • 3 comments

Reactions



+23



3 Comments



Home

My Network

Jobs



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

