newt|global
Aiming At Customer Delight

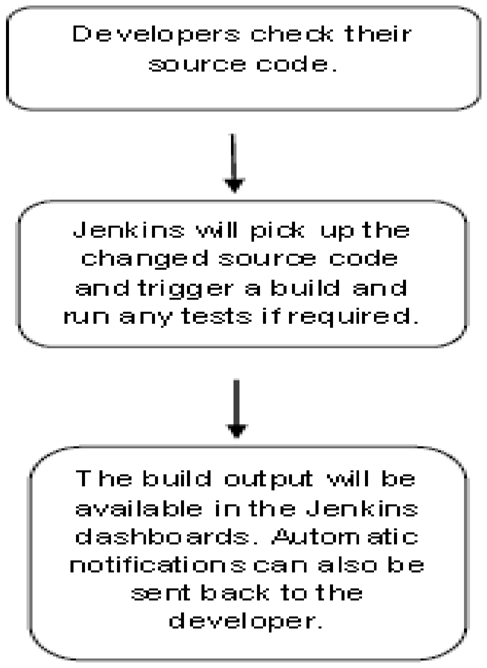
Created by : Mahender Muthyala
Date : 08/May/2017

*Using Jenkins to build **automatically** trigger **an Application** to push into Cloud Foundry*

1. Introduction

Jenkins:

Jenkins is an open-source **continuous** integration software tool written in the Java programming language for testing and reporting on isolated changes in a larger code base in real time. The software enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.


Developers check their source code.
Jenkins will pick up the changed source code and trigger a build and run any tests if required.
The build output will be available in the Jenkins dashboards. Automatic notifications can also be sent back to the developer.

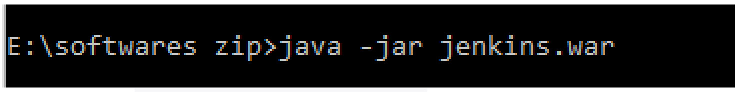
Jenkins – Installation

Download Jenkins

The official website for Jenkins is Jenkins. If you click the given link, you can get the page of the Jenkins official website and select the Jenkins version.

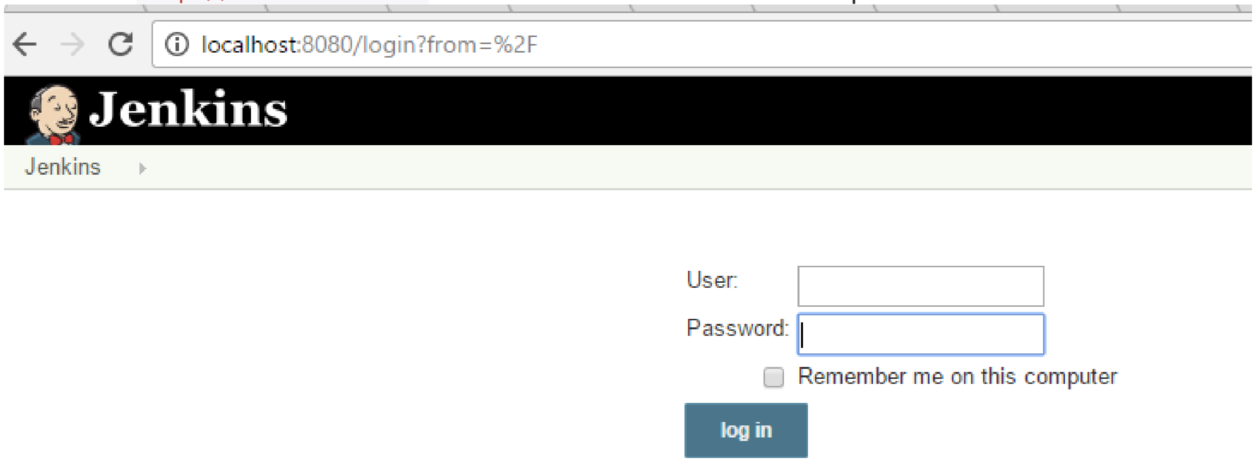
https://updates.jenkins-ci.org/download/war/

1. Open up a terminal in the download directory and run `java -jar jenkins.war`



```
E:\softwares zip>java -jar jenkins.war
```

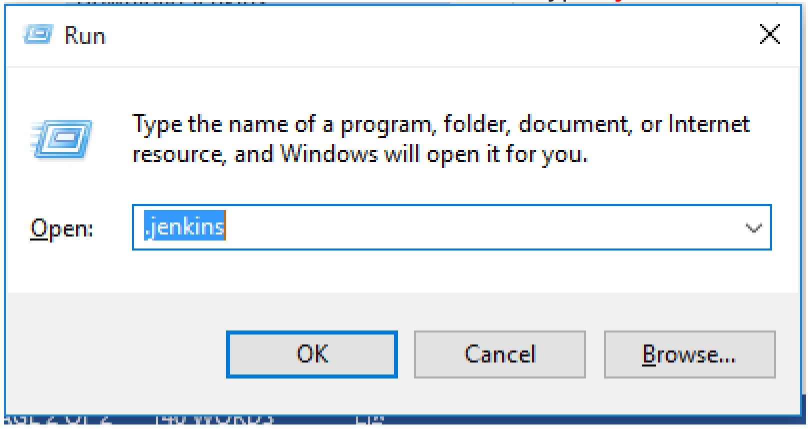
2. Browse to `http://localhost:8080` and follow the instructions to complete the installation



localhost:8080/login?from=%2F

3. Enter the user name and password. User Name is admin.

Click the windows button → select **RUN** → type **.jenkins**

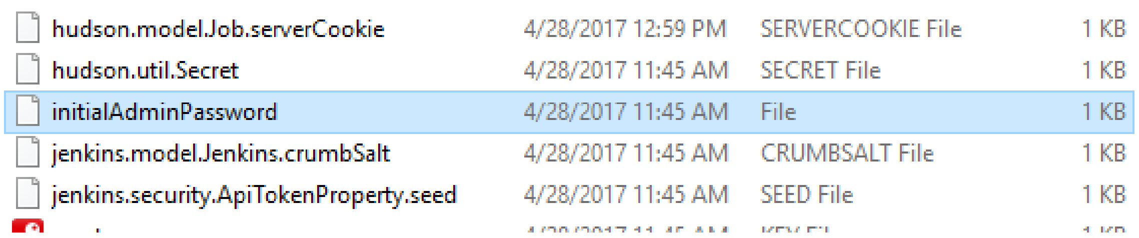


Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

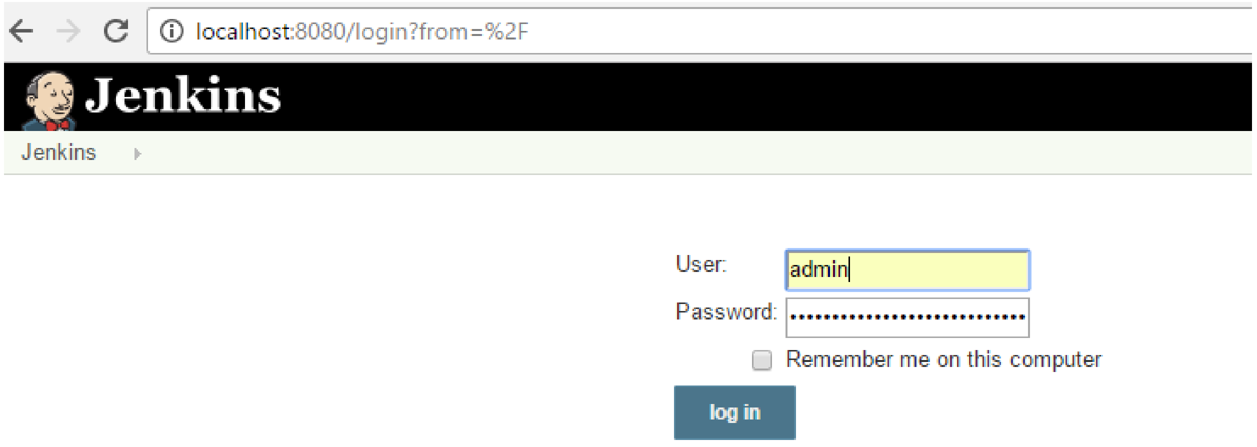
Open: .jenkins

Click the OK Button. And select **secrets** file and select the initialAdminPassword text file.



hudson.model.Job.serverCookie	4/28/2017 12:59 PM	SERVERCOOKIE File	1 KB
hudson.util.Secret	4/28/2017 11:45 AM	SECRET File	1 KB
initialAdminPassword	4/28/2017 11:45 AM	File	1 KB
jenkins.model.Jenkins.crumbSalt	4/28/2017 11:45 AM	CRUMBSALT File	1 KB
jenkins.security.ApiTokenProperty.seed	4/28/2017 11:45 AM	SEED File	1 KB

And open the initialAdminPassword text file in Notepad copy the line and paste in Jenkins password box.



localhost:8080/login?from=%2F

Jenkins

Jenkins

User: admin

Password: ••••••••••••••••••••••••••••••••

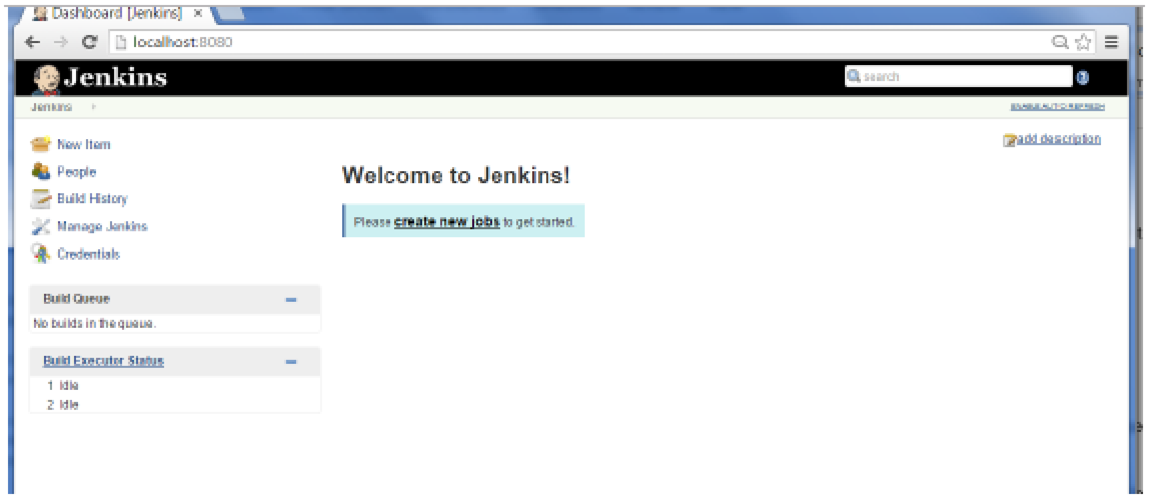
Remember me on this computer

log in

Accessing Jenkins

Once Jenkins is up and running, one can access Jenkins from the link – **http://localhost:8080**

This link will bring up the Jenkins dashboard.

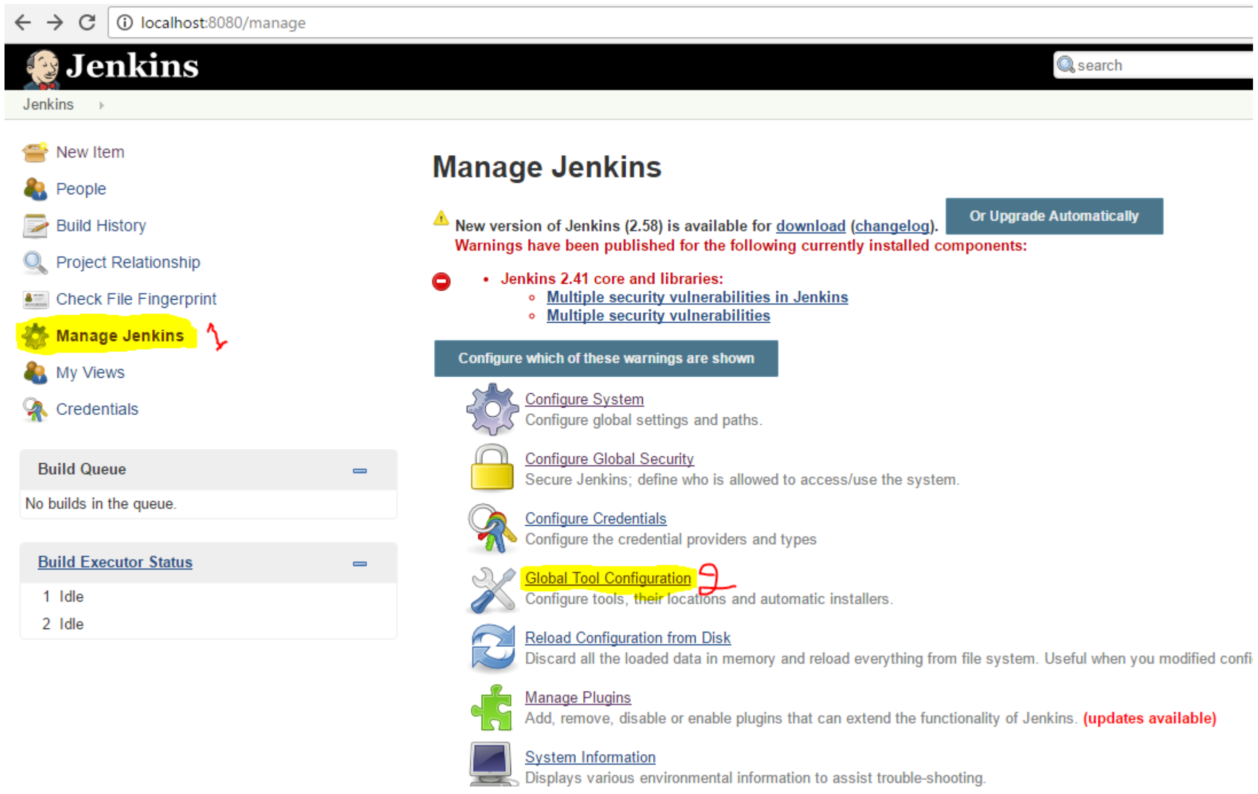


Dashboard [Jenkins]
localhost:8080
Jenkins
Jenkins
ENABLE AUTO REFRESH
New Item
People
Build History
Manage Jenkins
Credentials
add description
Welcome to Jenkins!
Please create new jobs to get started.
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

Jenkins – JDk,ANT Setup

Downloading and Setting Up JDK

Before using Jenkins to build Java applications, you need to configure the location or it where your JDK installation is. Select *Manage Jenkins* and afterwards *Global Tool Configuration*.



localhost:8080/manage
Jenkins
Jenkins
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Credentials
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

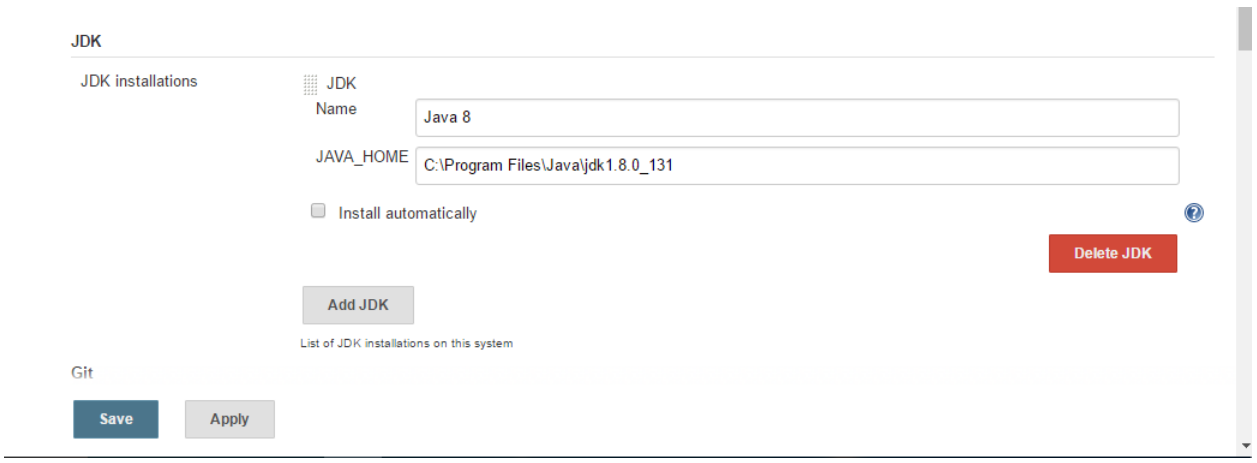
Manage Jenkins

New version of Jenkins (2.58) is available for download (changelog).
Or Upgrade Automatically
Warnings have been published for the following currently installed components:
- Jenkins 2.41 core and libraries:
  - Multiple security vulnerabilities in Jenkins
  - Multiple security vulnerabilities

Configure which of these warnings are shown

Configure System
Configure global settings and paths.
Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.
Configure Credentials
Configure the credential providers and types
Global Tool Configuration
Configure tools, their locations and automatic installers.
Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified confi
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. (updates available)
System Information
Displays various environmental information to assist trouble-shooting.

In the next screen you can enter the correct path to your JDK and press the save Button.Jenkins can also install these for your automatically.


JDK

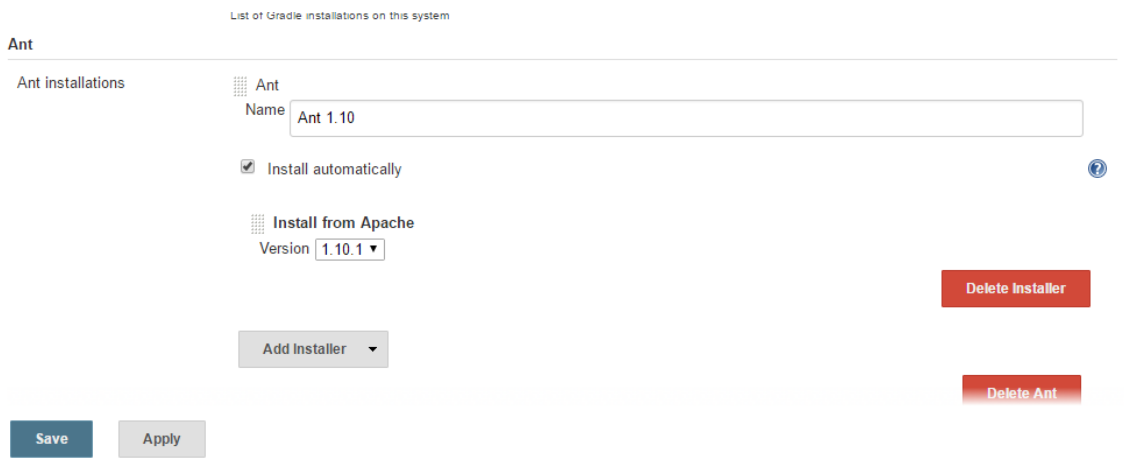
JDK installations

JDK
Name: Java 8
JAVA_HOME: C:\Program Files\Java\jdk1.8.0_131
[ ] Install automatically

List of JDK installations on this system

Git

After we select the *Ant* and Enter the *Ant Name* and select the *Instal automatically* and select the *version* then click the *save* button. Jenkins can also install these for your automatically.


List of Gradle installations on this system

Ant

Ant installations

Ant
Name: Ant 1.10
[✓] Install automatically

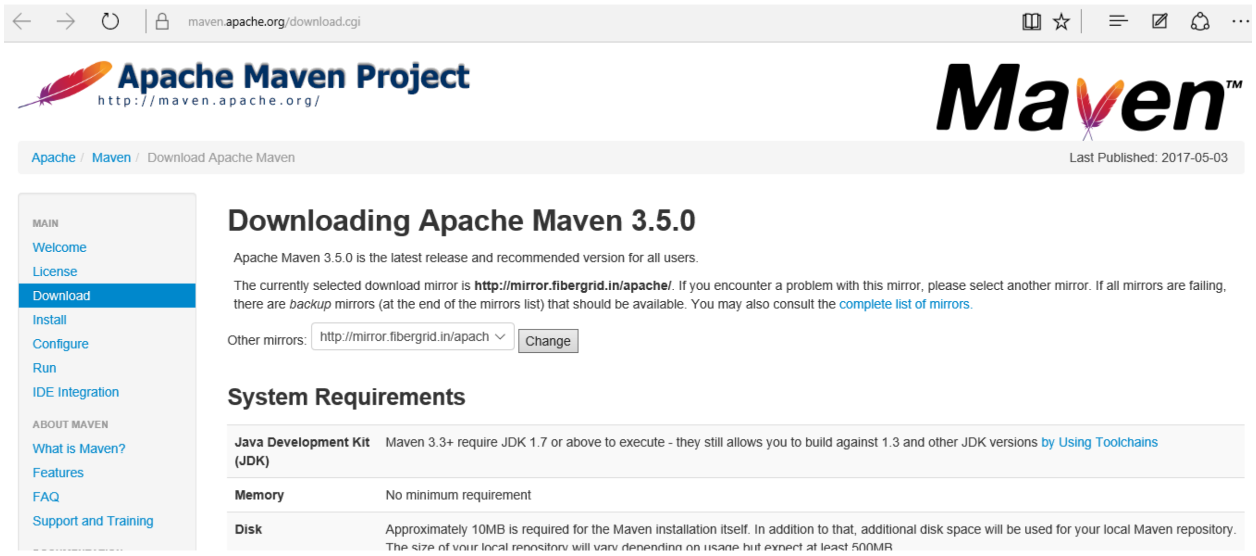
Install from Apache
Version: 1.10.1

Jenkins – Maven Setup

Downloading and Setting Up Maven

The official website for maven is Apache Maven. If you click the given link, https://maven.apache.org/download.cgi

you can get the home page of the maven official website as shown below.


maven.apache.org/download.cgi

Apache Maven Project
http://maven.apache.org/

Apache / Maven / Download Apache Maven Last Published: 2017-05-03

MAIN
Welcome
License
Download
Install
Configure
Run
IDE Integration
ABOUT MAVEN
What is Maven?
Features
FAQ
Support and Training

Downloading Apache Maven 3.5.0

Apache Maven 3.5.0 is the latest release and recommended version for all users.

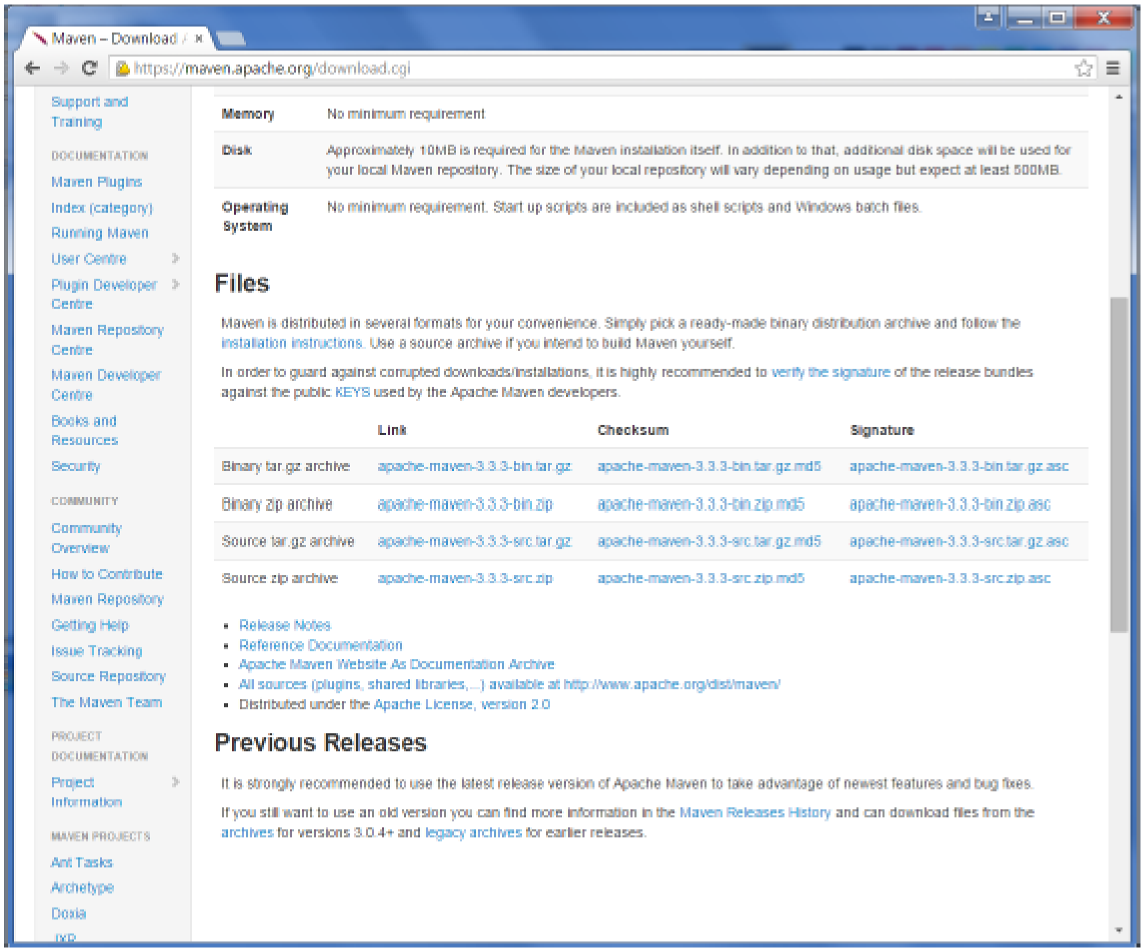
The currently selected download mirror is **http://mirror.fibergrid.in/apache/**. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available. You may also consult the complete list of mirrors.

Other mirrors: http://mirror.fibergrid.in/apach Change

System Requirements

Java Development Kit (JDK)	Maven 3.3+ require JDK 1.7 or above to execute - they still allows you to build against 1.3 and other JDK versions by Using Toolchains
Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB

While browsing to the site, go to the Files section and download the link to the Binary.zip file.



Memory	No minimum requirement
Disk	Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.
Operating System	No minimum requirement. Start up scripts are included as shell scripts and Windows batch files.

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksum	Signature
Binary tar.gz archive	apache-maven-3.3.3-bin.tar.gz	apache-maven-3.3.3-bin.tar.gz.md5	apache-maven-3.3.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.3.3-bin.zip	apache-maven-3.3.3-bin.zip.md5	apache-maven-3.3.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.3.3-src.tar.gz	apache-maven-3.3.3-src.tar.gz.md5	apache-maven-3.3.3-src.tar.gz.asc
Source zip archive	apache-maven-3.3.3-src.zip	apache-maven-3.3.3-src.zip.md5	apache-maven-3.3.3-src.zip.asc

- Release Notes
- Reference Documentation
- Apache Maven Website As Documentation Archive
- All sources (plugins, shared libraries,...) available at http://www.apache.org/dist/maven/
- Distributed under the Apache License, version 2.0

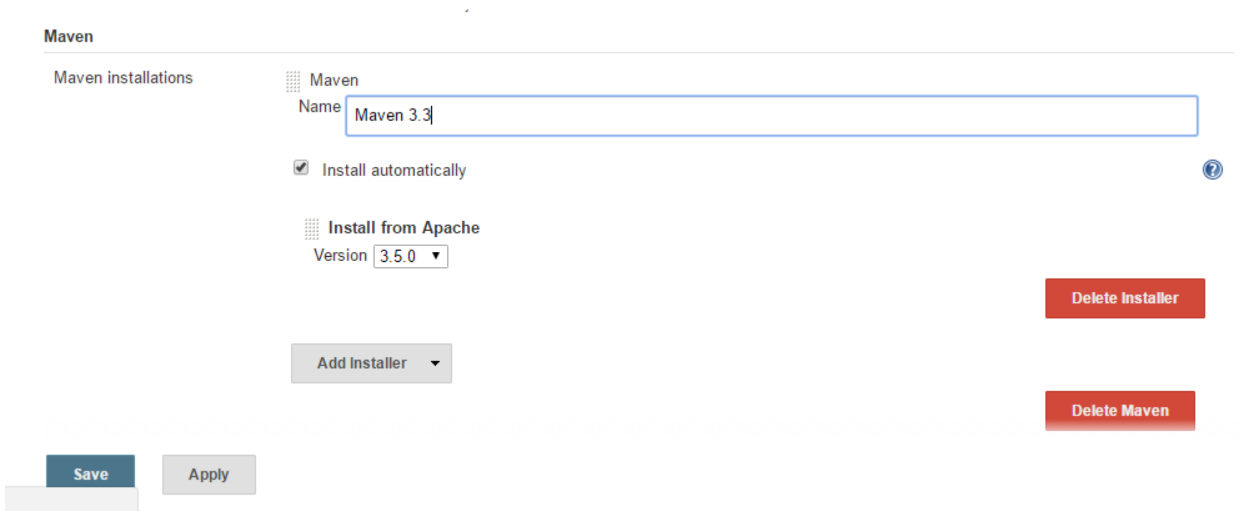
Previous Releases

It is strongly recommended to use the latest release version of Apache Maven to take advantage of newest features and bug fixes.

If you still want to use an old version you can find more information in the Maven Releases History and can download files from the archives for versions 3.0.4+ and legacy archives for earlier releases.

Once the file is downloaded, extract the files to the relevant application folder. For this purpose, the maven files will be placed in E:\Apps\apache-maven-3.3.3(any advanced vesrion).

After we select the *Maven* and Enter the *Maven Name* and select the *Instal automatically* and select the *version* then click the *save* button. Jenkins can also install these for your automatically.


Maven

Maven installations

Maven
Name: Maven 3.3

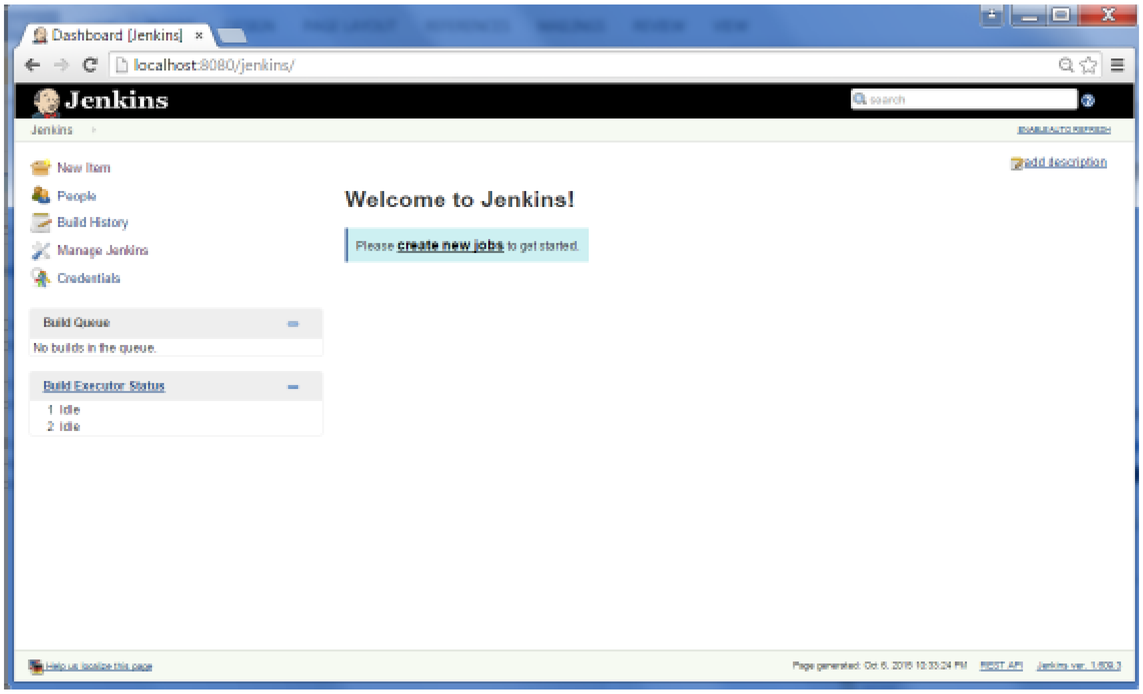
Install automatically

Install from Apache
Version: 3.5.0

Jenkins - Git Setup

For this exercise, you have to ensure that Internet connectivity is present from the machine on which Jenkins is installed.

In your Jenkins Dashboard (Home screen), click the Manage Jenkins option on the left hand side.



Welcome to Jenkins!

Please create new jobs to get started.

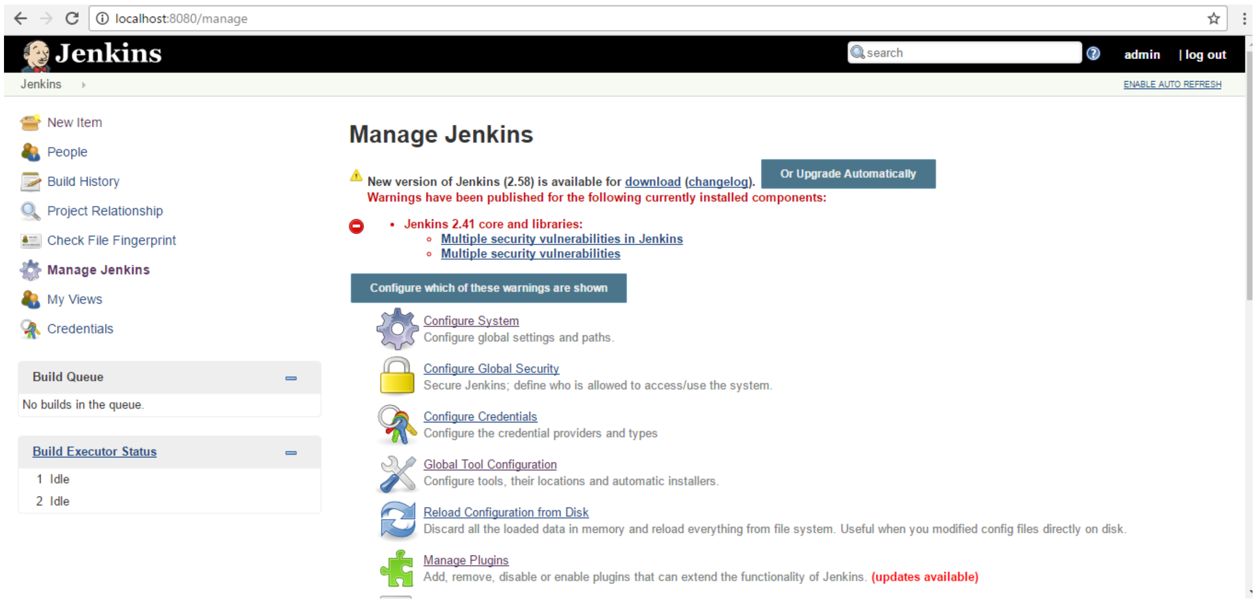
No builds in the queue.

1 Idle
2 Idle

Help us localize this page

Page generated: Oct 6, 2015 10:33:24 PM REST API Jenkins ver. 1.609.3

In the next screen, click the ‘Manage Plugins’ option.



localhost:8080/manage

Jenkins — search — admin | log out

Jenkins › ENABLE AUTO REFRESH

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Credentials

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Manage Jenkins

New version of Jenkins (2.58) is available for download (changelog). Or Upgrade Automatically
Warnings have been published for the following currently installed components:

- Jenkins 2.41 core and libraries:
  - Multiple security vulnerabilities in Jenkins
  - Multiple security vulnerabilities

Configure which of these warnings are shown

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

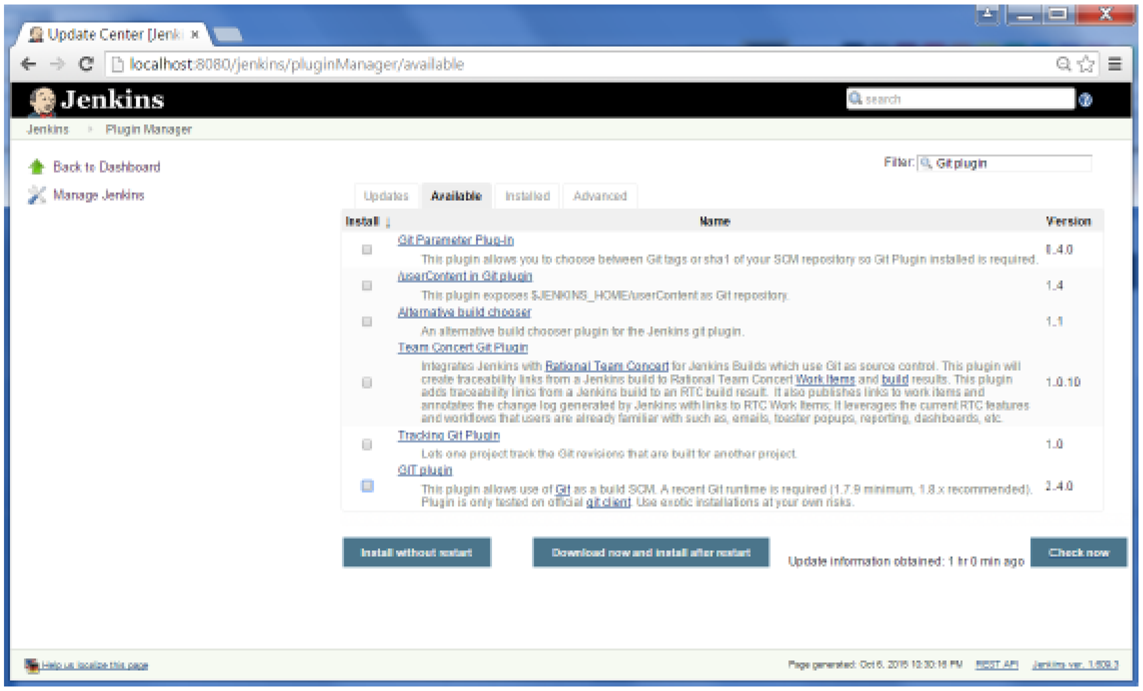
Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. (updates available)

In the next screen, click the Manage Plugins. This tab will give a list of plugins which are available for downloading. In the ‘Filter’ tab type ‘Git plugin’ and what are plugins required you have to search and select the plugins.



Update Center [Jenk

localhost:8080/jenkins/pluginManager/available

Jenkins — search

Jenkins › Plugin Manager

- Back to Dashboard
- Manage Jenkins

Filter: Git plugin

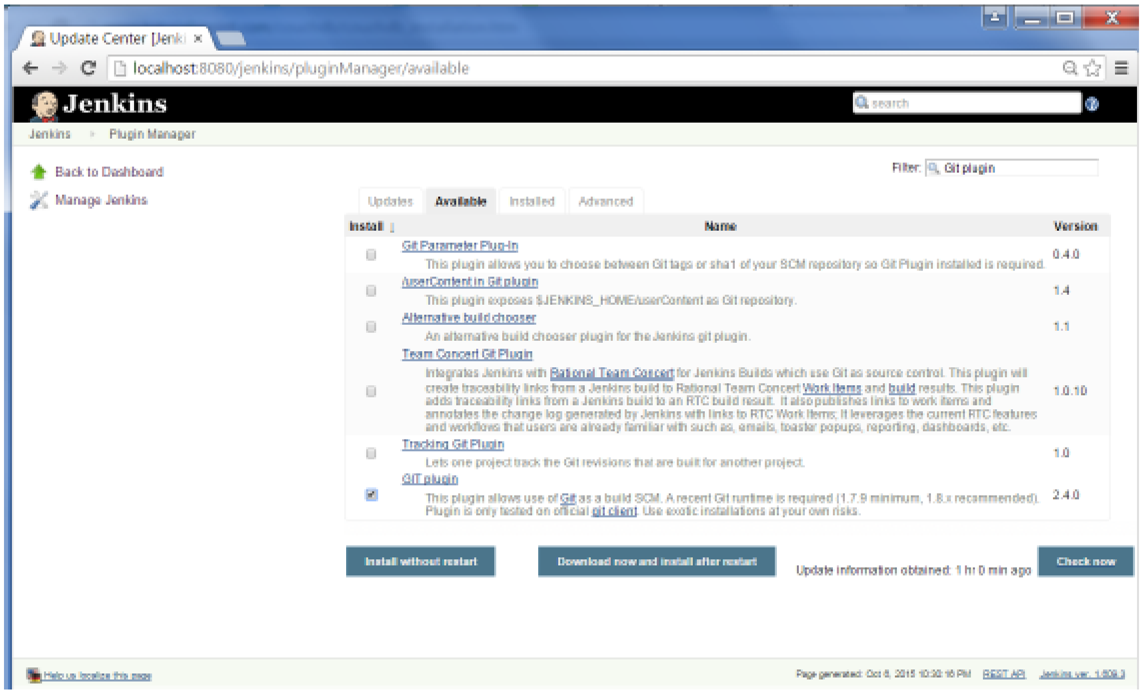
Updates | Available | Installed | Advanced

Install ↓	Name	Version
☐	Git Parameter Plug-In This plugin allows you to choose between Git tags or sha1 of your SCM repository so Git Plugin installed is required.	0.4.0
☐	UserContent in Git plugin This plugin exposes $JENKINS_HOME/userContent as Git repository.	1.4
☐	Alternative build chooser An alternative build chooser plugin for the Jenkins git plugin.	1.1
☐	Team Concert Git Plugin Integrates Jenkins with Rational Team Concert for Jenkins Builds which use Git as source control. This plugin will create traceability links from a Jenkins build to Rational Team Concert Work Items and build results. This plugin adds traceability links from a Jenkins build to an RTC build result. It also publishes links to work items and annotates the change log generated by Jenkins with links to RTC Work Items; It leverages the current RTC features and workflows that users are already familiar with such as, emails, toaster popups, reporting, dashboards, etc.	1.0.10
☐	Tracking Git Plugin Lets one project track the Git revisions that are built for another project.	1.0
☐	GIT plugin This plugin allows use of Git as a build SCM. A recent Git runtime is required (1.7.9 minimum, 1.8.x recommended). Plugin is only tested on official git client. Use exotic installations at your own risks.	2.4.0

Install without restart | Download now and install after restart | Update information obtained: 1 hr 0 min ago | Check now

Help us localize this page — Page generated: Oct 6, 2015 10:30:16 PM — REST API — Jenkins ver. 1.609.3

The list will then be filtered. Check the Git Plugin option and click on the button 'Install without restart'.



Update Center [Jenki
localhost:8080/jenkins/pluginManager/available
Jenkins › Plugin Manager

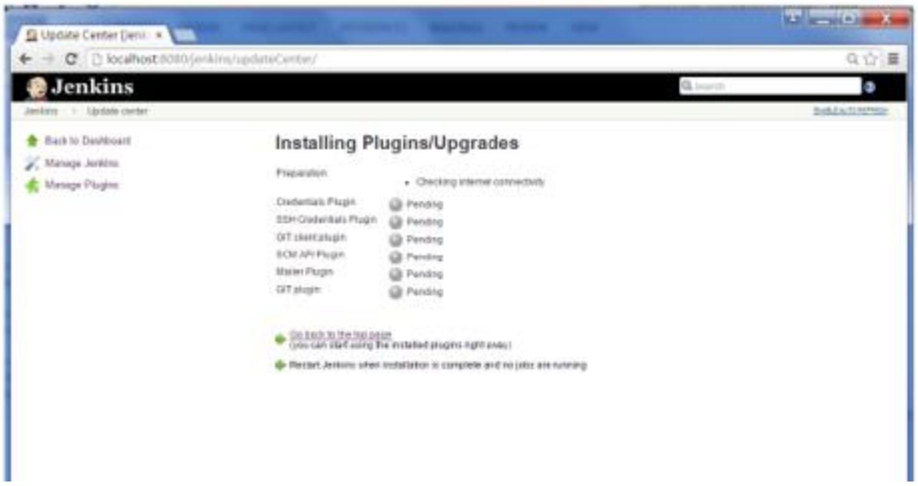
Filter: Git plugin

Install	Name	Version
☐	Git Parameter Plug-In This plugin allows you to choose between Git tags or sha1 of your SCM repository so Git Plugin installed is required.	0.4.0
☐	/userContent in Git plugin This plugin exposes $JENKINS_HOME/userContent as Git repository.	1.4
☐	Alternative build chooser An alternative build chooser plugin for the Jenkins git plugin.	1.1
☐	Team Concert Git Plugin Integrates Jenkins with Rational Team Concert for Jenkins Builds which use Git as source control. This plugin will create traceability links from a Jenkins build to Rational Team Concert Work Items and build results. This plugin adds traceability links from a Jenkins build to an RTC build result. It also publishes links to work items and annotates the change log generated by Jenkins with links to RTC Work Items. It leverages the current RTC features and workflows that users are already familiar with such as, emails, toaster popups, reporting, dashboards, etc.	1.0.10
☐	Tracking Git Plugin Lets one project track the Git revisions that are built for another project.	1.0
✓	GIT plugin This plugin allows use of Git as a build SCM. A recent Git runtime is required (1.7.9 minimum, 1.8.x recommended). Plugin is only tested on official git client. Use exotic installations at your own risks.	2.4.0

Update information obtained: 1 hr 0 min ago

Page generated: Oct 6, 2015 10:32:16 PM REST API Jenkins ver. 1.609.3

The installation will then begin and the screen will be refreshed to show the status of the download.



Update Center [Jenk
localhost:8080/jenkins/updateCenter/
Jenkins › Update center

Installing Plugins/Upgrades

Preparation
- Checking internet connectivity

Credentials Plugin	Pending
SSH Credentials Plugin	Pending
GIT client plugin	Pending
SCM API Plugin	Pending
Mailer Plugin	Pending
GIT plugin	Pending

Go back to the top page
(you can start using the installed plugins right away)

Restart Jenkins when installation is complete and no jobs are running

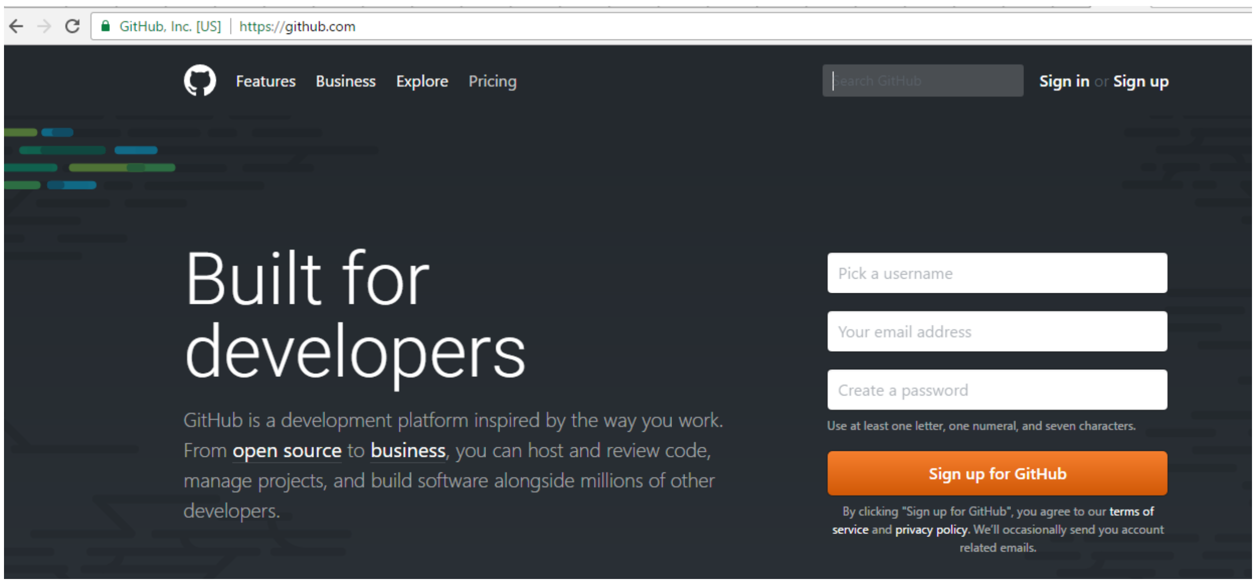
Once all installations are complete, restart Jenkins by issue the following command in the browser. **http://localhost:8080/jenkins/restart**

Before you are going to Jenkins-Git Setup, must and should you have a Github account and Github URL.

➔ Create the Github account, click the bellow URL.

https://github.com/

Then you can see the bellow home page.



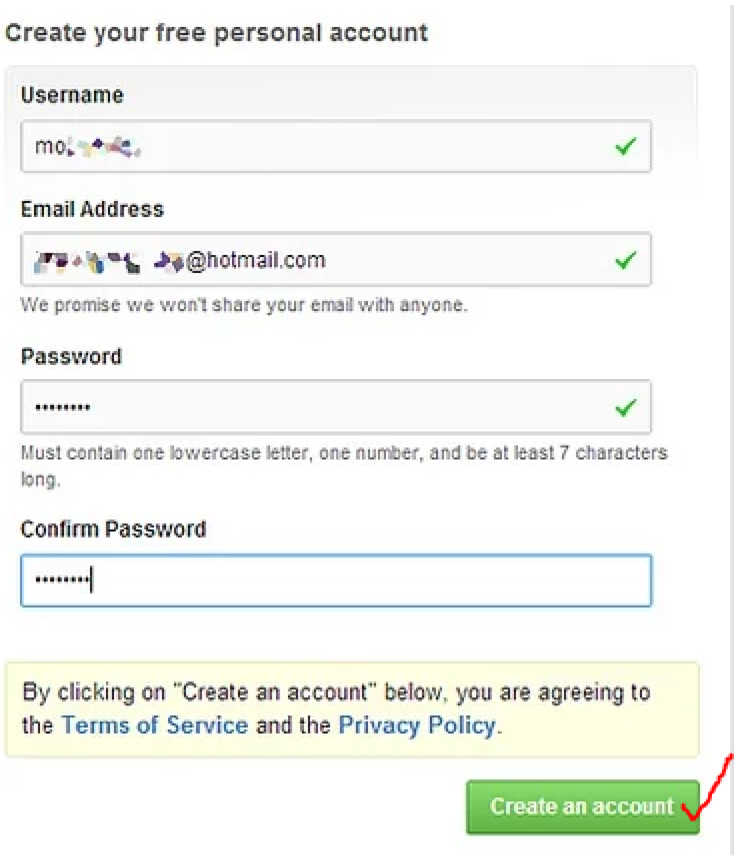
Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers.

Use at least one letter, one numeral, and seven characters.

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

→Then click the *Sign UP* and enter the required details and click the *create account* button.



Create your free personal account

Username

mo[redacted]

Email Address

[redacted]@hotmail.com

We promise we won't share your email with anyone.

Password

••••••••

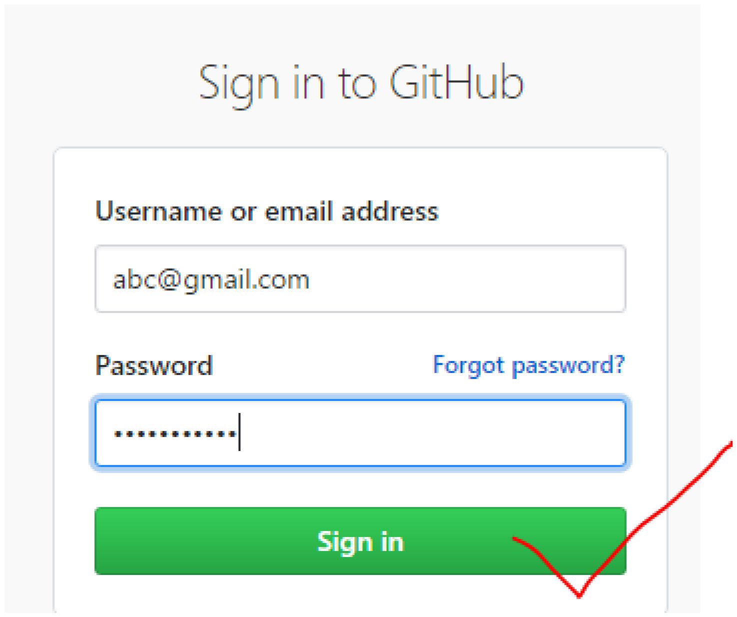
Must contain one lowercase letter, one number, and be at least 7 characters long.

Confirm Password

••••••••

By clicking on "Create an account" below, you are agreeing to the Terms of Service and the Privacy Policy.

➔ Then login into the github account.



Sign in to GitHub

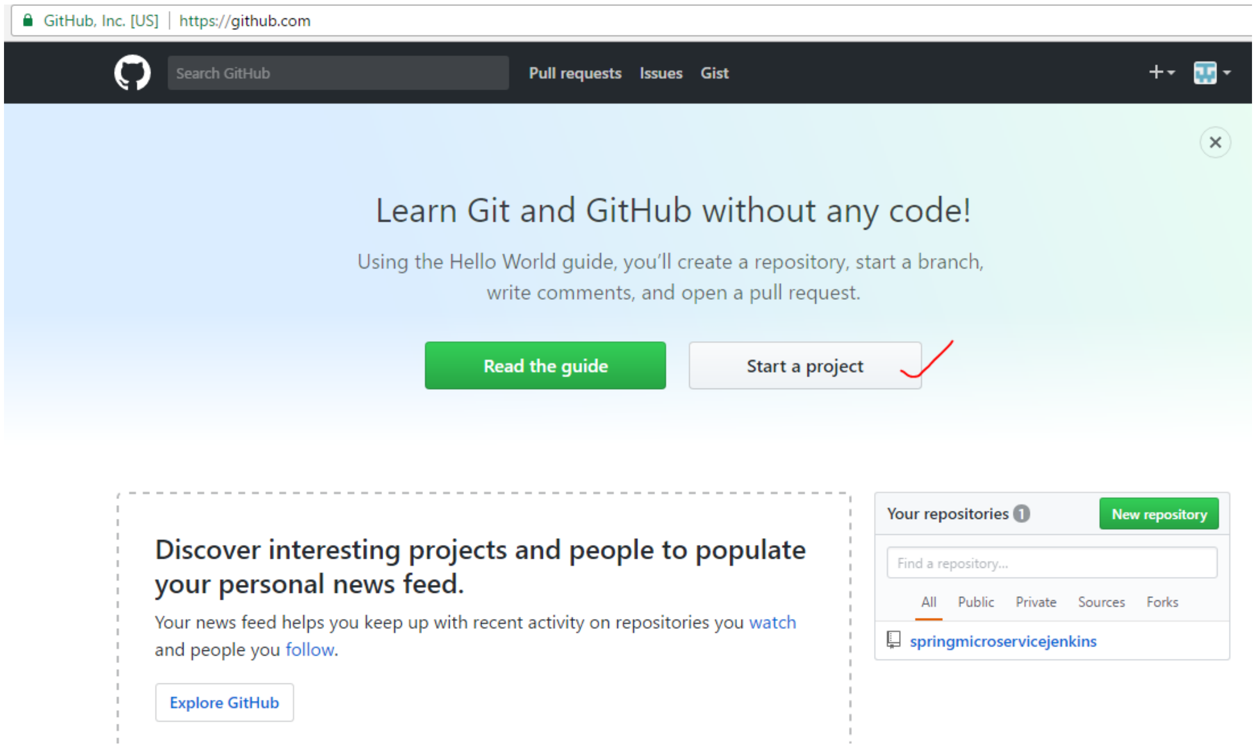
Username or email address

abc@gmail.com

Password

••••••••••••

➔ In the next screen click the *start a project*



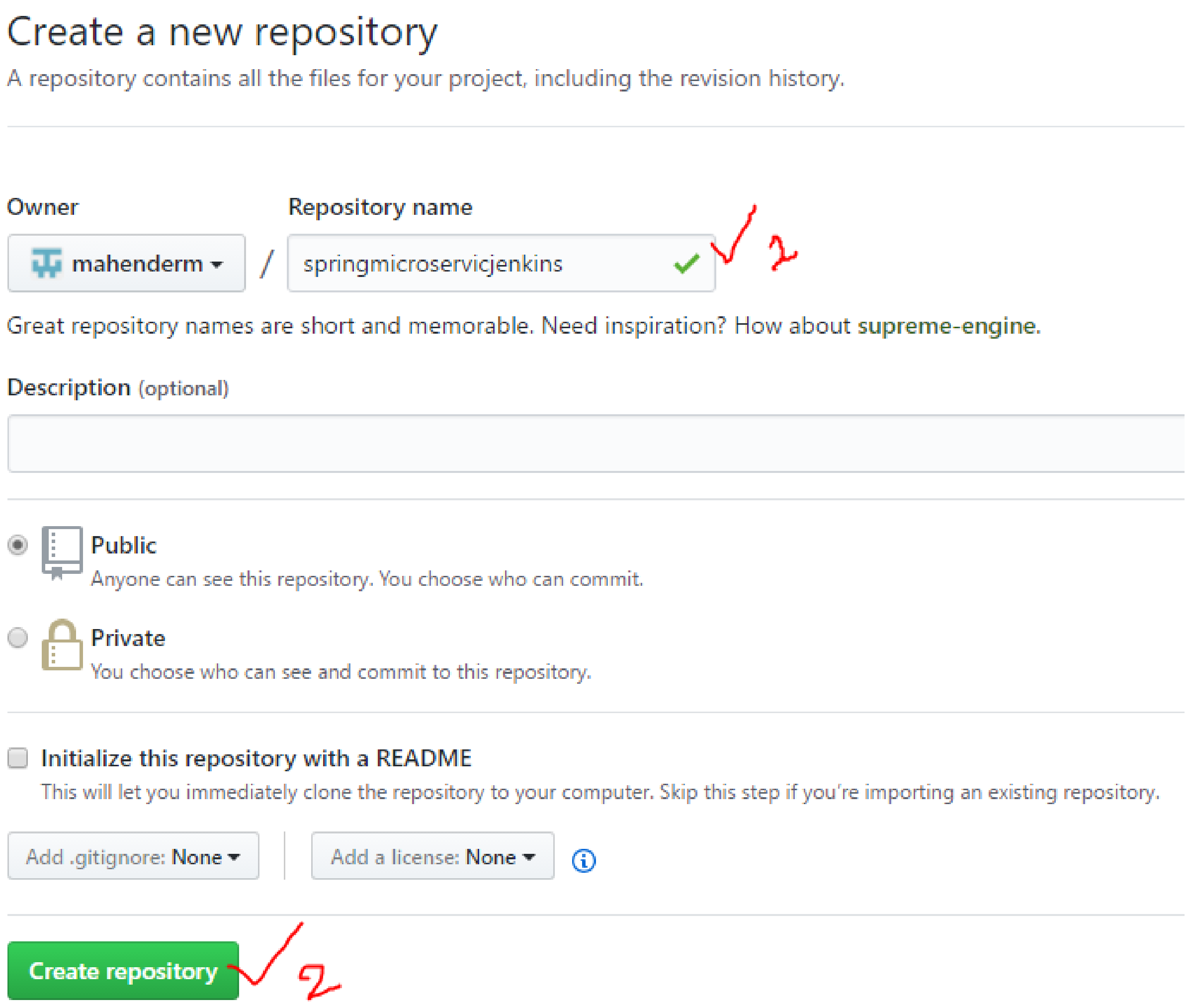
Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch and people you follow.

➔ In the next screen create a new repository with any name and click the *create repository*



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: mahenderm / Repository name: springmicroservicjenkins ✓ 2

Great repository names are short and memorable. Need inspiration? How about supreme-engine.

Description (optional)

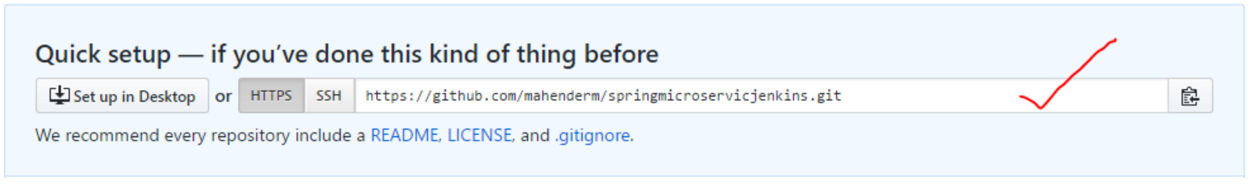
● Public — Anyone can see this repository. You choose who can commit.

○ Private — You choose who can see and commit to this repository.

☐ Initialize this repository with a README — This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

➔ Next screen we will see your URL with new Repository.

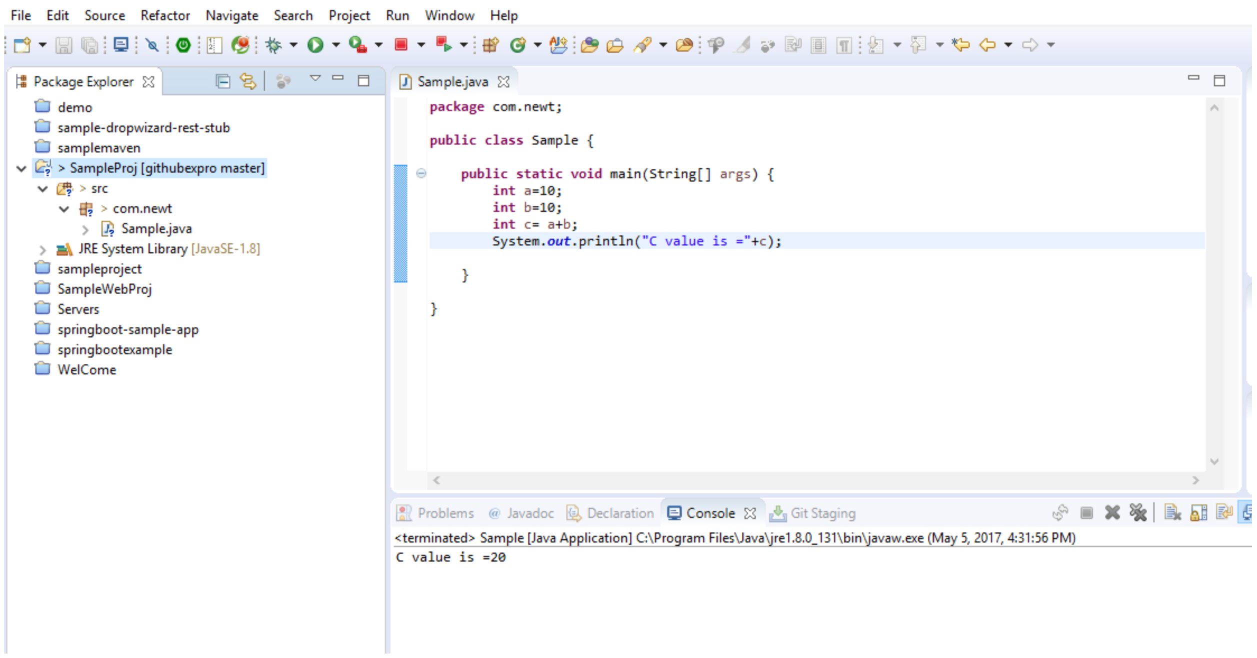


Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/mahenderm/springmicroservicjenkins.git ✓

We recommend every repository include a README, LICENSE, and .gitignore.

➔ Create the Sample java or any application.(use eclipse or any IDE).



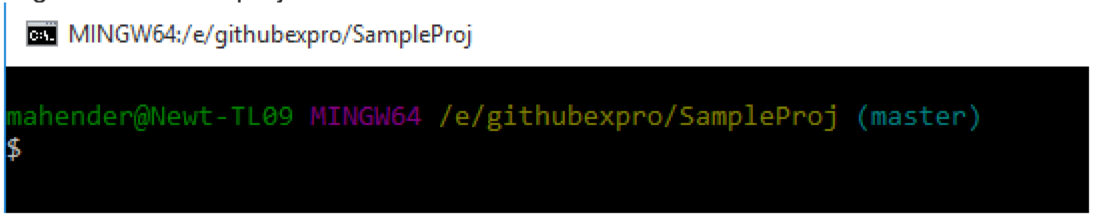
```
package com.newt;

public class Sample {

    public static void main(String[] args) {
        int a=10;
        int b=10;
        int c= a+b;
        System.out.println("C value is ="+c);

    }

}
```

```
<terminated> Sample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (May 5, 2017, 4:31:56 PM)
C value is =20
```

We push the above code in *github*. Before we are going to push the code we can download the *git* in our local system. Click the bellow URL.

https://git-scm.com/download/win

After downloaded git in your local system you can go the your project folder path and Right click on the project and select the *Gitbash*. Will see one terminal window.



```
MINGW64:/e/githubexpro/SampleProj

mahender@Newt-TL09 MINGW64 /e/githubexpro/SampleProj (master)
$
```

➔ Now we push the code in our github account using following commands one by one.

…or create a new repository on the command line

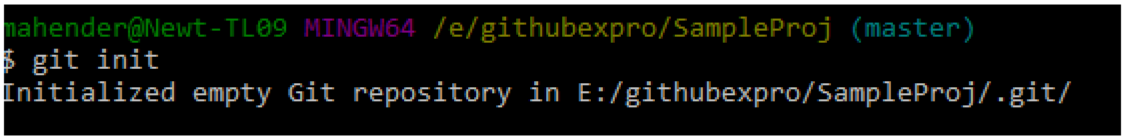
```
echo "# springmicroservicjenkins" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mahenderm/springmicroservicjenkins.git
git push -u origin master
```

…or push an existing repository from the command line

```
git remote add origin https://github.com/mahenderm/springmicroservicjenkins.git
git push -u origin master
```

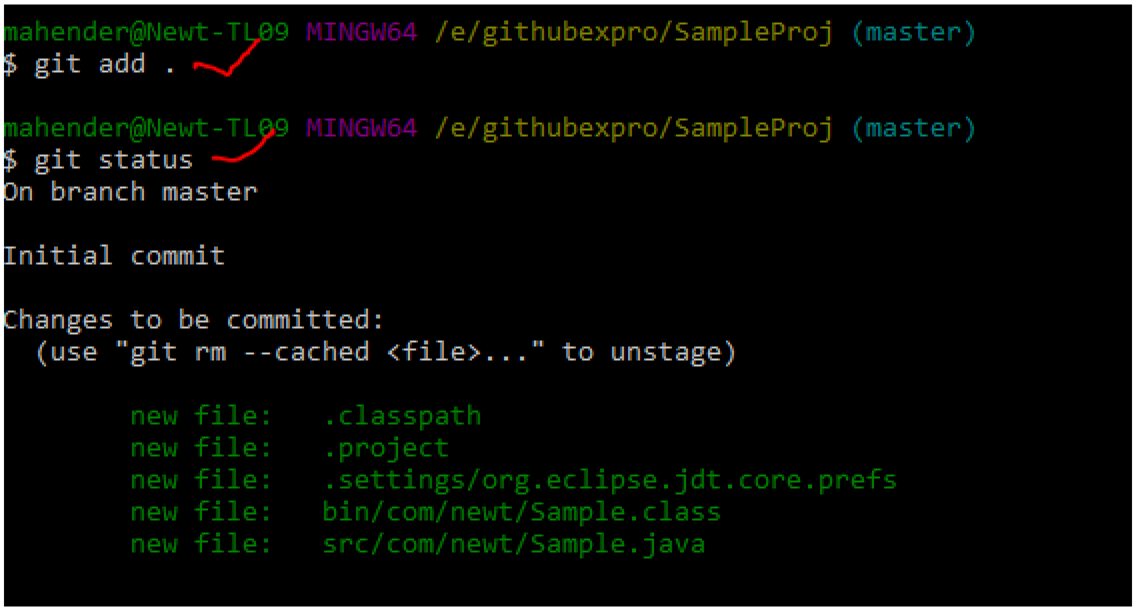
→ We can type *$ git init*. That is empty repository

.



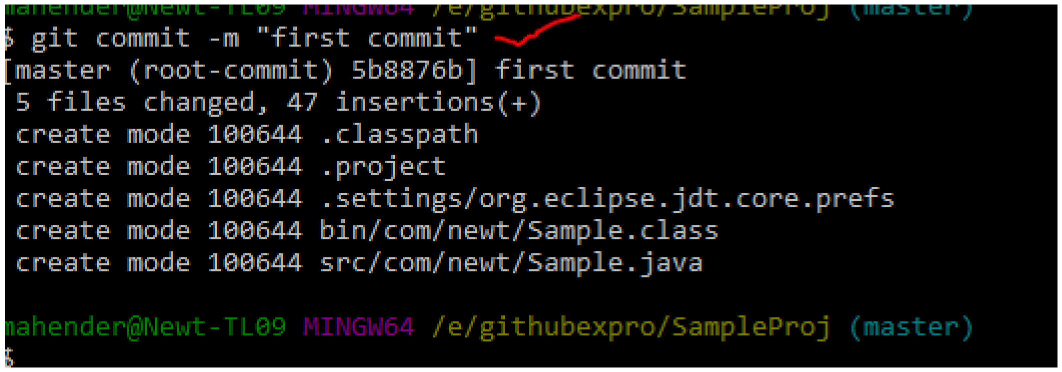
```
mahender@Newt-TL09 MINGW64 /e/githubexpro/SampleProj (master)
$ git init
Initialized empty Git repository in E:/githubexpro/SampleProj/.git/
```

→ we have to add all file you can use ***$ git add .*** (Or) you can add single file use ***$ git add filename.*** Here now I am adding all the files after you can use ***$ git status*** will see what are the files added.



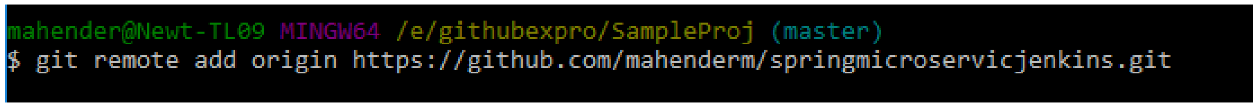
```
mahender@Newt-TL09 MINGW64 /e/githubexpro/SampleProj (master)
$ git add .

mahender@Newt-TL09 MINGW64 /e/githubexpro/SampleProj (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .classpath
        new file:   .project
        new file:   .settings/org.eclipse.jdt.core.prefs
        new file:   bin/com/newt/Sample.class
        new file:   src/com/newt/Sample.java
```

→ After commit the code using ***$ git commit –m "we can write any comment here"***



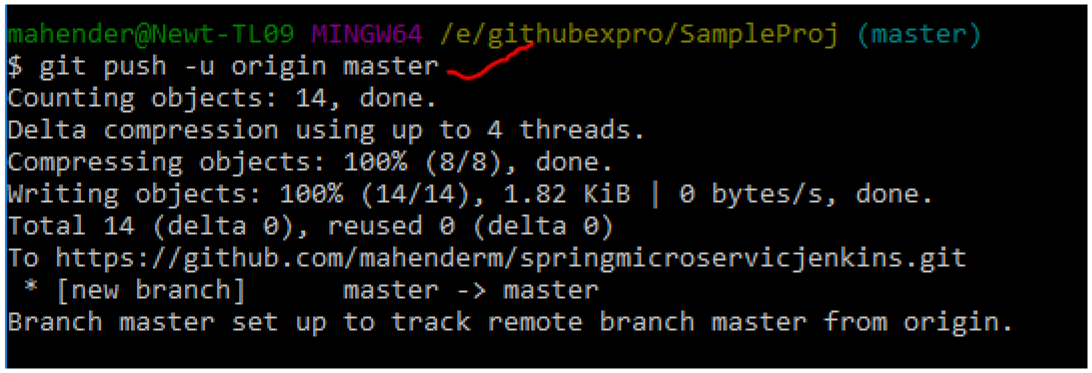
```
$ git commit -m "first commit"
[master (root-commit) 5b8876b] first commit
 5 files changed, 47 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .project
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 bin/com/newt/Sample.class
 create mode 100644 src/com/newt/Sample.java

mahender@Newt-TL09 MINGW64 /e/githubexpro/SampleProj (master)
$
```

→ We add the Our Repository URL



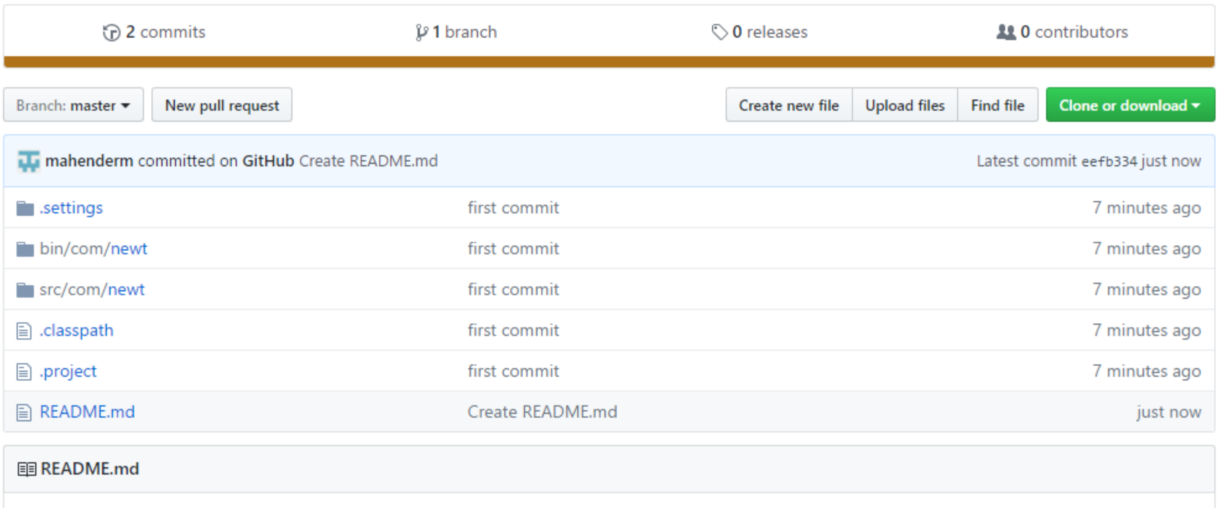
```
mahender@Newt-TL09 MINGW64 /e/githubexpro/SampleProj (master)
$ git remote add origin https://github.com/mahenderm/springmicroservicjenkins.git
```

→ Push the code to our branch $ git push –u origin master



```
mahender@Newt-TL09 MINGW64 /e/githubexpro/SampleProj (master)
$ git push -u origin master
Counting objects: 14, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (14/14), 1.82 KiB | 0 bytes/s, done.
Total 14 (delta 0), reused 0 (delta 0)
To https://github.com/mahenderm/springmicroservicjenkins.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

→ Will see the our code is commited or not in our github.



2 commits · 1 branch · 0 releases · 0 contributors

Branch: master · New pull request · Create new file · Upload files · Find file · Clone or download

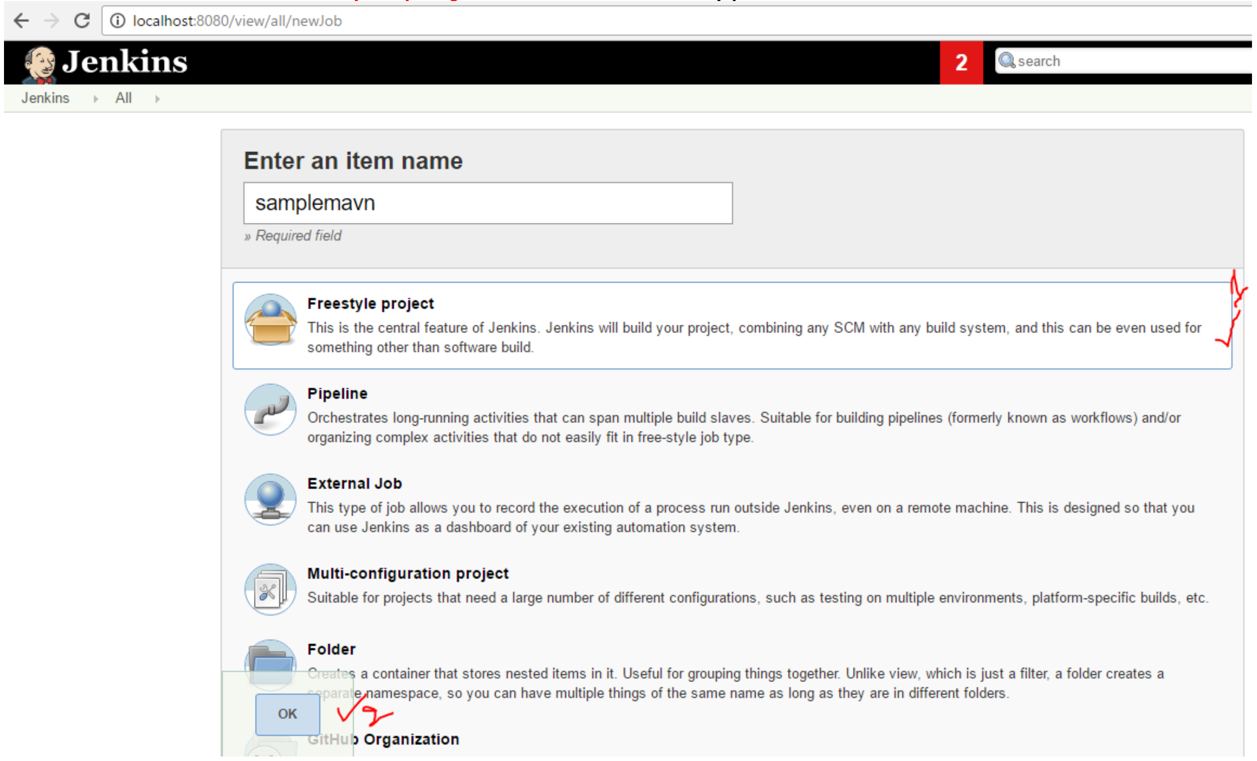
mahenderm committed on GitHub Create README.md — Latest commit eefb334 just now

.settings	first commit	7 minutes ago
bin/com/newt	first commit	7 minutes ago
src/com/newt	first commit	7 minutes ago
.classpath	first commit	7 minutes ago
.project	first commit	7 minutes ago
README.md	Create README.md	just now

README.md

Creating a Jenkins job

First of all, login to your Jenkins server. Click on New Item in the menu options for Jenkins. Then enter a name for a job, in the following case, the name entered is 'SampleMaven'. Select 'Freestyle project' as the item type. Click the Ok button.



localhost:8080/view/all/newJob

Jenkins › All

Enter an item name

samplemavn

» 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.

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

External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

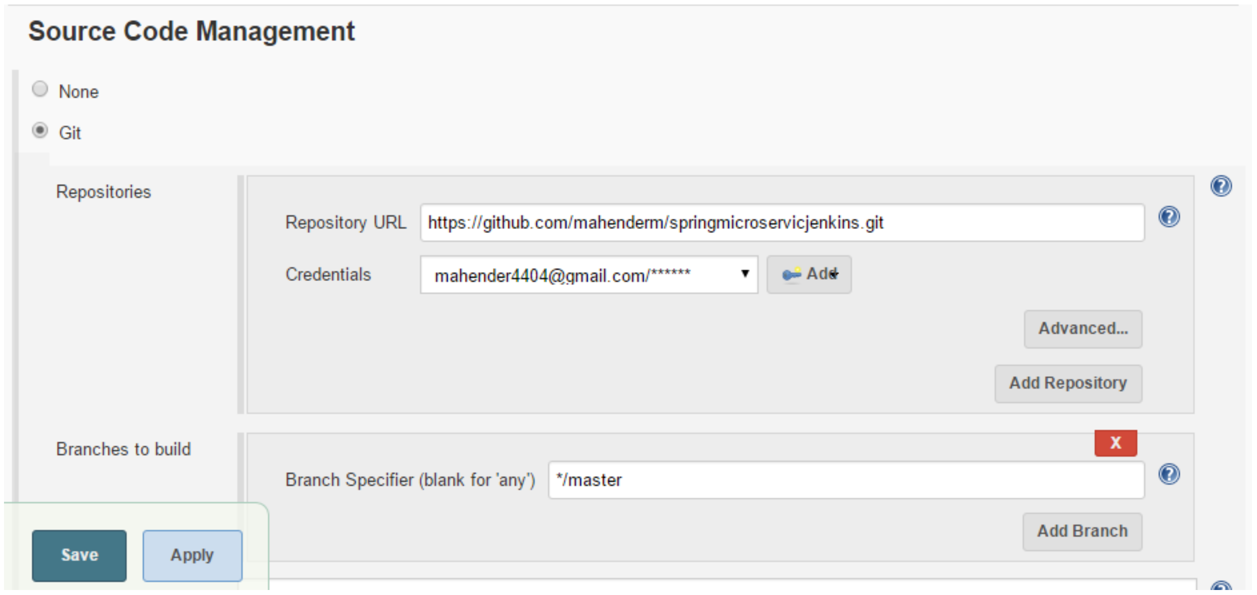
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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 namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

GitHub Organization

In the next screen, if you browse to the *"Source-Code-Management"* select the *"Git"* option. If you don't see it, check your GitHub Plugin installation. Enter the repository clone-URL in the appearing text field. When you are using a public GitHub repository, you don't have to specify any credentials otherwise enter them.



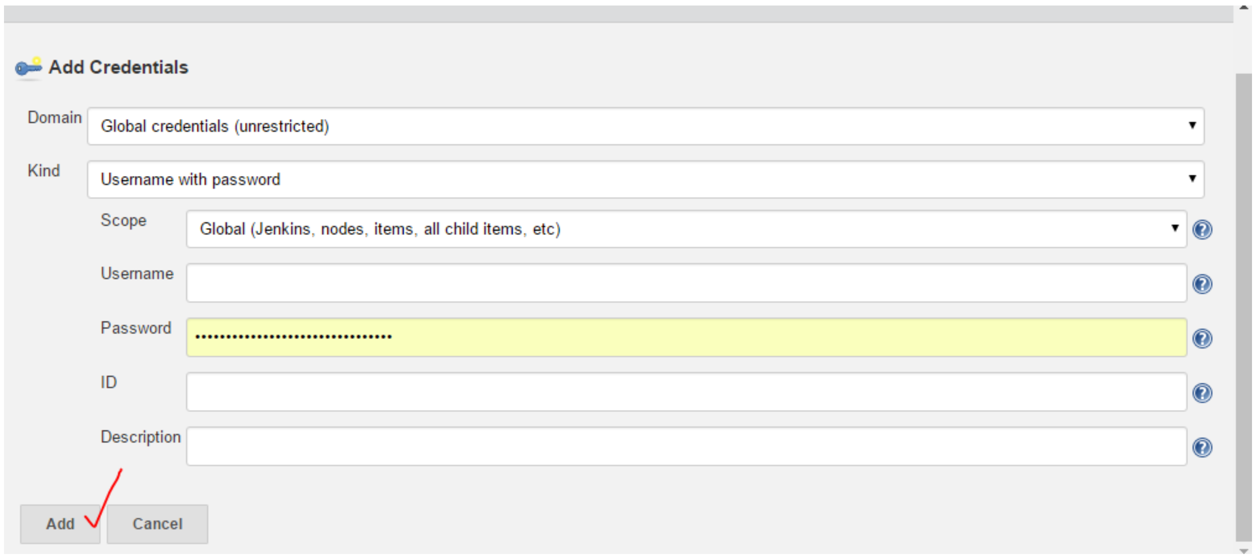
Source Code Management

None
Git

Repositories
Repository URL: https://github.com/mahenderm/springmicroservicjenkins.git
Credentials: mahender4404@gmail.com/******

Branches to build
Branch Specifier (blank for 'any'): */master

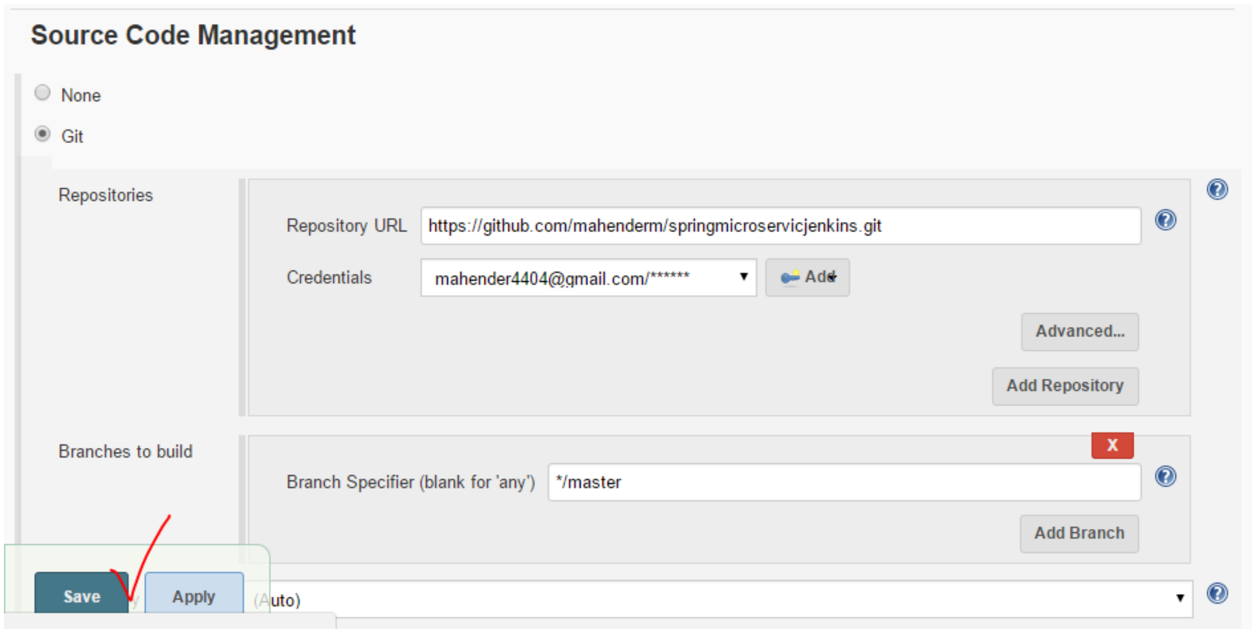
→ Add the git proper credentials and click the add button.



Add Credentials

Domain: Global credentials (unrestricted)
Kind: Username with password
Scope: Global (Jenkins, nodes, items, all child items, etc)
Username:
Password: ••••••••••••••••••••••••••••••••••
ID:
Description:

→Finally click the save button

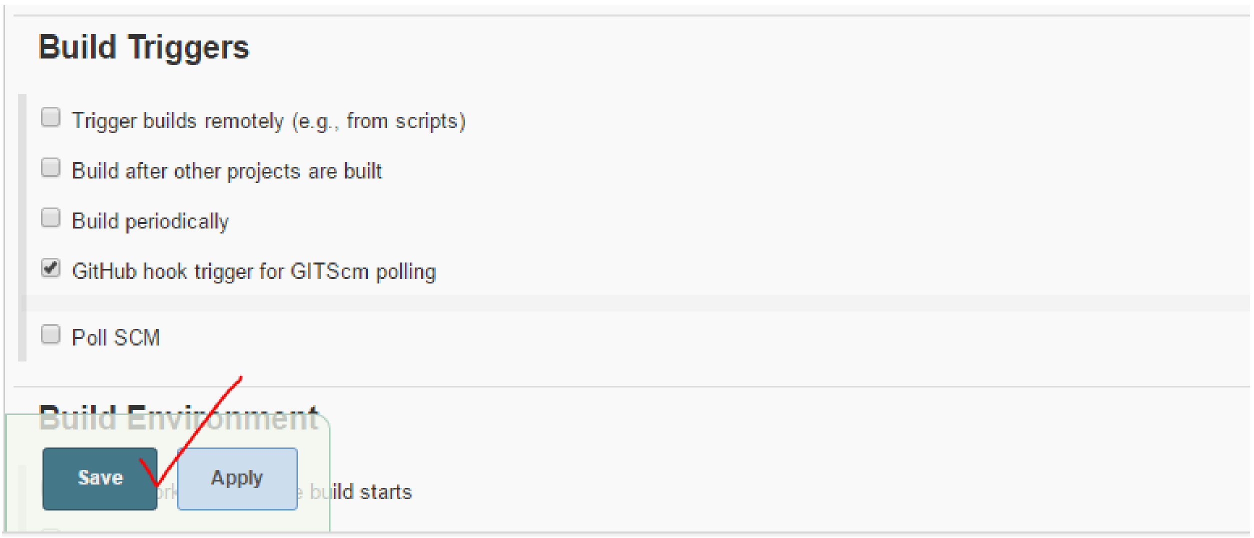

Source Code Management

None [✓ Git]

Repositories
Repository URL: https://github.com/mahenderm/springmicroservicjenkins.git
Credentials: mahender4404@gmail.com/******

Branches to build
Branch Specifier (blank for 'any'): */master

→ In the next step, we will trigger it by pushing to the GitHub repository. Select the *GitHub hook trigger for GITScm polling.* Click the *save* button.



Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [✓] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Build Environment

Configure the GitHub repository

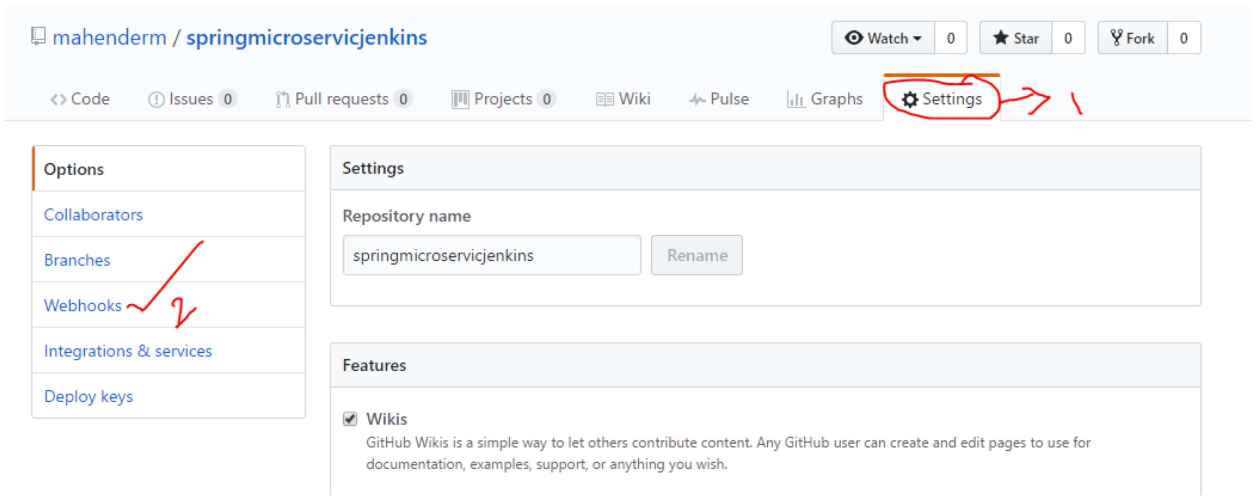
To be able to trigger the build process by GitHub, you have to configure the Jenkins instance which should be triggered after the push. For this purpose the Jenkins web hook URL is required and must be submitted in the GitHub project.

Web hook

Web hooks allow you to build or set up integrations which subscribe to certain events on GitHub.com. When one of those events is triggered, we'll send a HTTP POST payload to the web hook's configured URL. Web hooks can be used to update an external issue tracker, trigger CI builds, update a backup mirror, or even deploy to your production server. You're only limited by your imagination.

Each web hook can be installed on an organization or a specific repository. Once installed, they will be triggered each time one or more subscribed events occurs on that organization or repository.

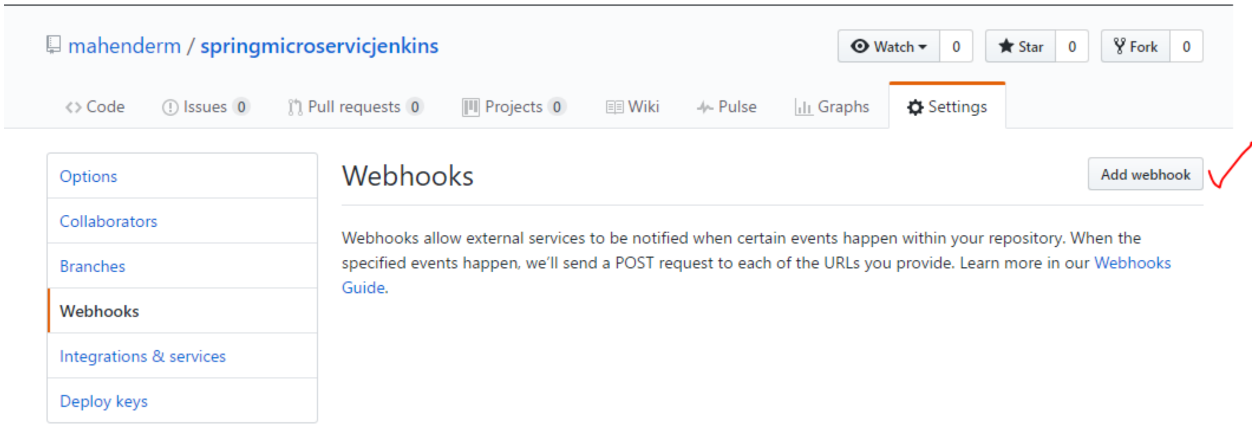
→ Go to the git hub setting and select the web hook.



springmicroservicjenkins

GitHub Wikis is a simple way to let others contribute content. Any GitHub user can create and edit pages to use for documentation, examples, support, or anything you wish.

→ Then you click the Add Webhook.



Webhooks

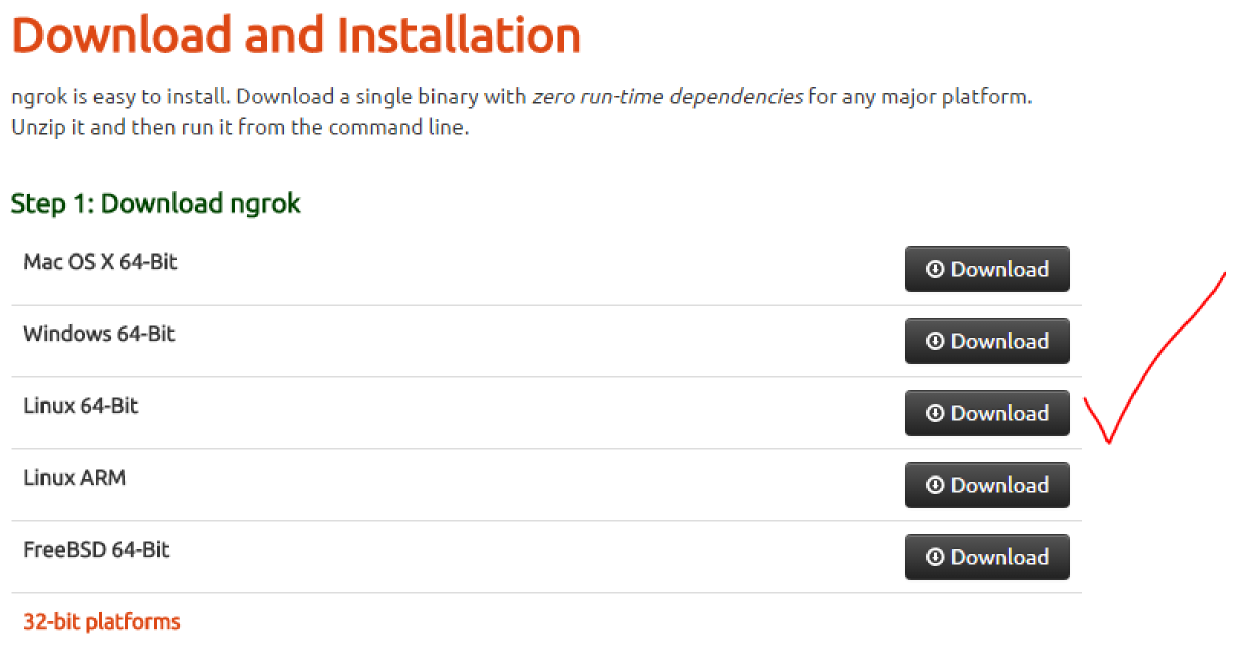
Webhooks allow external services to be notified when certain events happen within your repository. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

→Next screen we add the payload URL. So here we need ngrok. Use Bellow link to download ngrok.

https://ngrok.com/download

and select the windows 64-bit(Depending on OS).

Ngrok: Ngrok is a multiplatform tunnelling, reverse proxy software that establishes secure tunnels from a public endpoint such as internet to a locally running network service while capturing all traffic for detailed inspection and replay.



Download and Installation

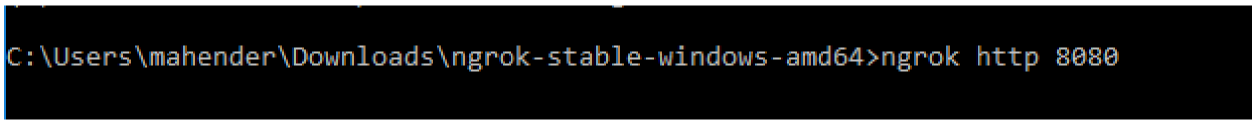
ngrok is easy to install. Download a single binary with *zero run-time dependencies* for any major platform. Unzip it and then run it from the command line.

Step 1: Download ngrok

Mac OS X 64-Bit	Download
Windows 64-Bit	Download
Linux 64-Bit	Download ✓
Linux ARM	Download
FreeBSD 64-Bit	Download

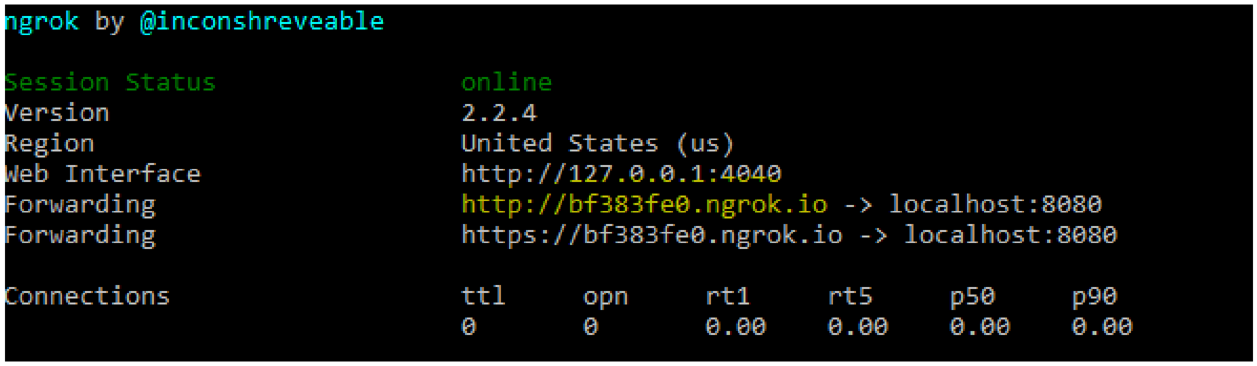
32-bit platforms

You downloaded Zip File so you extract the file. Go to the cmd prompt and set the ngrok path and type the command $ ngrok http 8080(here "http 8080" is Jenkins port number).



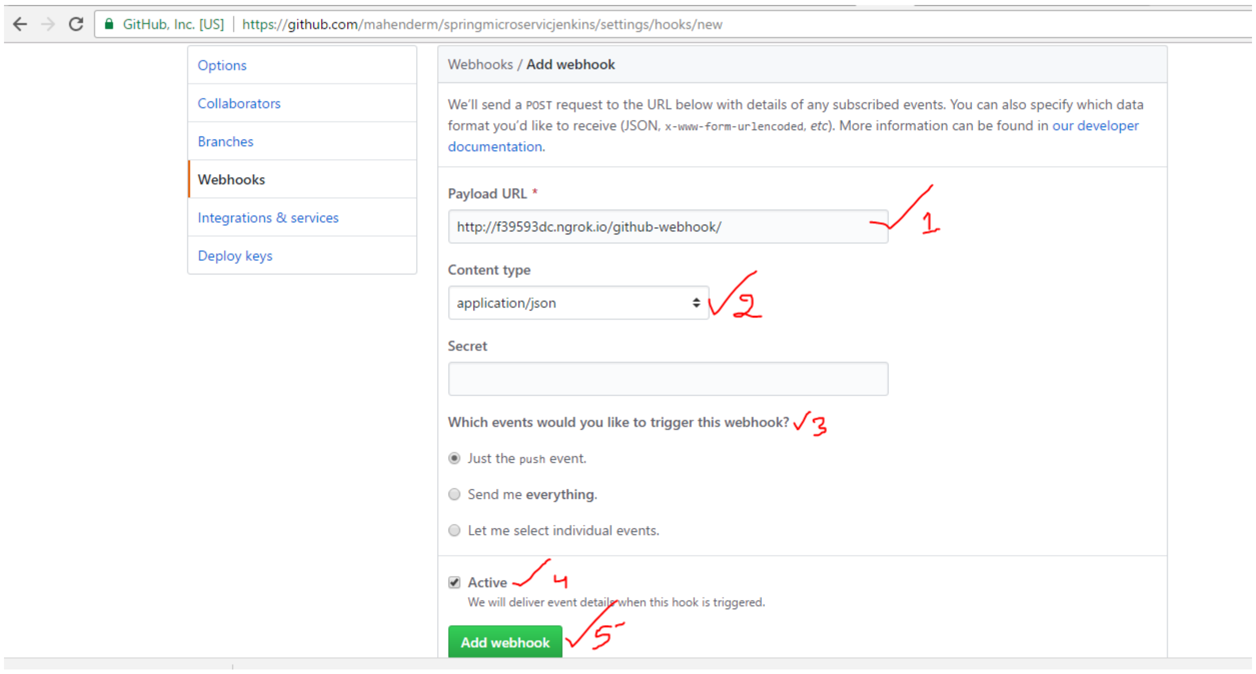
```
C:\Users\mahender\Downloads\ngrok-stable-windows-amd64>ngrok http 8080
```

→Enter and we copy the highlighted line



```
ngrok by @inconshreveable

Session Status                online
Version                       2.2.4
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    http://bf383fe0.ngrok.io -> localhost:8080
Forwarding                    https://bf383fe0.ngrok.io -> localhost:8080

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

→Paste into the payload URL and select the content type is application/json and select the required events and click the Active check box and click the Add webhook finally.



https://github.com/mahenderm/springmicroservicjenkins/settings/hooks/new

Options
Collaborators
Branches
Webhooks
Integrations & services
Deploy keys

Webhooks / **Add webhook**

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *
http://f39593dc.ngrok.io/github-webhook/ ✓1

Content type
application/json ✓2

Secret

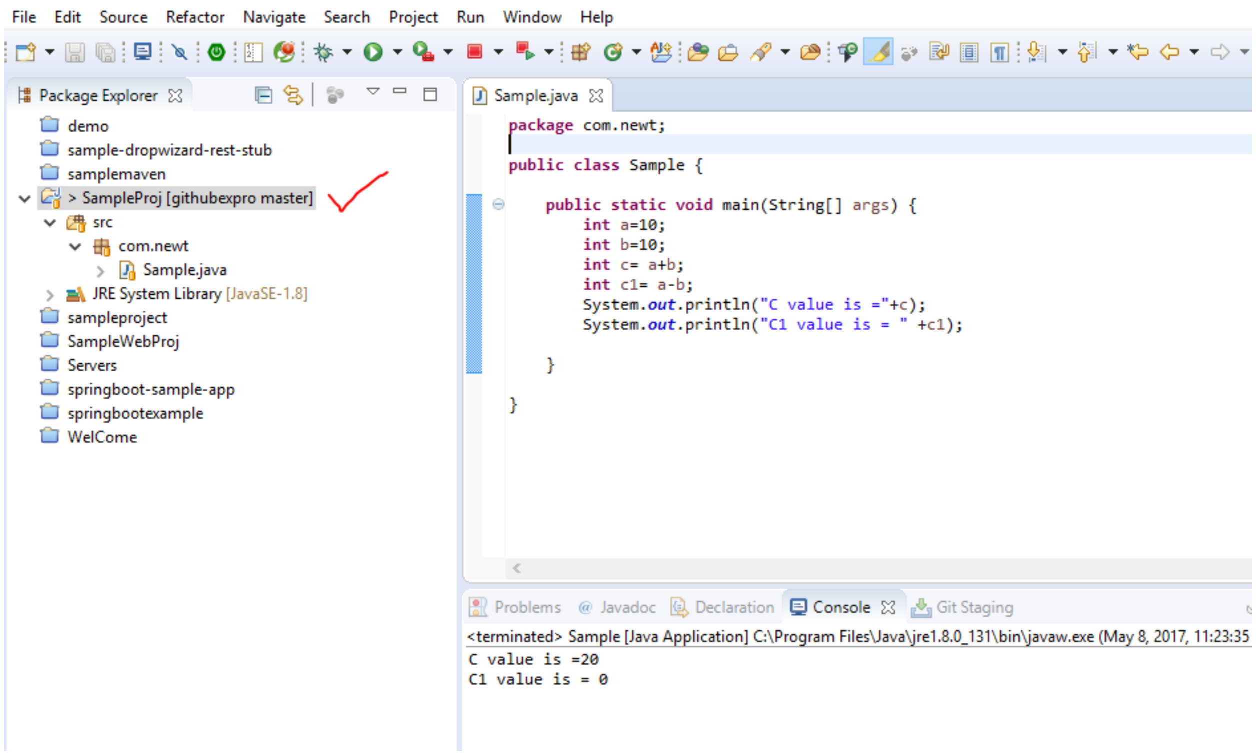
Which events would you like to trigger this webhook? ✓3
(●) Just the push event.
( ) Send me everything.
( ) Let me select individual events.

[✓] Active ✓4
We will deliver event details when this hook is triggered.

Add webhook ✓5

→We create any sample pogram and push into the cloud repository(github) by using above commands and see the automatic triggering in Jenkins.

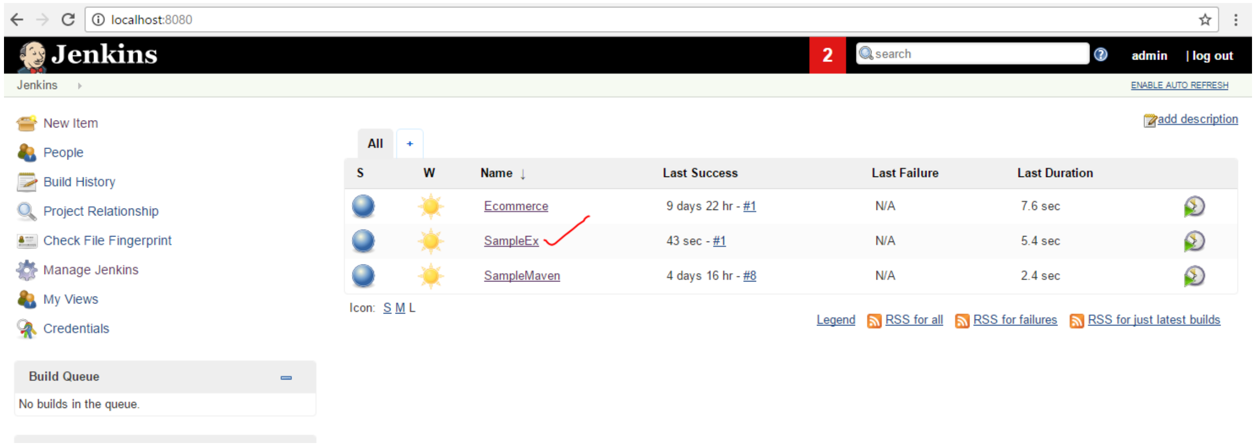
→ I created sample application that app name is sampleproj.


File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
demo
sample-dropwizard-rest-stub
samplemaven
> SampleProj [githubexpro master]
src
com.newt
Sample.java
JRE System Library [JavaSE-1.8]
sampleproject
SampleWebProj
Servers
springboot-sample-app
springbootexample
WelCome
Sample.java
Problems Javadoc Declaration Console Git Staging

```
package com.newt;

public class Sample {

    public static void main(String[] args) {
        int a=10;
        int b=10;
        int c= a+b;
        int c1= a-b;
        System.out.println("C value is ="+c);
        System.out.println("C1 value is = " +c1);

    }

}
```

```
<terminated> Sample [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (May 8, 2017, 11:23:35
C value is =20
C1 value is = 0
```

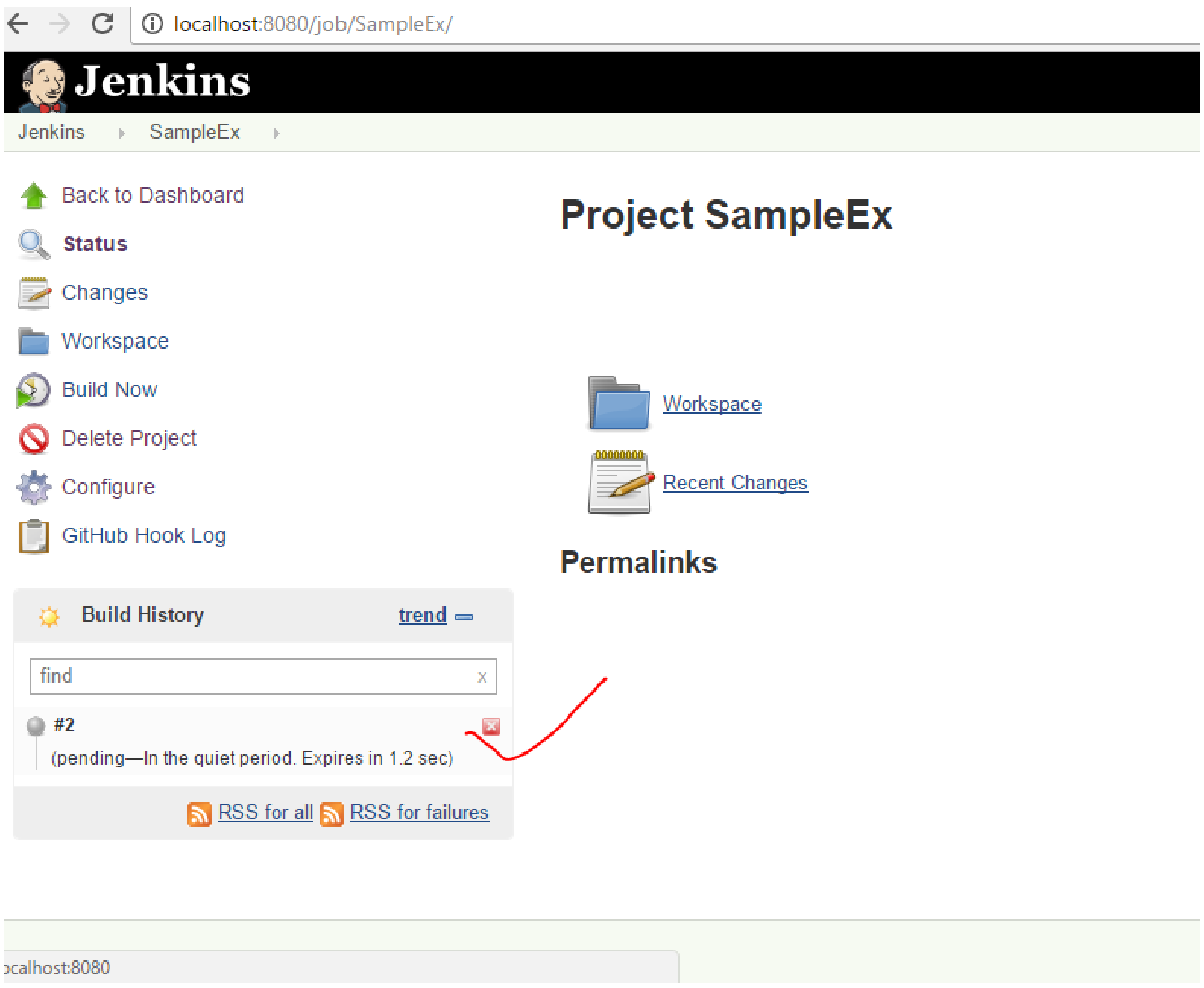
→I created new job in Jenkins name with sampleEx and this code I pushed into the my cloud repository .



S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	Ecommerce	9 days 22 hr - #1	N/A	7.6 sec
●	☀	SampleEx	43 sec - #1	N/A	5.4 sec
●	☀	SampleMaven	4 days 16 hr - #8	N/A	2.4 sec

Build Queue
No builds in the queue.

→See the automatic triggering.


localhost:8080/job/SampleEx/

Jenkins › SampleEx ›

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- GitHub Hook Log

Build History trend

find

#2
(pending—In the quiet period. Expires in 1.2 sec)

RSS for all RSS for failures

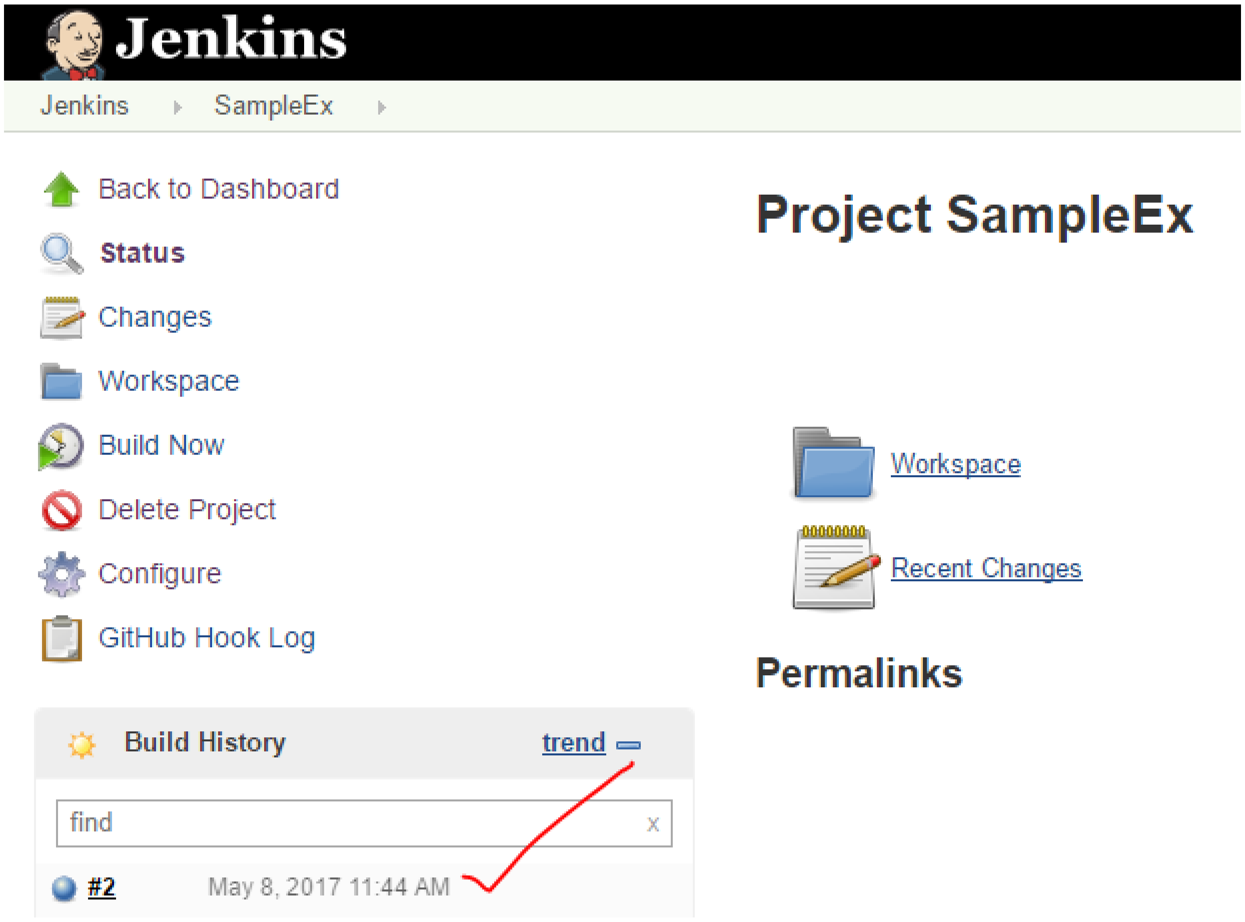
Project SampleEx

Workspace

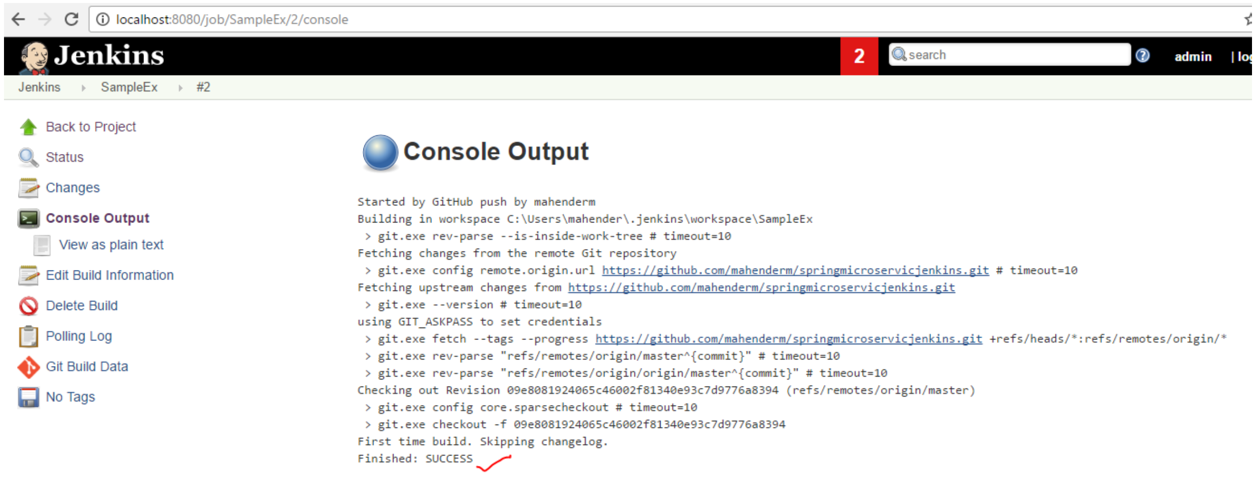
Recent Changes

Permalinks

→ see the automatic triggering.



→Click on the Build History and check the console output will see the success.



```
Started by GitHub push by mahenderm
Building in workspace C:\Users\mahender\.jenkins\workspace\SampleEx
 > git.exe rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/mahenderm/springmicroservicjenkins.git # timeout=10
Fetching upstream changes from https://github.com/mahenderm/springmicroservicjenkins.git
 > git.exe --version # timeout=10
using GIT_ASKPASS to set credentials
 > git.exe fetch --tags --progress https://github.com/mahenderm/springmicroservicjenkins.git +refs/heads/*:refs/remotes/origin/*
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
Checking out Revision 09e8081924065c46002f81340e93c7d9776a8394 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 09e8081924065c46002f81340e93c7d9776a8394
First time build. Skipping changelog.
Finished: SUCCESS
```