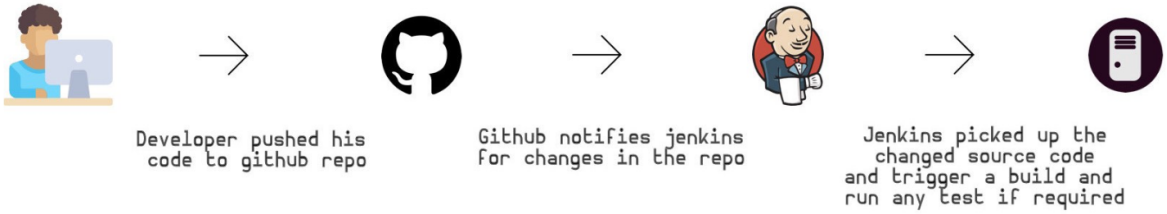
Jenkins on Ubuntu

Jenkins is an open source automation server intended to automate repetitive technical tasks involved in the continuous integration and delivery of software.

Continuous Integration is a development practice that requires developers to integrate code into a shared repository at regular intervals. This concept was meant to remove the problem of finding later occurrence of issues in the build lifecycle. Continuous integration requires the developers to have frequent builds. The common practice is that whenever a code commit occurs, a build should be triggered.

Jenkins will be installed on a server where the central build will take place. The following flowchart demonstrates a very simple workflow of how Jenkins works.



- First, we'll add the repository key to the system.

wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -

- If the system will return OK, it means the key is added successfully. Next, we'll append the Debian package repository address to the server's sources.list

echo deb http://pkg.jenkins.io/debian-stable binary/| sudo tee /etc/apt/sources.list.d/jenkins.list

- When both of these are in place, we'll run updateso that apt-getwill use the new repository

sudo apt-get update

- Finally, we'll install Jenkins and its dependencies, including Java

sudo apt-get install jenkins

Now that Jenkins and its dependencies are in place, we'll start the Jenkins server.

Starting Jenkins Server

Using systemctl we'll start Jenkins :
sudo systemctl start jenkins

Since systemctl doesn't display output, we'll use its status command to verify that it started successfully
sudo systemctl status jenkins

If everything went well, the beginning of the output should show that the service is active and configured to start at boot

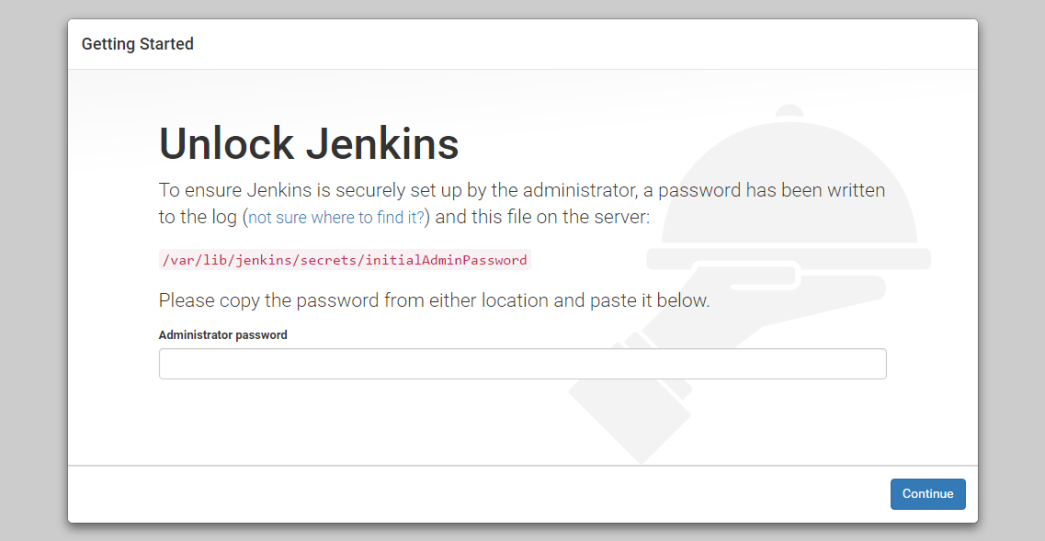
● jenkins.service — LSB: Start Jenkins at boot time
Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)
Active: active (exited) since Thu 2017–08–10 09:38:44 UTC; 6 days ago
Docs: man:systemd-sysv-generator(8)
Process: 1422 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)
Tasks: 0
Memory: 0B
CPU: 0

Now that Jenkins is installed, we can complete the initial setup.

Setting up Jenkins

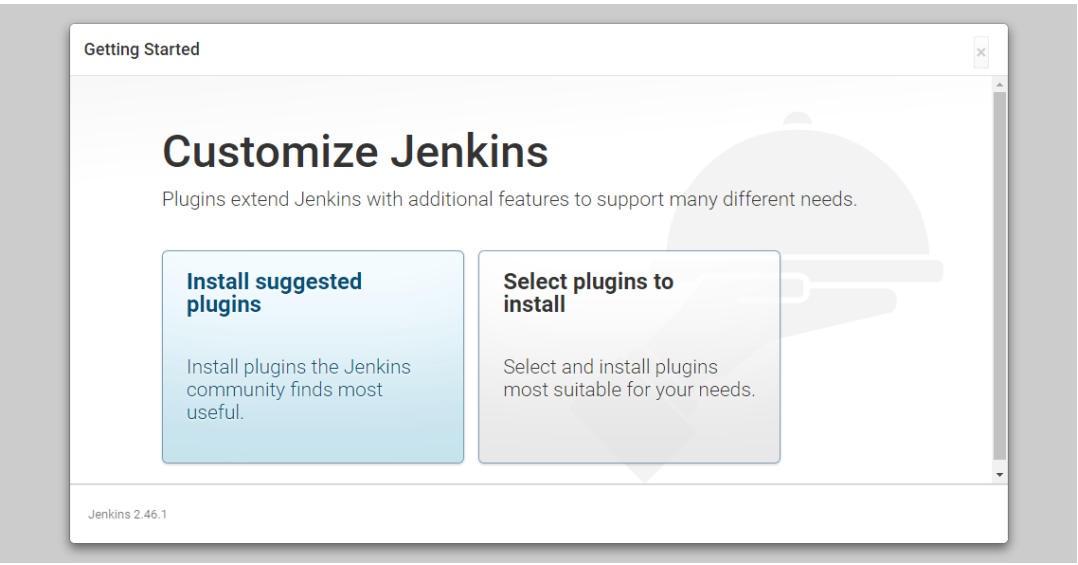
To set up our installation, we'll visit Jenkins by opening web browser and open the link below. Make sure you change ip_address_or_domain_name with your the server domain name or IP address :
http://ip_address_or_domain_name:8080

We should see "Unlock Jenkins" screen, which displays the location of the initial password

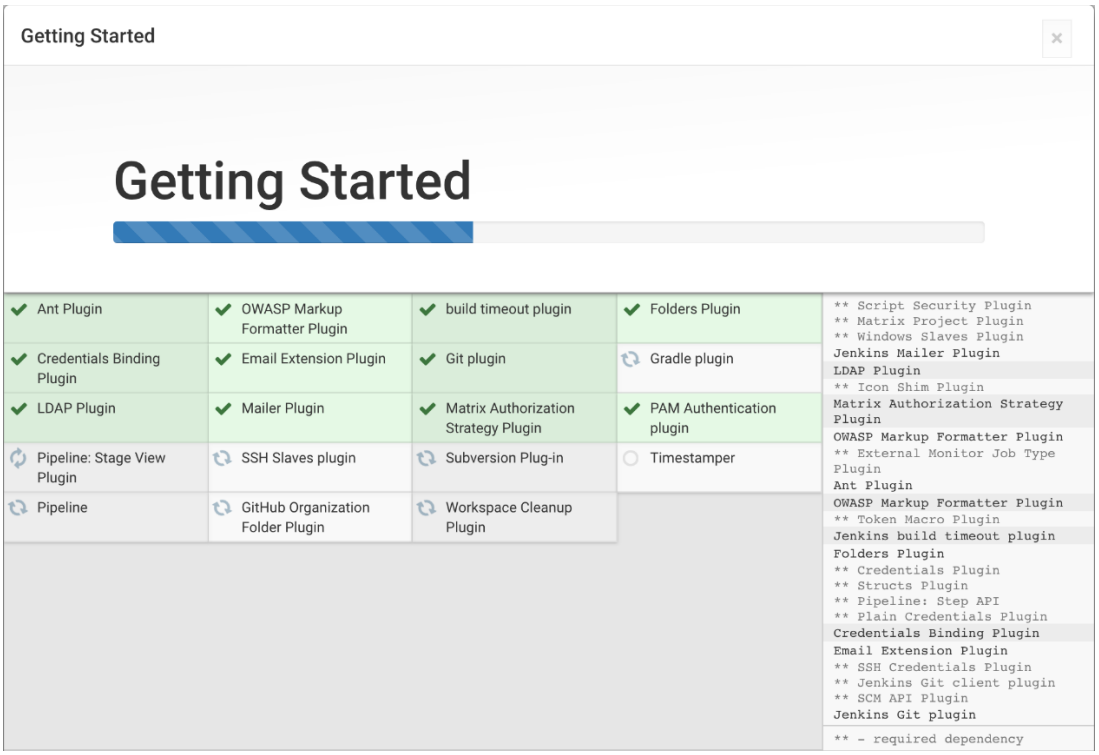


In the terminal window, we'll use the cat command to display the password :
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

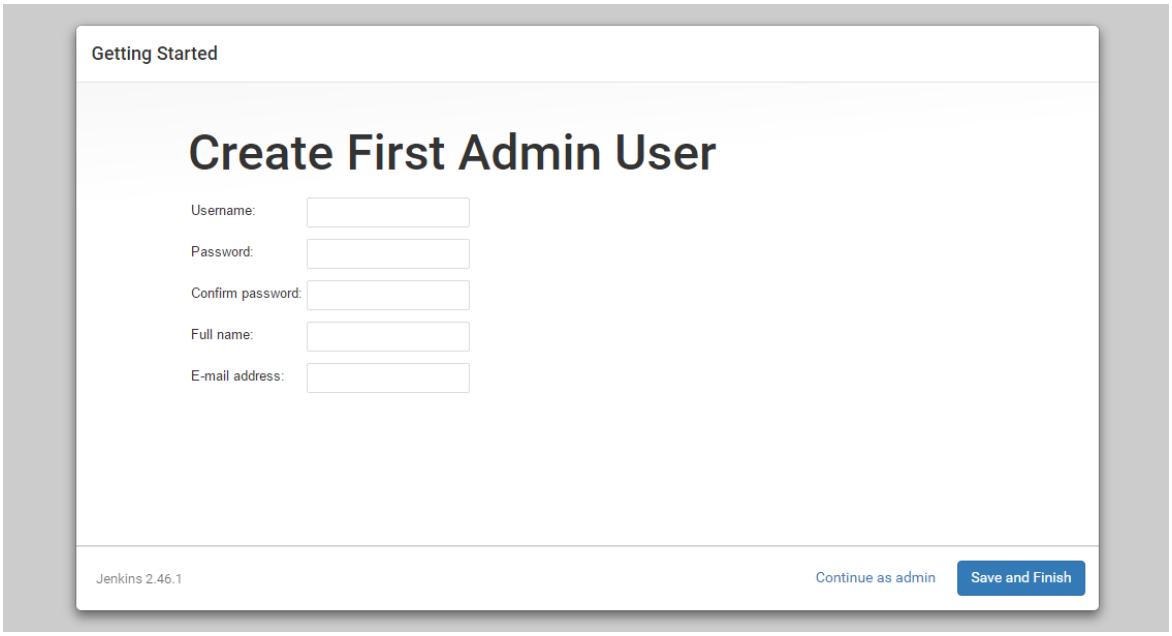
We'll copy the 32-character alphanumeric password from the terminal and paste it into the "Administrator password" field, then click "Continue". The next screen presents the option of installing suggested plugins or selecting specific plugins.



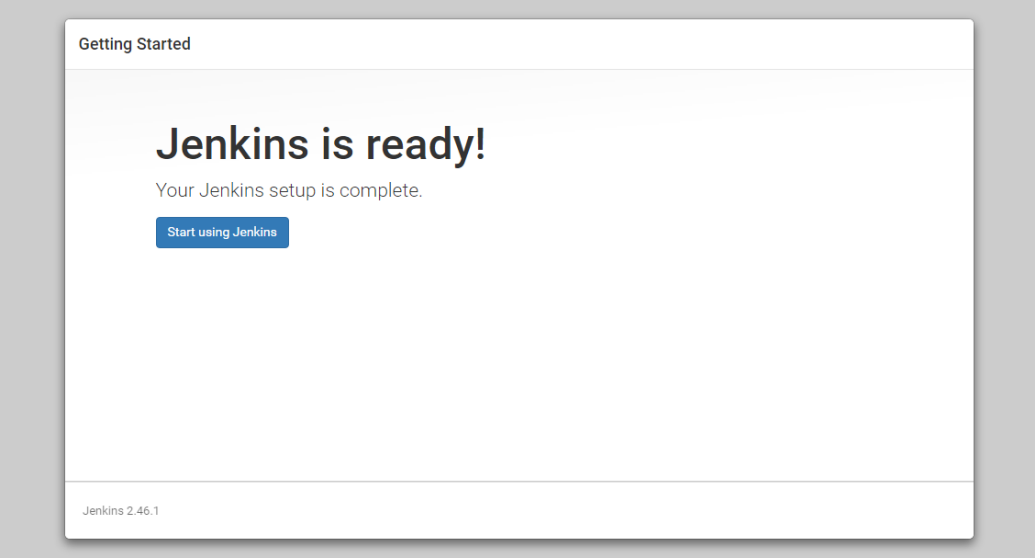
We'll click the "Install suggested plugins" option, which will immediately begin the installation process



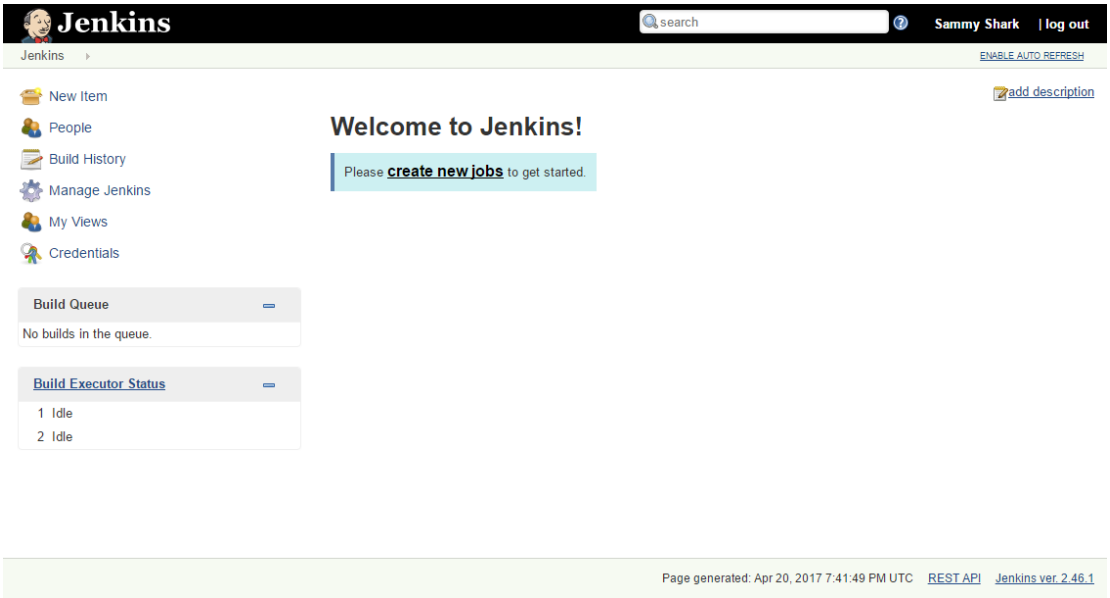
When the installation is complete, we'll be prompted to set up the first administrative user. It's possible to skip this step and continue as admin using the initial password we used above, but we'll take a moment to create the user.



Once the first admin user is in place, you should see a "Jenkins is ready!" confirmation screen.



Click “Start using Jenkins” to visit the main Jenkins dashboard :



At this point, Jenkins has been successfully installed.

Jenkins on CentOS 7

Jenkins is free and open source continues integration tool and it’s code is written in Java. It provides the feature of continues build and deployment or in other words we can say it is a automation server. Jenkins are used where continues build and integration is going on for software development.

Step:1 Add Jenkins Repository

Jenkins package is not available in the default CentOS and RHEL repositories. So we need to add jenkins repository using the beneath commands.

[root@jenkins ~]# wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins.io/redhat-stable/jenkins.repo

[root@jenkins ~]# rpm --import http://pkg.jenkins.io/redhat-stable/jenkins.io.key

Step:2 Install Jenkins and Java

Run the below yum command to install Jenkins and java.

[root@jenkins ~]# yum install jenkins java-1.8.0-openjdk –y

Step:3 Start and Enable Jenkins Service

Run the following systemctl commands to start and enable the jenkins service

[root@jenkins ~]# systemctl start jenkins

[root@jenkins ~]# systemctl enable Jenkins

Step:4 Open the ports (80 and 8080) in OS firewall.

In case firewall is enabled on your Linux server then run the following commands to open jenkins related ports like 80 and 8080.

[root@jenkins ~]# firewall-cmd --zone=public --add-port=8080/tcp --permanent

success

[root@jenkins ~]# firewall-cmd --zone=public --add-service=http --permanent

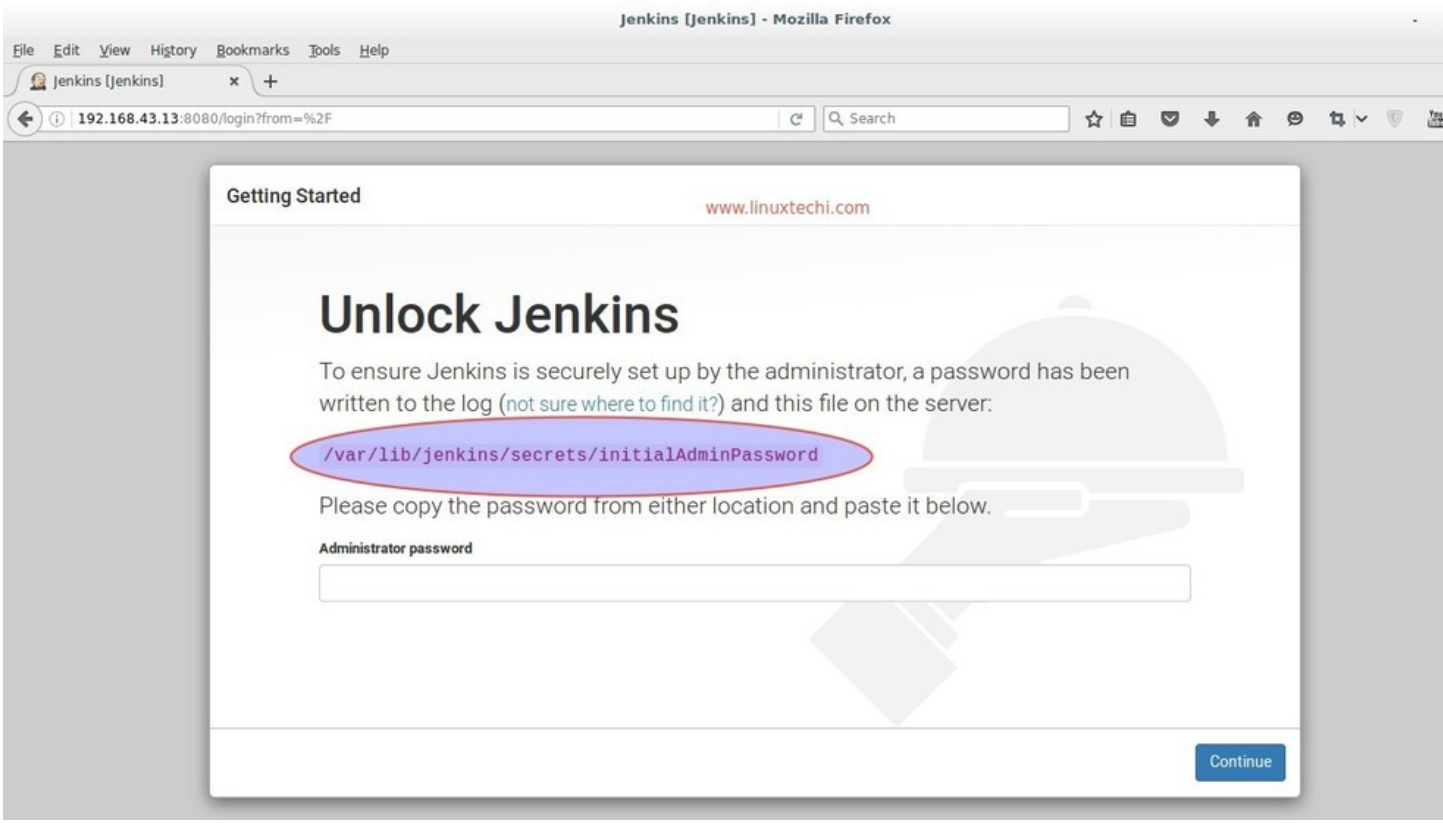
success

[root@jenkins ~]# firewall-cmd --reload

Success

Step:5 Access the Jenkins Web portal

Access the URL : http://<Ip-Address-of-your-Server>:8080

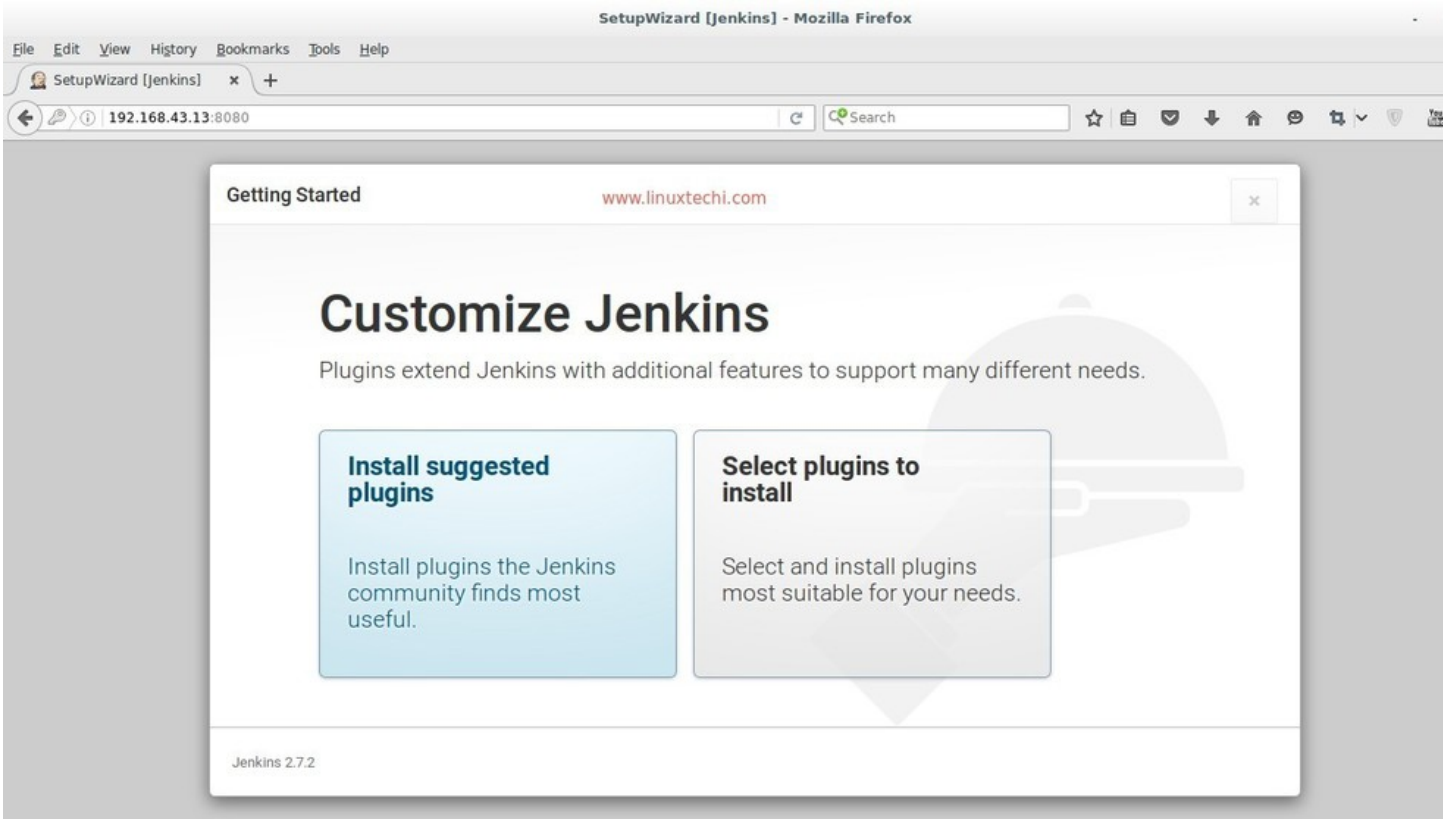


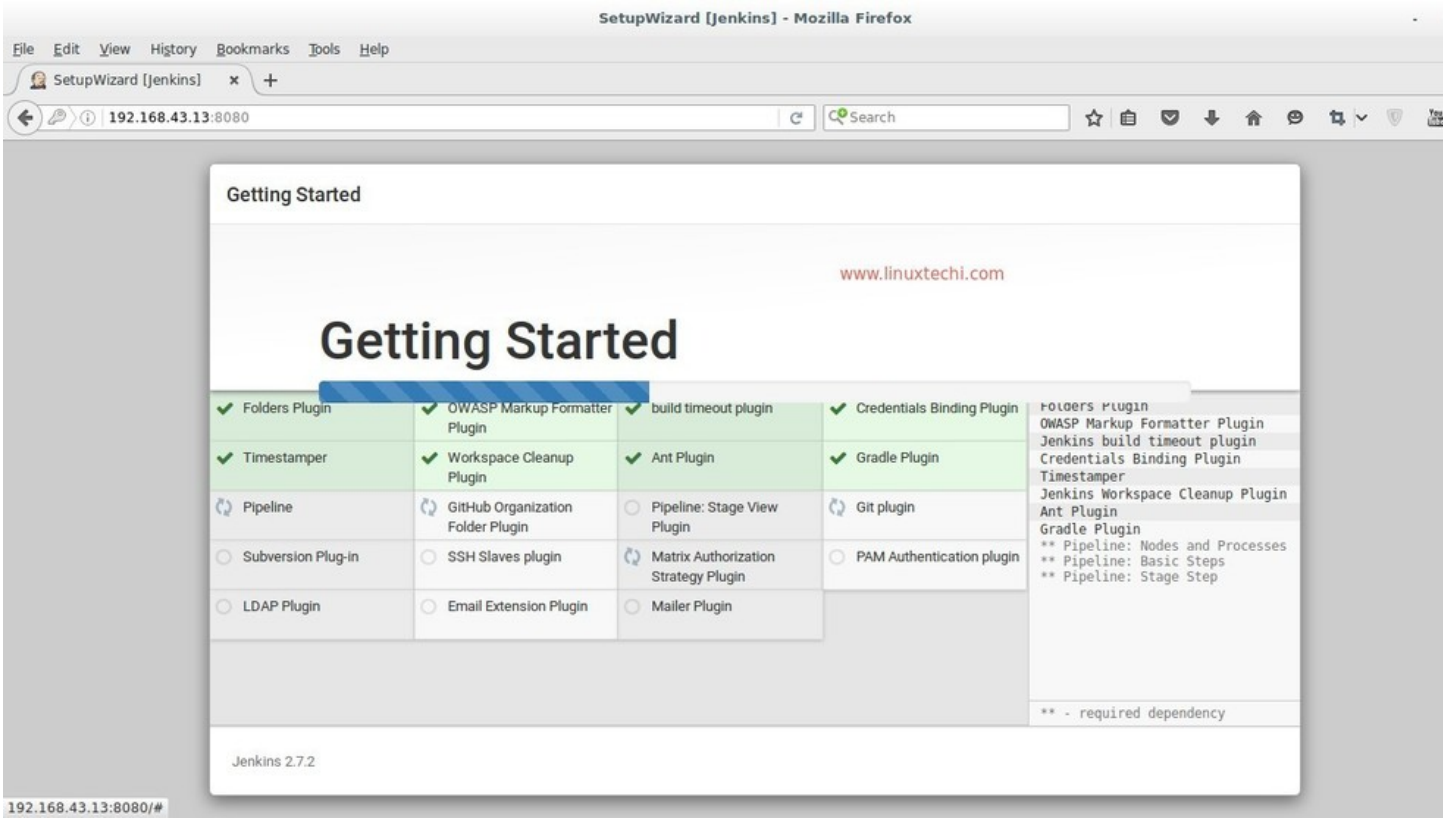
Admin password is created and stored in the log file "**/var/log/jenkins/jenkins.log**". Run the below command to get the password.

[root@jenkins ~]# grep -A 5 password /var/log/jenkins/jenkins.log

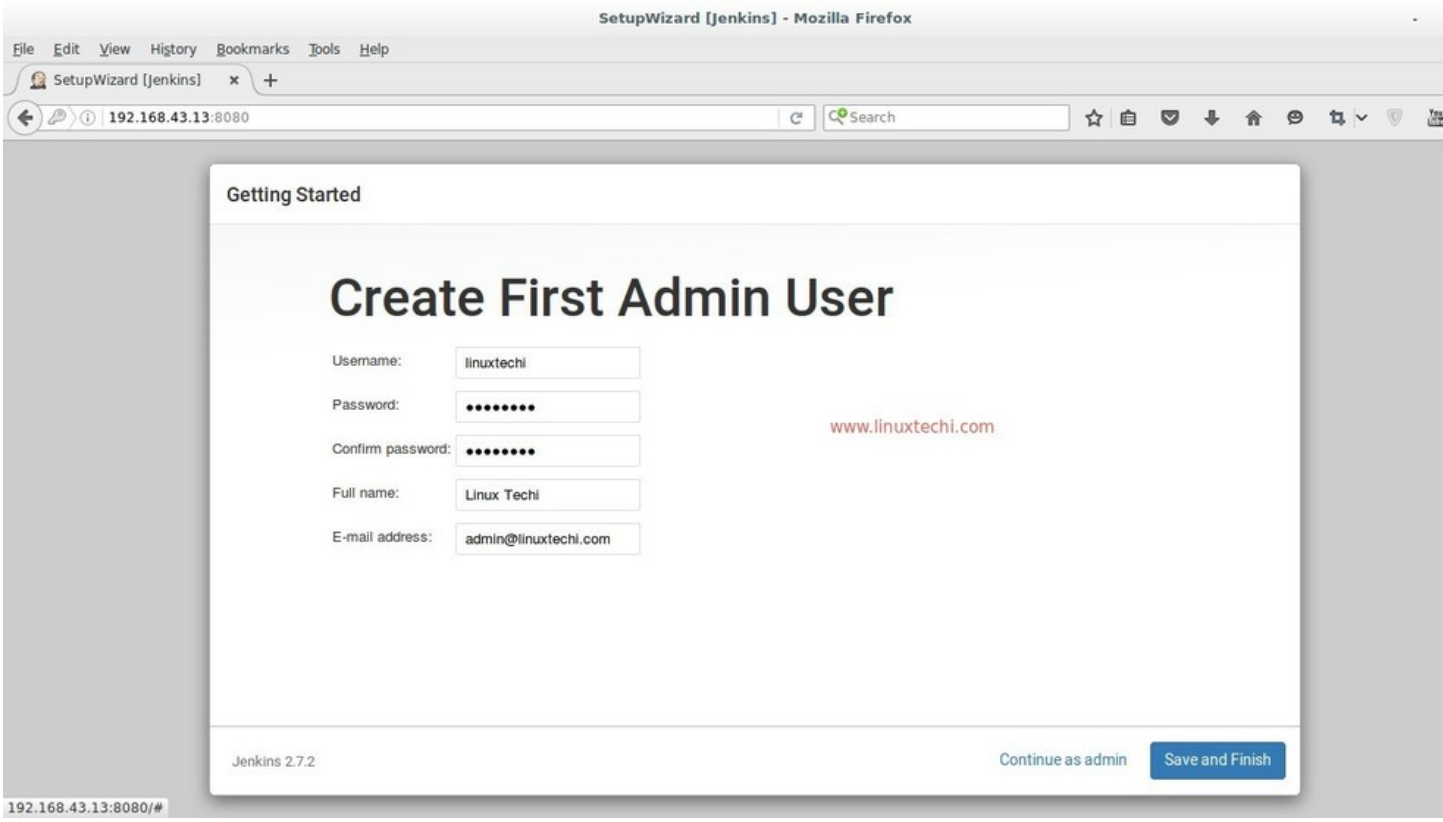
Copy the password and paste it in above windows and click on Continue..

In the next windows Select the option : **Install suggested plugins**

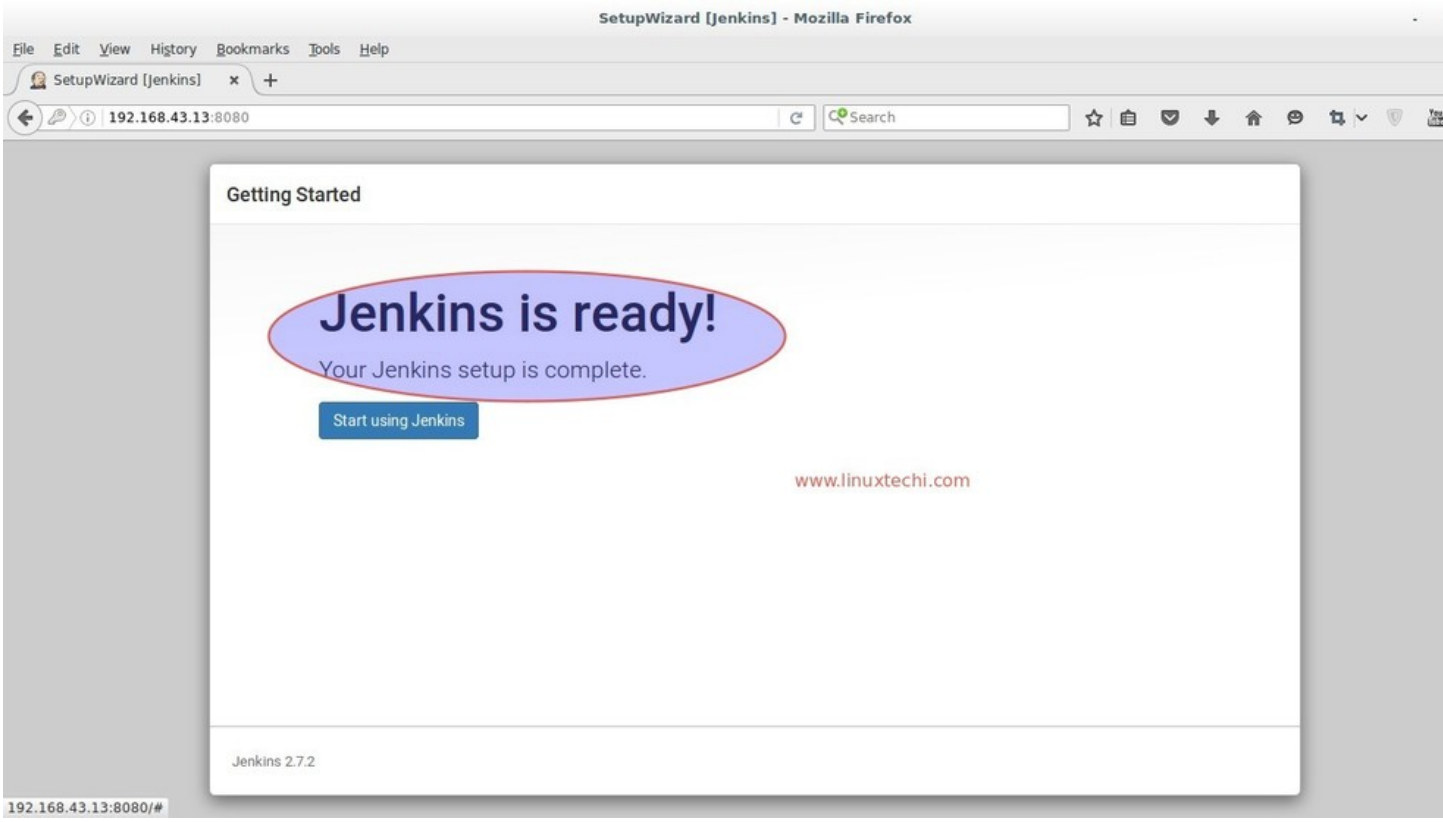




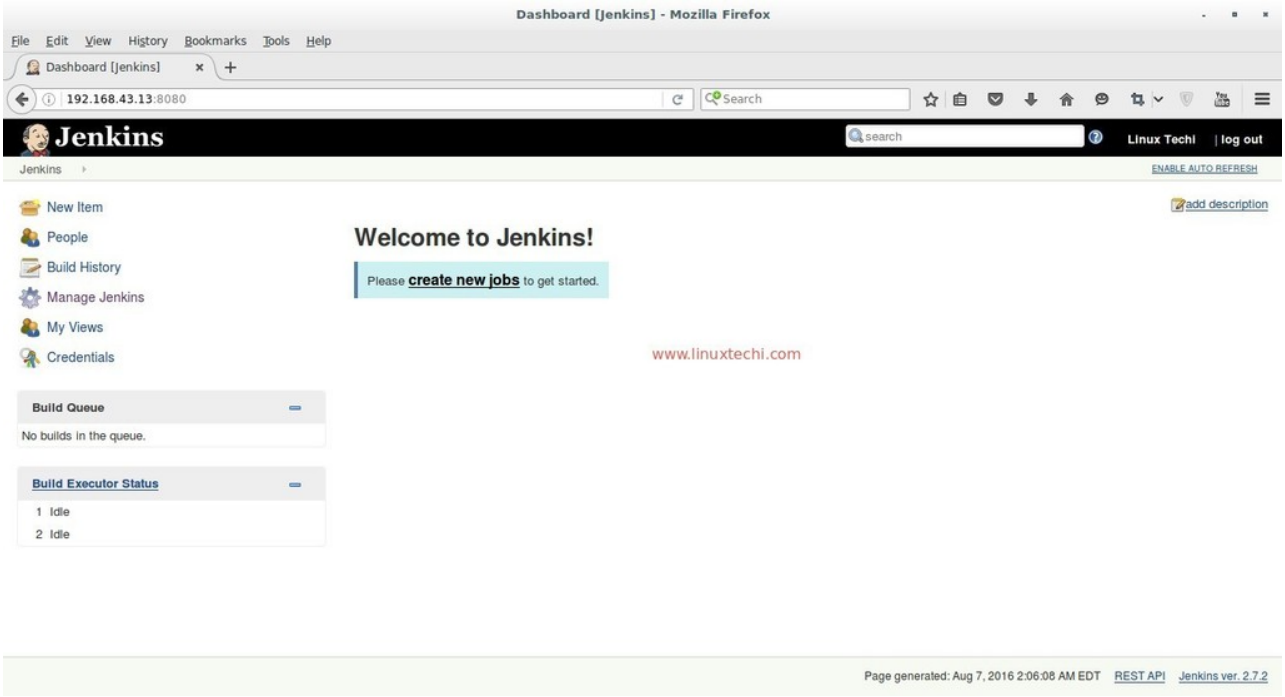
As we can see required plugin installation is in progress for Jenkins. Once it is done with plugin installation. It will ask to create Admin User



Click on Save and Finish



click on "**Start using Jenkins**"



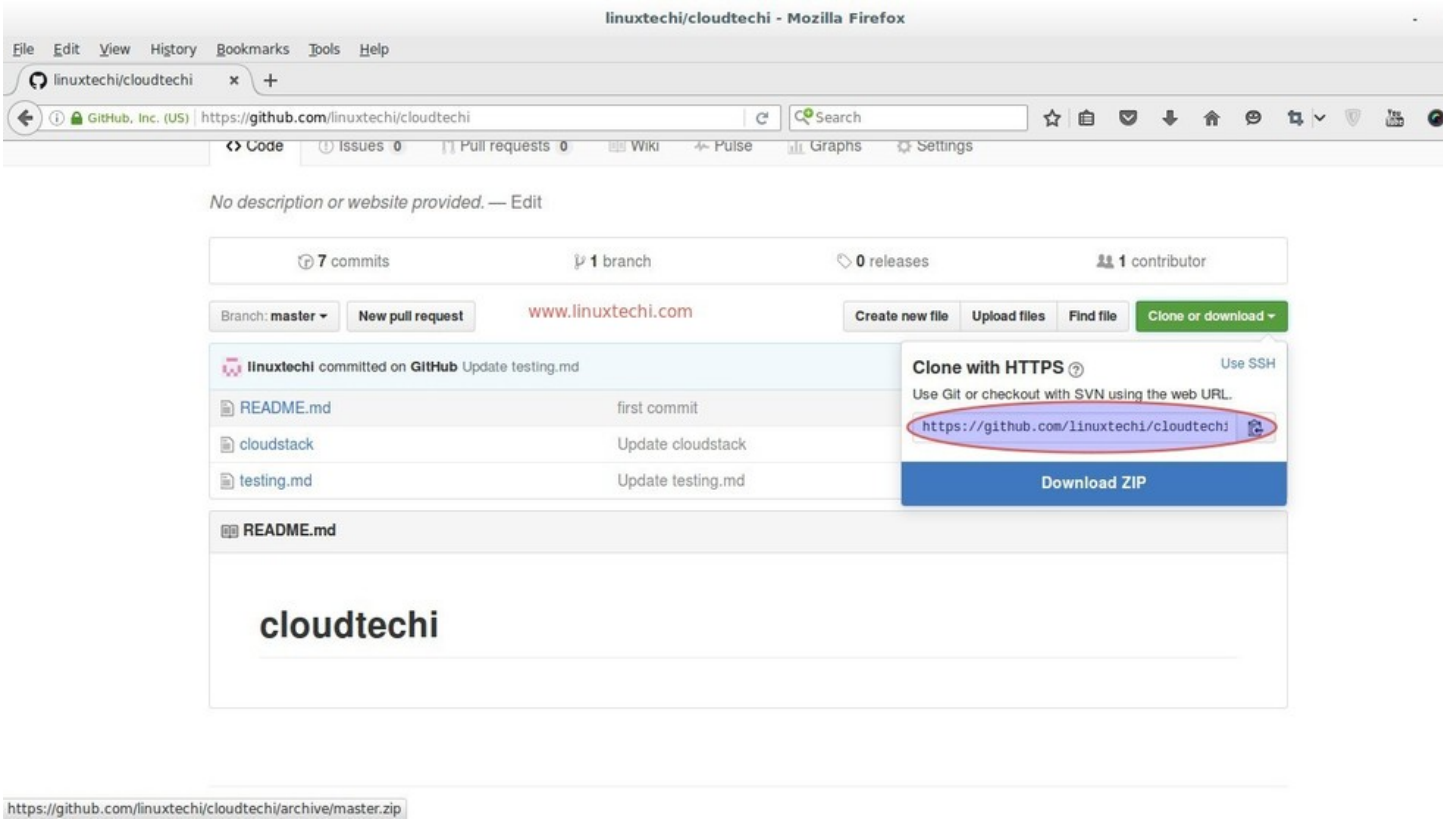
Now Configure GitHub project using git plugin in Jenkins.

Let's Assume I have a '**cloudtechi'** project on GitHub and wants to integrate this project in Jenkins using git plugin.

Let's first install git package on your machine on which you have installed Jenkins because Jenkins use git command to pull the GitHub project code.

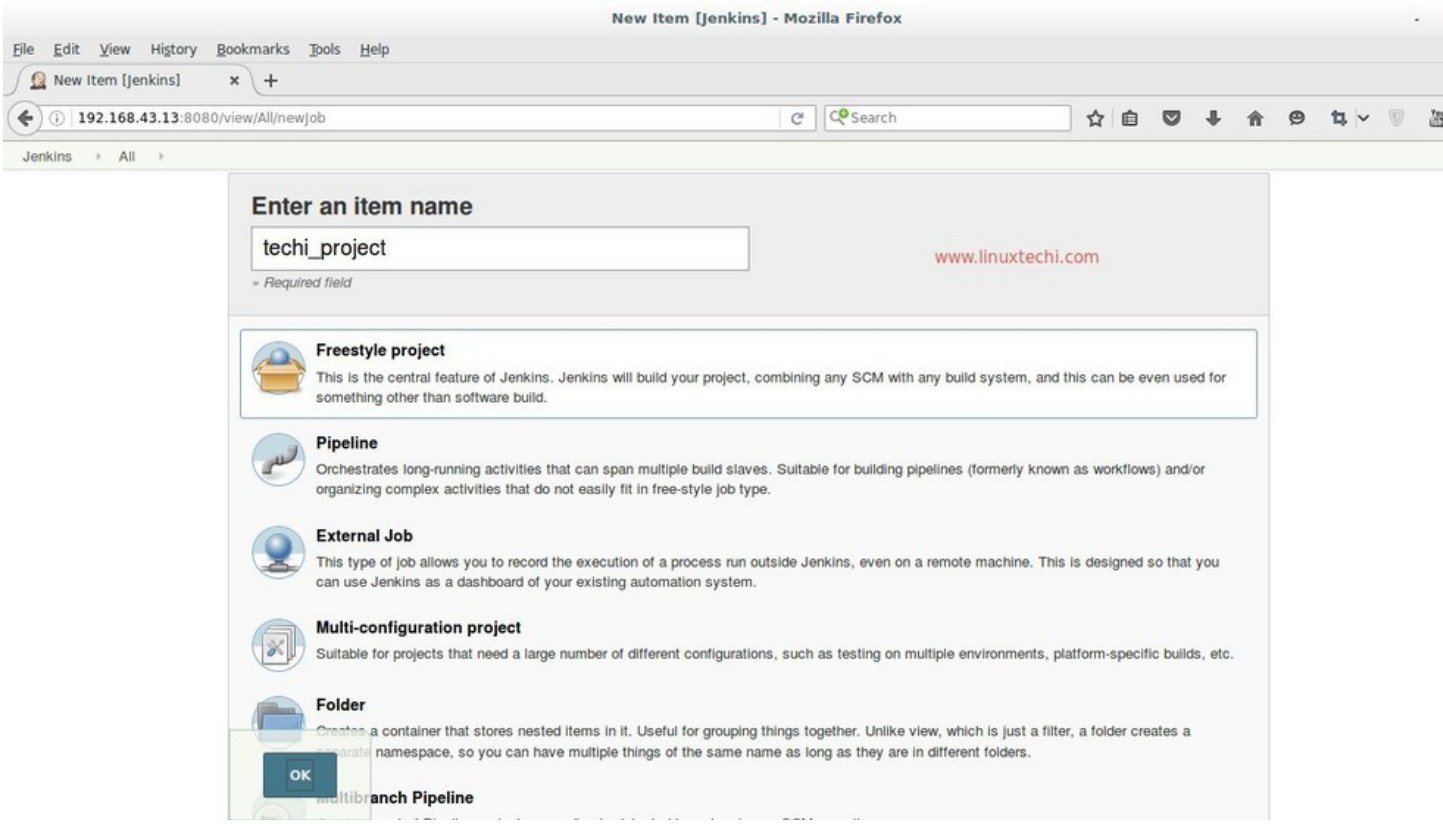
[root@jenkins ~]# yum install git

Login to the GitHub and get the Web URL of your project.



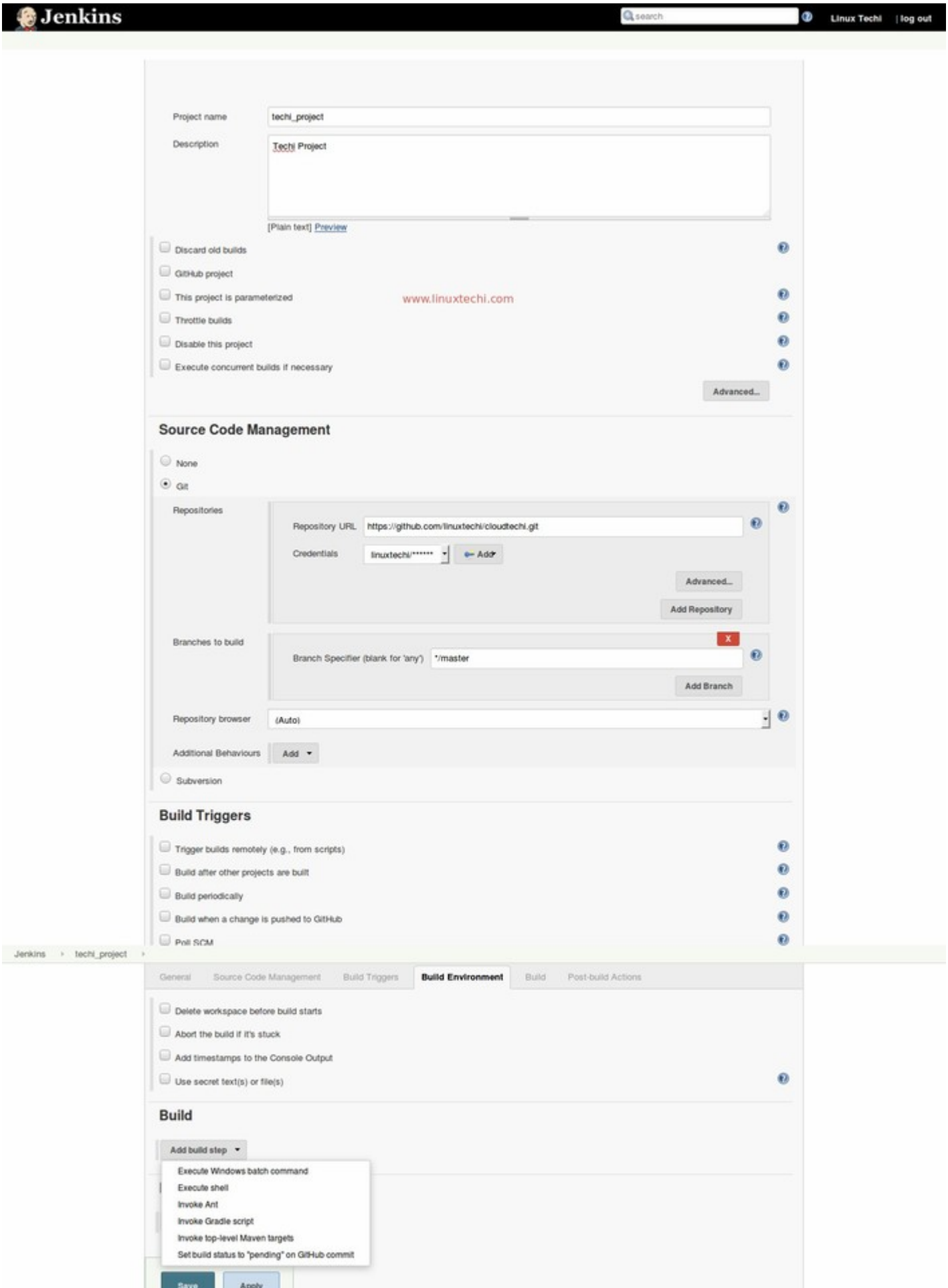
Login to the Jenkins portal, Click on "**New Item**"

Select the Freestyle Project and Specify the name as per your setup, In my case I putting as "**techi_project**"

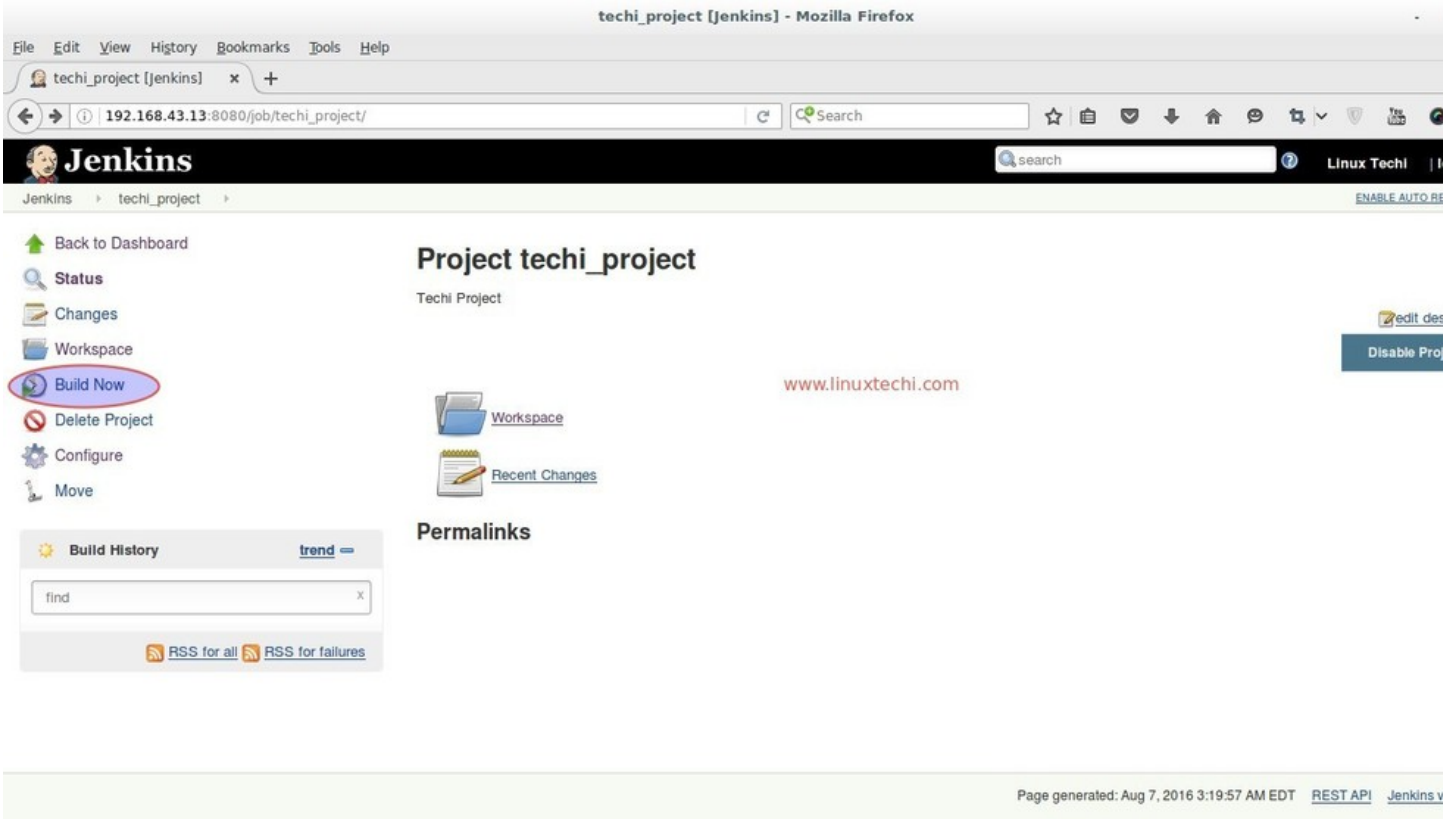


Click on OK

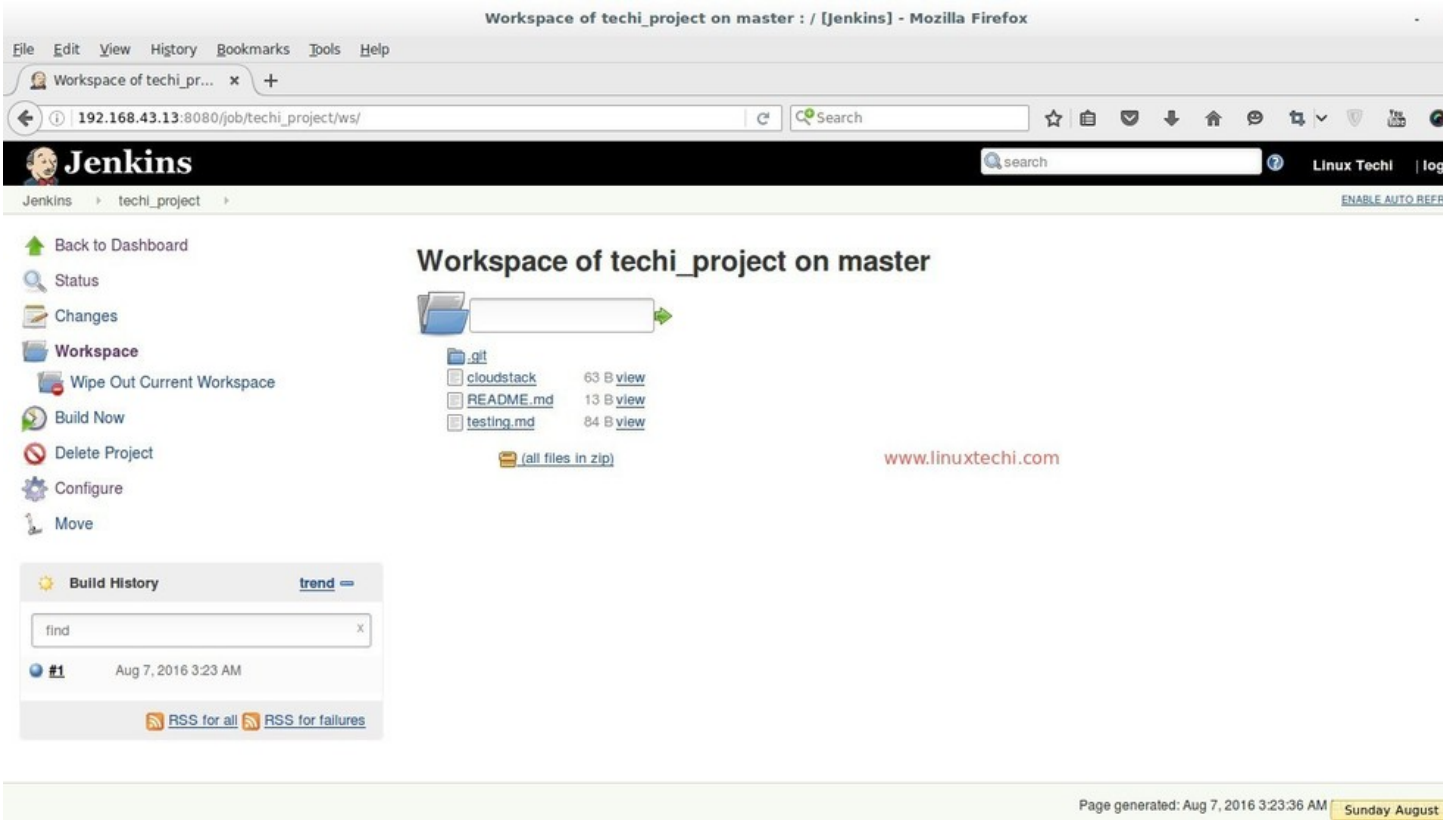
Specify the Project Description and Select Git option in Source Code Management Tab and specify the Web URL of your GitHub Project and its credentials. In the Build Tab select the option that suits your setup and then finally click on Apply.


Jenkins
search
Linux Techi
| log out
Project name
techi_project
Description
Techi Project
[Plain text] Preview
Discard old builds
GitHub project
This project is parameterized
www.linuxtechi.com
Throttle builds
Disable this project
Execute concurrent builds if necessary
Advanced...
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/linuxtechi/cloudtechi.git
Credentials
linuxtechi/******
Add
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/master
Add Branch
Repository browser
(Auto)
Additional Behaviours
Add
Subversion
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
Build when a change is pushed to GitHub
Poll SCM
Jenkins
techi_project
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Delete workspace before build starts
Abort the build if it's stuck
Add timestamps to the Console Output
Use secret text(s) or file(s)
Build
Add build step
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Set build status to "pending" on GitHub commit
Save
Apply

In the Next step click on the "**Build Now**" option from Jenkins Dashboard to pull the GitHub Project Code.



Click on the Workspace Option to view Code or files of GitHub Project. Whenever a new code is pushed on the GitHub project it will be automatically push to Jenkins workspace with new versions.



we can also view the project workspace from terminal as well.

```
[root@jenkins ~]# cd /var/lib/jenkins/workspace/techi_project/

[root@jenkins techi_project]# ls -l

total 12

-rw-r--r--. 1 jenkins jenkins 63 Aug  7 03:23 cloudstack

-rw-r--r--. 1 jenkins jenkins 13 Aug  7 03:23 README.md

-rw-r--r--. 1 jenkins jenkins 84 Aug  7 03:23 testing.md

[root@jenkins techi_project]#
```

We can this code to deploy on other machines as well. That’s all, Basic Jenkins installation and configuration is completed