Don Bosco Institute of Technology, Mumbai 400070
Department of Information Technology

Exp 2: Install and configure Jenkins

Batch : A
Roll No : 46

Aim: To Install and Configure Jenkins to test and deploy Java application and to create, evaluate &analyse versioning.

Theory: Jenkins is an open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat. It supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands.

Plugins have been released for Jenkins that extend its use to projects written in languages other than Java. Plugins are available for integrating Jenkins with most version control systems and bug databases. Many build tools are supported via their respective plugins. Plugins can also change the way Jenkins looks or add new functionality. There are a set of plugins dedicated forthe purpose of unit testing that generate test reports in various formats (for example, JUnit bundled with Jenkins, MSTest, NUnit, etc.) and automated testing that supports automated tests. Builds can generate test reports in various formats supported by plugins (JUnit support is currently bundled) and Jenkins can display the reports and generate trends and render them inthe GUI.

Procedure:

Configure your environment

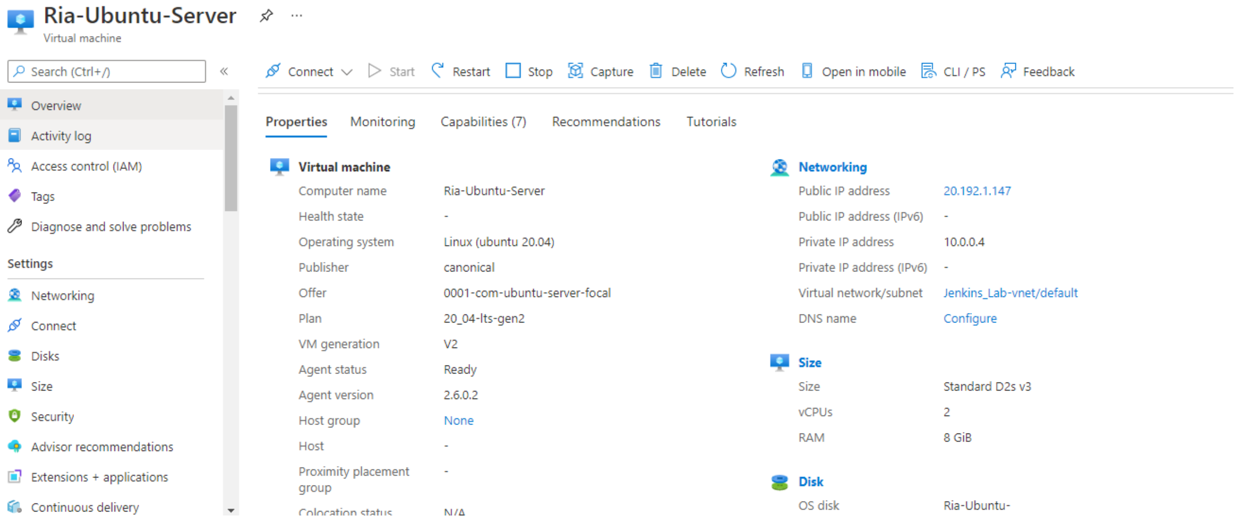
Azure subscription: If you don't have an Azure subscription, create a free account

(https://azure.microsoft.com/en-in/free/students/) before you begin.

Open Cloud Shell

1. If you already have a Cloud Shell session open, you can skip to the next section.
2. Browse to the Azure portal
3. If necessary, log in to your Azure subscription and change the Azure directory.
4. Open Cloud Shell.
5. If you haven't previously used Cloud Shell, configure the environment and storage settings.
6. Select the command-line environment.

Create a virtual machine



Virtual machine

Computer name	Ria-Ubuntu-Server
Health state	-
Operating system	Linux (ubuntu 20.04)
Publisher	canonical
Offer	0001-com-ubuntu-server-focal
Plan	20_04-lts-gen2
VM generation	V2
Agent status	Ready
Agent version	2.6.0.2
Host group	None
Host	-
Proximity placement group	-
Colocation status	N/A

Networking

Public IP address	20.192.1.147
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	Jenkins_Lab-vnet/default
DNS name	Configure

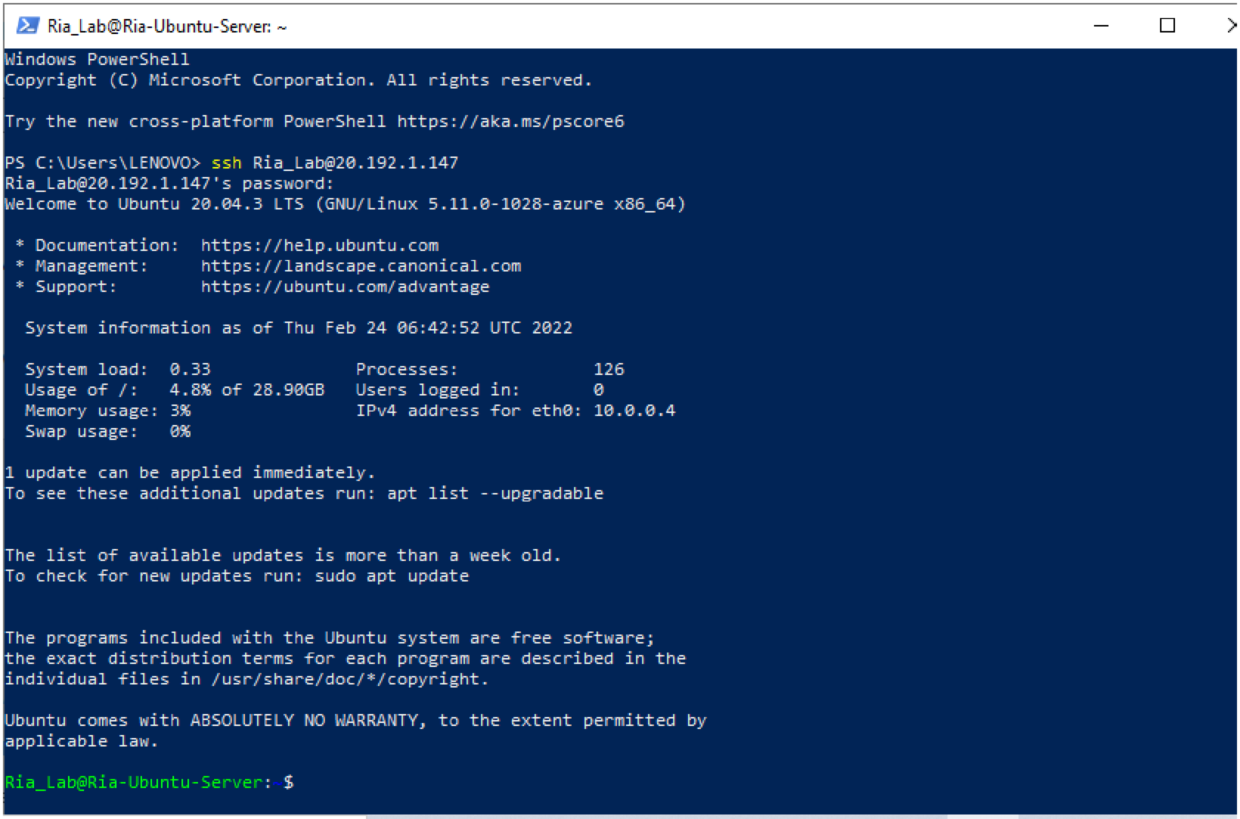
Size

Size	Standard D2s v3
vCPUs	2
RAM	8 GiB

Disk

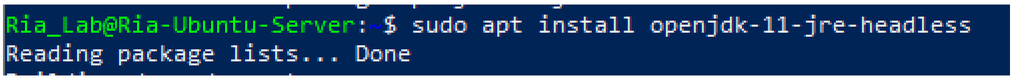
OS disk	Ria-Ubuntu-

Login to your virtual machine



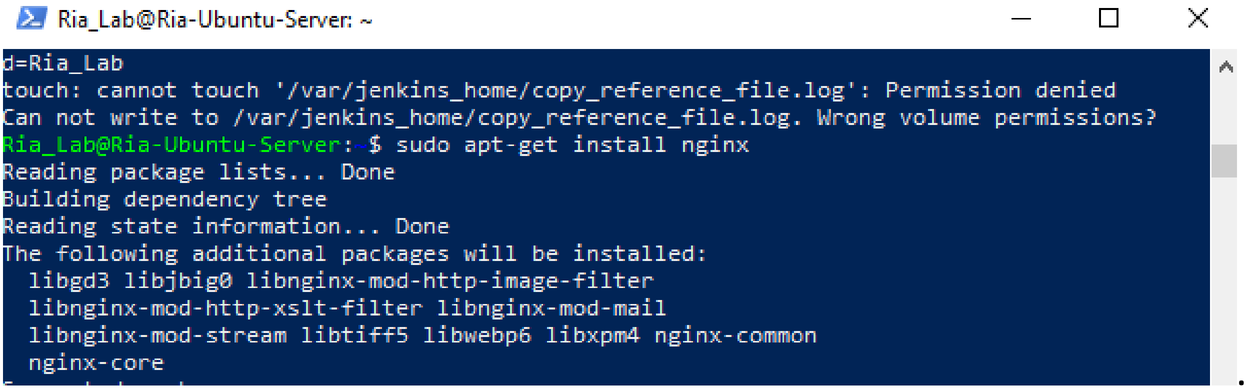
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\LENOVO> ssh Ria_Lab@20.192.1.147
Ria_Lab@20.192.1.147's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1028-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Feb 24 06:42:52 UTC 2022

  System load:  0.33              Processes:             126
  Usage of /:   4.8% of 28.90GB   Users logged in:       0
  Memory usage: 3%                IPv4 address for eth0: 10.0.0.4
  Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Ria_Lab@Ria-Ubuntu-Server: $
```

Install Java



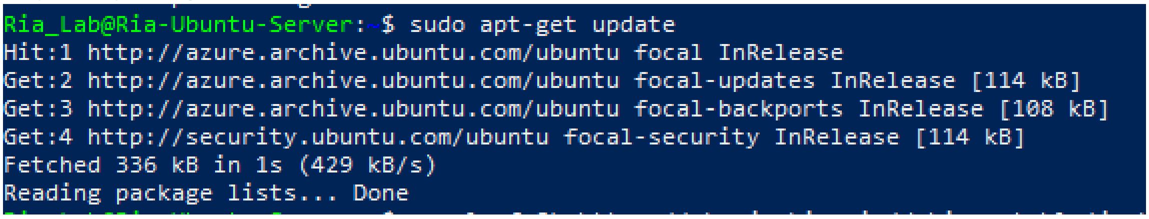
```
Ria_Lab@Ria-Ubuntu-Server:~$ sudo apt install openjdk-11-jre-headless
Reading package lists... Done
```

Install Nginx



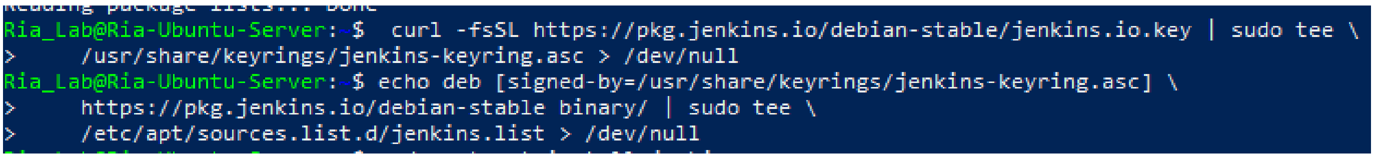
```
d=Ria_Lab
touch: cannot touch '/var/jenkins_home/copy_reference_file.log': Permission denied
Can not write to /var/jenkins_home/copy_reference_file.log. Wrong volume permissions?
Ria_Lab@Ria-Ubuntu-Server:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libgd3 libjbig0 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail
  libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx-common
  nginx-core
```

Update packages

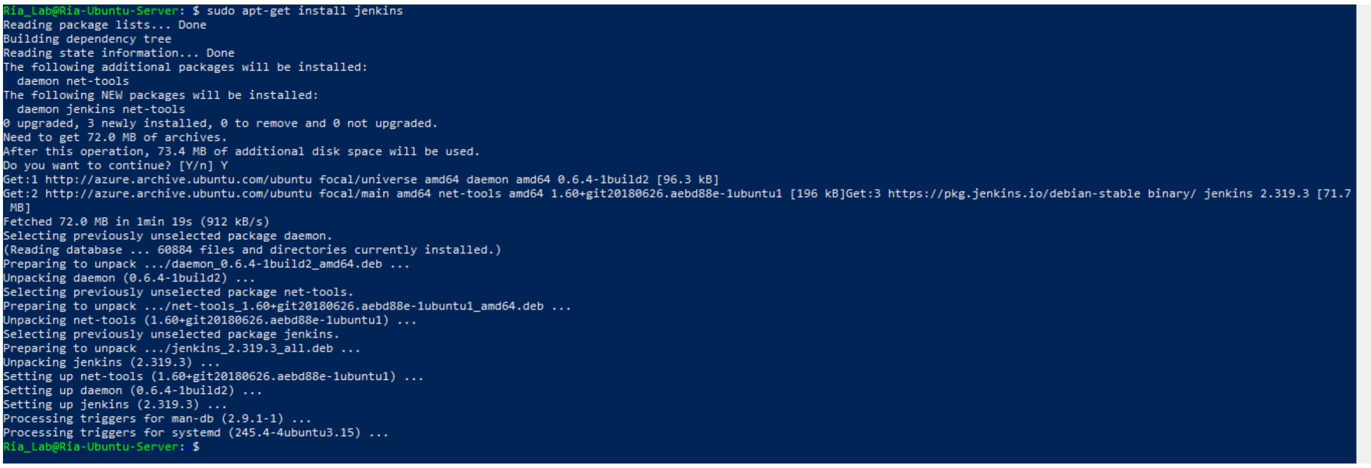


```
Ria_Lab@Ria-Ubuntu-Server:~$ sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 1s (429 kB/s)
Reading package lists... Done
```

Install jenkins package from: https://www.jenkins.io/doc/book/installing/linux/#debianubuntu

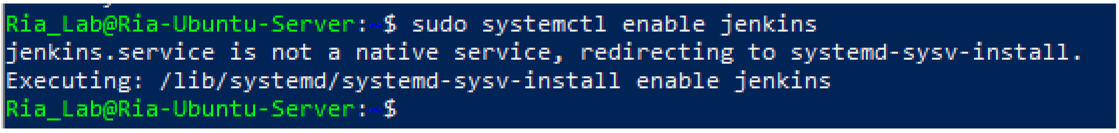


```
Reading package lists... Done
Ria_Lab@Ria-Ubuntu-Server:~$  curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
>     /usr/share/keyrings/jenkins-keyring.asc > /dev/null
Ria_Lab@Ria-Ubuntu-Server:~$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
>     https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
>     /etc/apt/sources.list.d/jenkins.list > /dev/null
```



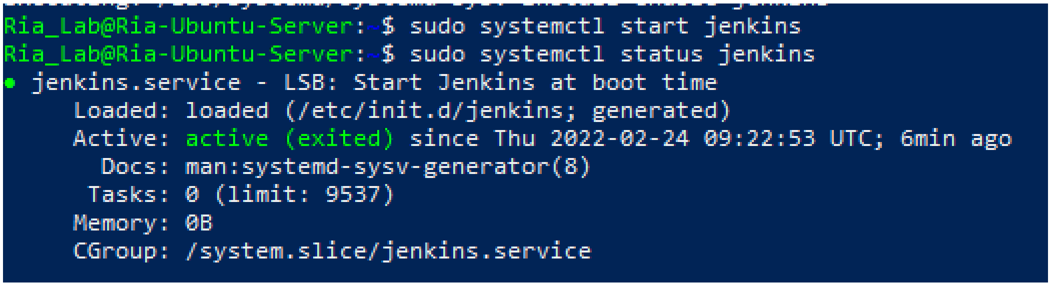
```
Ria_Lab@Ria-Ubuntu-Server:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon net-tools
The following NEW packages will be installed:
  daemon jenkins net-tools
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 72.0 MB of archives.
After this operation, 73.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 daemon amd64 0.6.4-1build2 [96.3 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]Get:3 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.319.3 [71.7
 MB]
Fetched 72.0 MB in 1min 19s (912 kB/s)
Selecting previously unselected package daemon.
(Reading database ... 60884 files and directories currently installed.)
Preparing to unpack .../daemon_0.6.4-1build2_amd64.deb ...
Unpacking daemon (0.6.4-1build2) ...
Selecting previously unselected package net-tools.
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.319.3_all.deb ...
Unpacking jenkins (2.319.3) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up daemon (0.6.4-1build2) ...
Setting up jenkins (2.319.3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.15) ...
Ria_Lab@Ria-Ubuntu-Server:~$
```

Enable jenkins



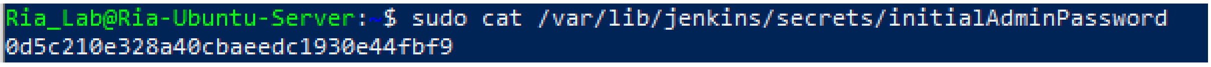
```
Ria_Lab@Ria-Ubuntu-Server:~$ sudo systemctl enable jenkins
jenkins.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
Ria_Lab@Ria-Ubuntu-Server:~$
```

Start Jenkins and check status:



```
Ria_Lab@Ria-Ubuntu-Server:~$ sudo systemctl start jenkins
Ria_Lab@Ria-Ubuntu-Server:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
     Loaded: loaded (/etc/init.d/jenkins; generated)
     Active: active (exited) since Thu 2022-02-24 09:22:53 UTC; 6min ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 0 (limit: 9537)
     Memory: 0B
     CGroup: /system.slice/jenkins.service
```

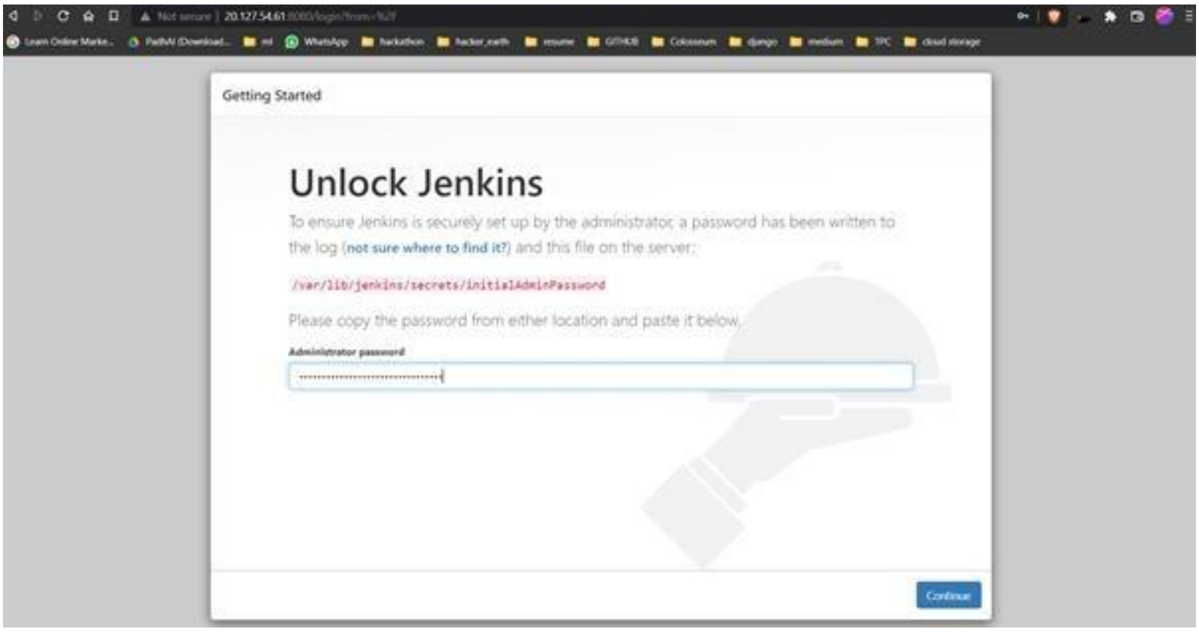
Generate Admin Password:



```
Ria_Lab@Ria-Ubuntu-Server:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
0d5c210e328a40cbaeedc1930e44fbf9
```

3: Go to the URL: http://<ip_address>:8080 and setup Jenkins

Enter the password generated before and continue:



Not secure | 20.127.54.61:8080/login?from=%2F

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

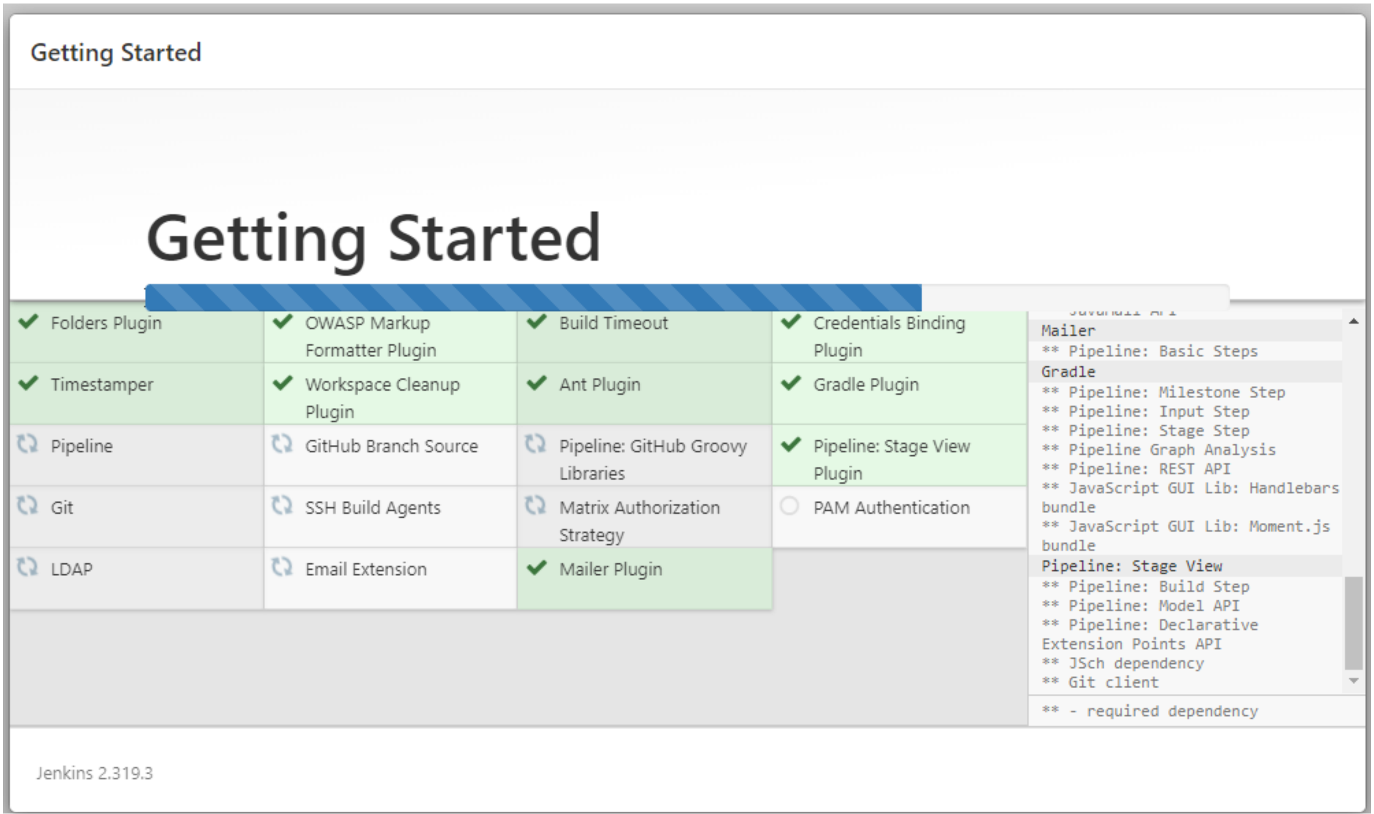
/var/lib/jenkins/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password

••••••••••••••••••••••••••••••••

Install Additional packages



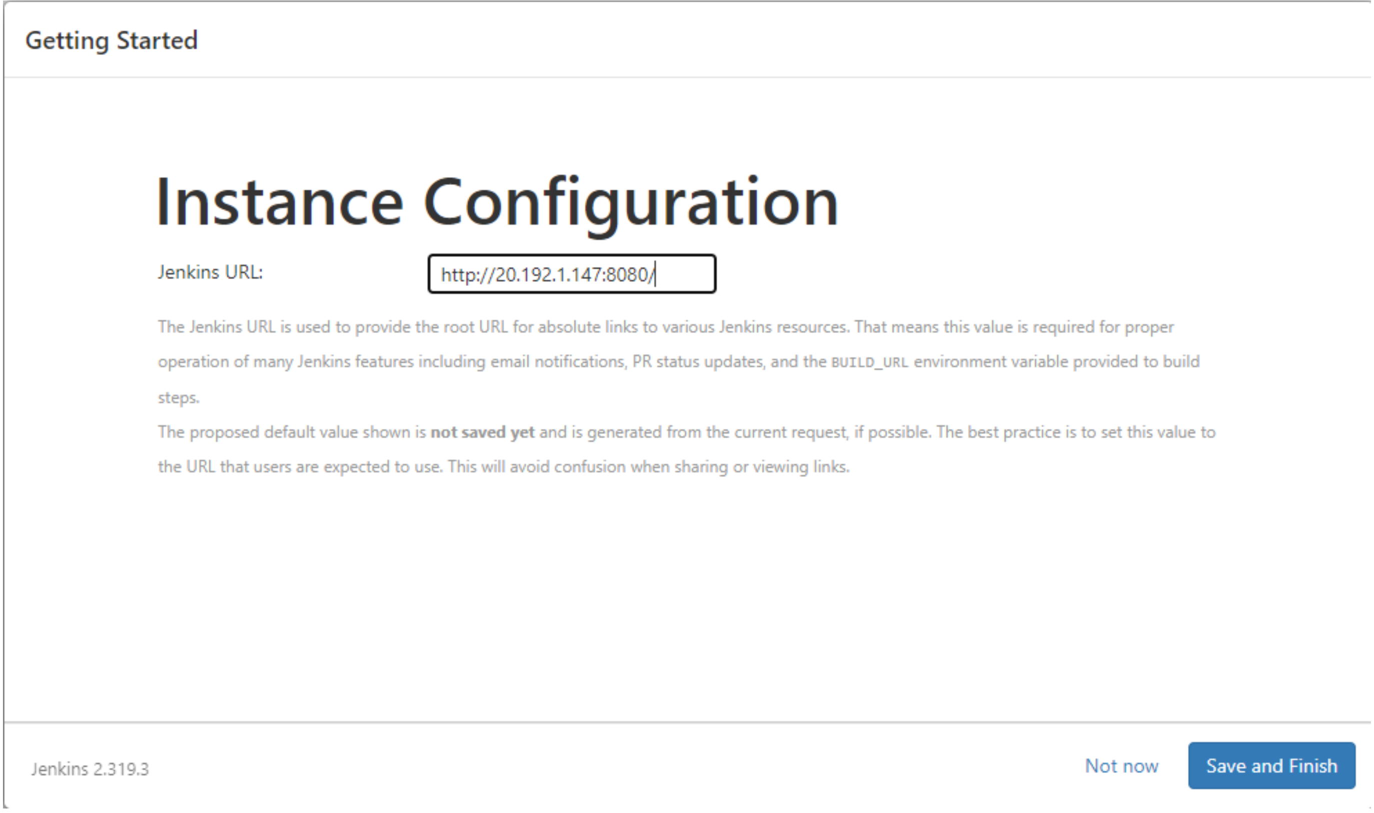
Getting Started

Getting Started

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding Plugin
✓ Timestamper	✓ Workspace Cleanup Plugin	✓ Ant Plugin	✓ Gradle Plugin
⟳ Pipeline	⟳ GitHub Branch Source	⟳ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View Plugin
⟳ Git	⟳ SSH Build Agents	⟳ Matrix Authorization Strategy	○ PAM Authentication
⟳ LDAP	⟳ Email Extension	✓ Mailer Plugin	

```
Mailer
** Pipeline: Basic Steps
Gradle
** Pipeline: Milestone Step
** Pipeline: Input Step
** Pipeline: Stage Step
** Pipeline Graph Analysis
** Pipeline: REST API
** JavaScript GUI Lib: Handlebars
bundle
** JavaScript GUI Lib: Moment.js
bundle
Pipeline: Stage View
** Pipeline: Build Step
** Pipeline: Model API
** Pipeline: Declarative
Extension Points API
** JSch dependency
** Git client
** - required dependency
```

Jenkins 2.319.3



Getting Started

Instance Configuration

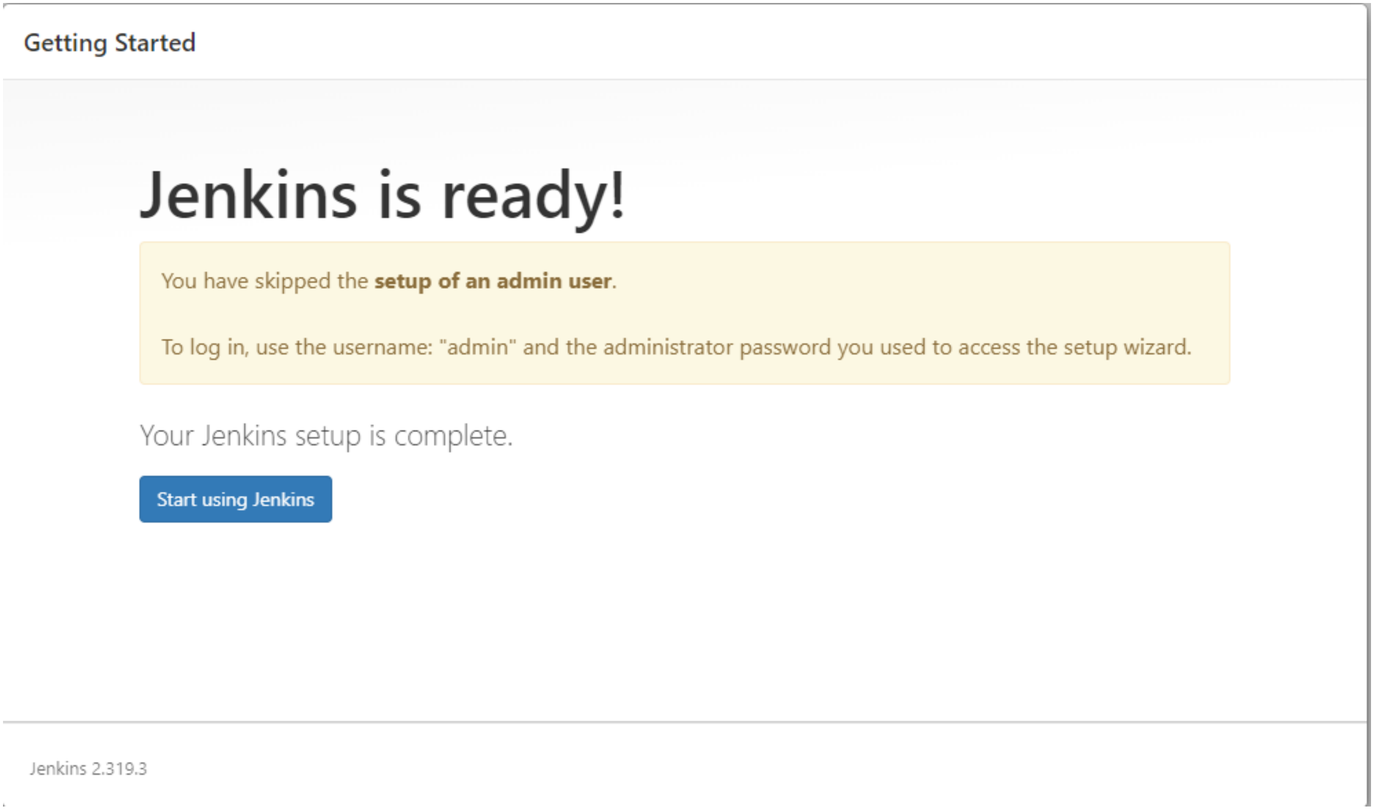
Jenkins URL: http://20.192.1.147:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.319.3

Jenkins is ready to use!



Getting Started

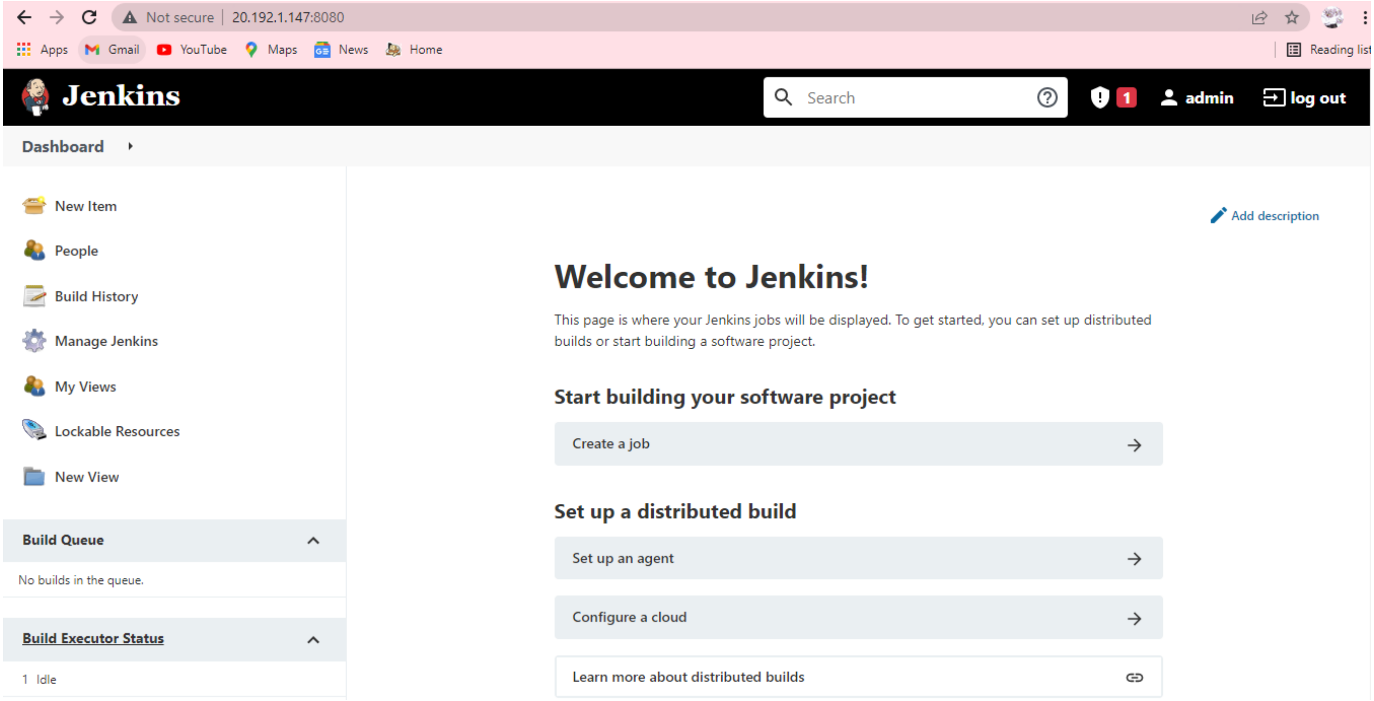
Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

Jenkins Dashboard:



No builds in the queue.

1 Idle

Welcome to Jenkins!

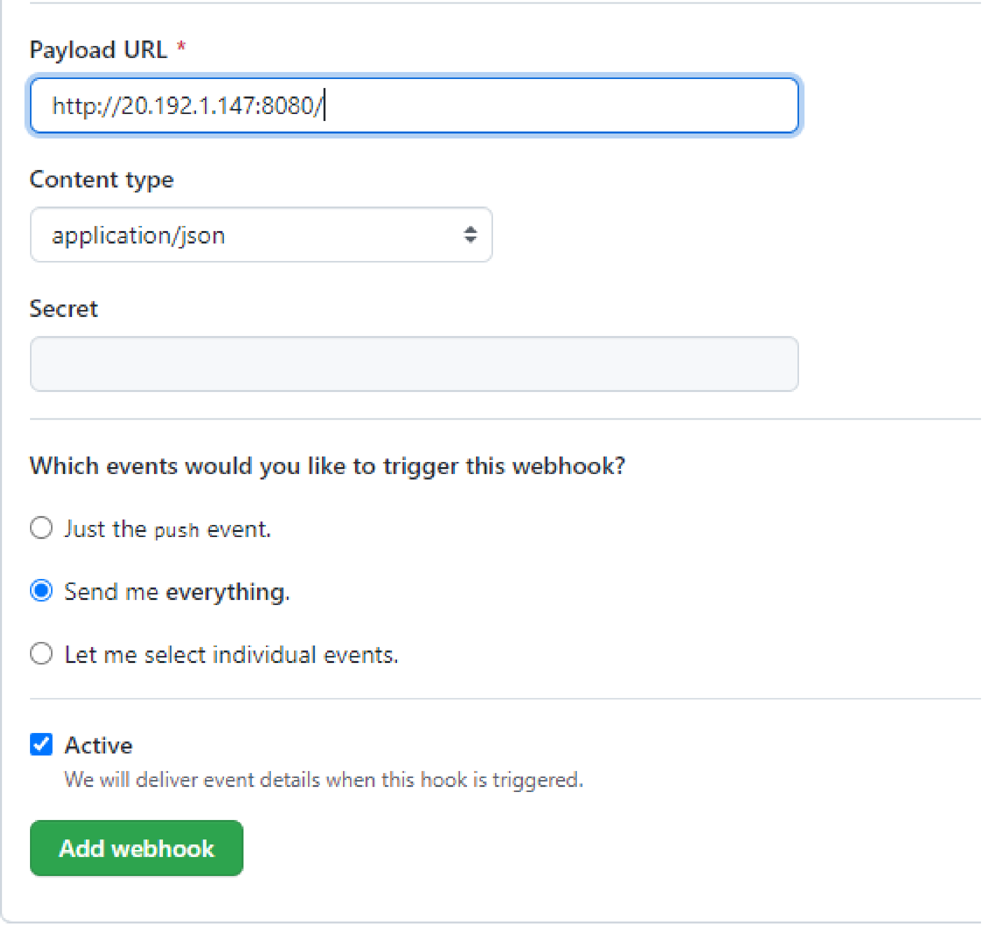
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

4 Link with GitHub:

Open the GitHub repository and add a new webhook



Payload URL *

http://20.192.1.147:8080/

Content type

application/json

Secret

Which events would you like to trigger this webhook?

○ Just the push event.

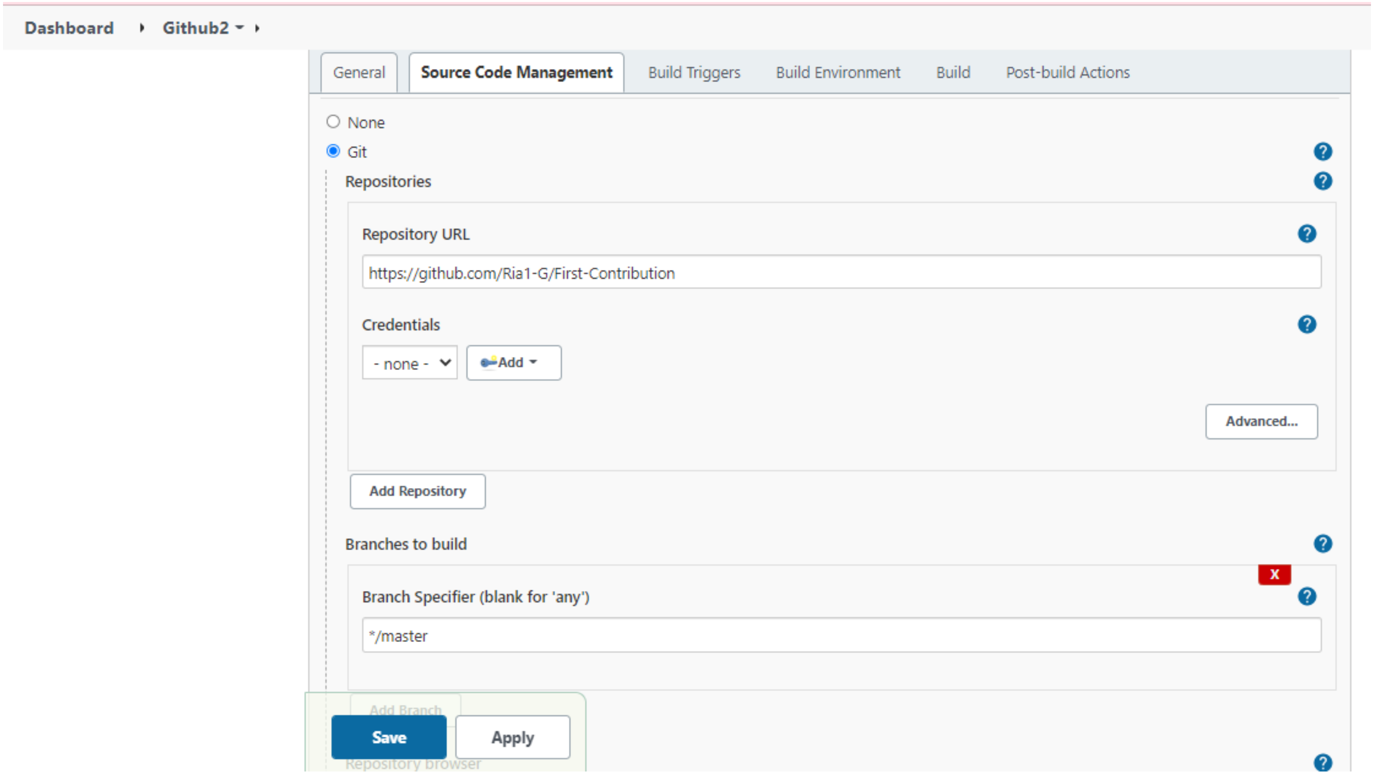
● Send me everything.

○ Let me select individual events.

☑ Active

We will deliver event details when this hook is triggered.

Then go to Jenkins and create a new project



○ None

● Git

Repositories

Repository URL

https://github.com/Ria1-G/First-Contribution

Credentials

- none -

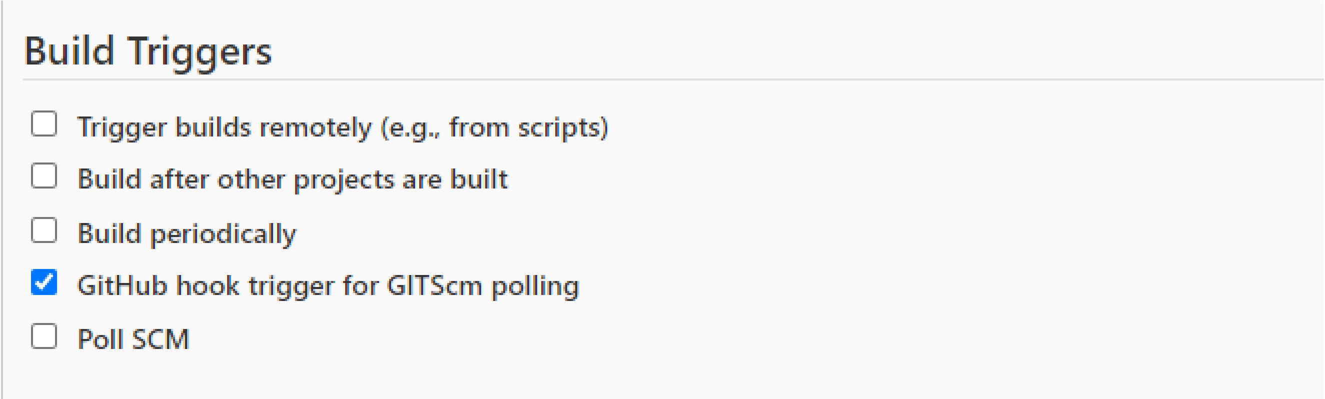
Branches to build

Branch Specifier (blank for 'any')

*/master

Enter the repository URL in source code management

Enable Git Hook trigger



Build Triggers

□ Trigger builds remotely (e.g., from scripts)

□ Build after other projects are built

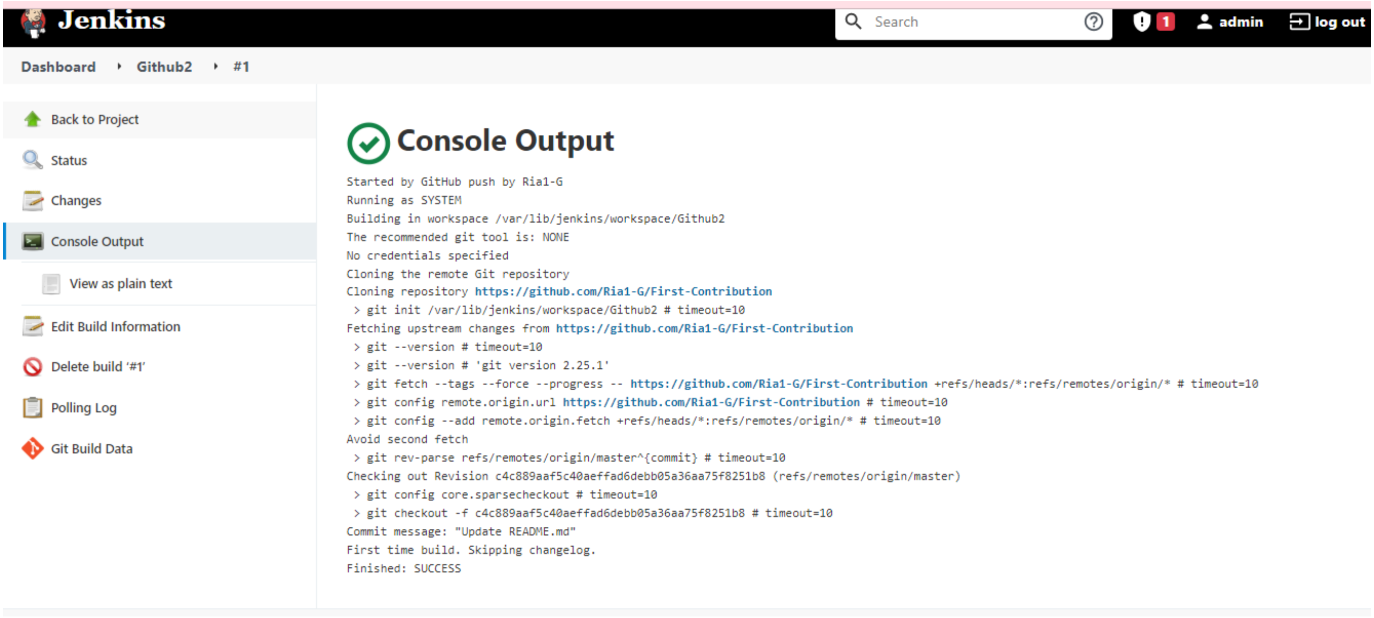
□ Build periodically

☑ GitHub hook trigger for GITScm polling

□ Poll SCM

Then save the project

Now add a new commit on GitHub and view the commit details on GitHub console output.



Console Output

```
Started by GitHub push by Ria1-G
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Github2
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Ria1-G/First-Contribution
 > git init /var/lib/jenkins/workspace/Github2 # timeout=10
Fetching upstream changes from https://github.com/Ria1-G/First-Contribution
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- https://github.com/Ria1-G/First-Contribution +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Ria1-G/First-Contribution # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision c4c889aaf5c40aeffad6debb05a36aa75f8251b8 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f c4c889aaf5c40aeffad6debb05a36aa75f8251b8 # timeout=10
Commit message: "Update README.md"
First time build. Skipping changelog.
Finished: SUCCESS
```

Conclusion: The installation and configuration of Jenkins on Azure VM has been performed successfully and is linked to GitHub using webhooks.

References:

https://medium.com/bb-tutorials-and-thoughts/how-to-run-jenkins-on-azure-vm-73e24804730d

https://www.jenkins.io/doc/book/installing/linux/#debianubuntu

https://www.blazemeter.com/blog/how-to-integrate-your-github-repository-to-your-jenkins-project