Table of Contents

Installation

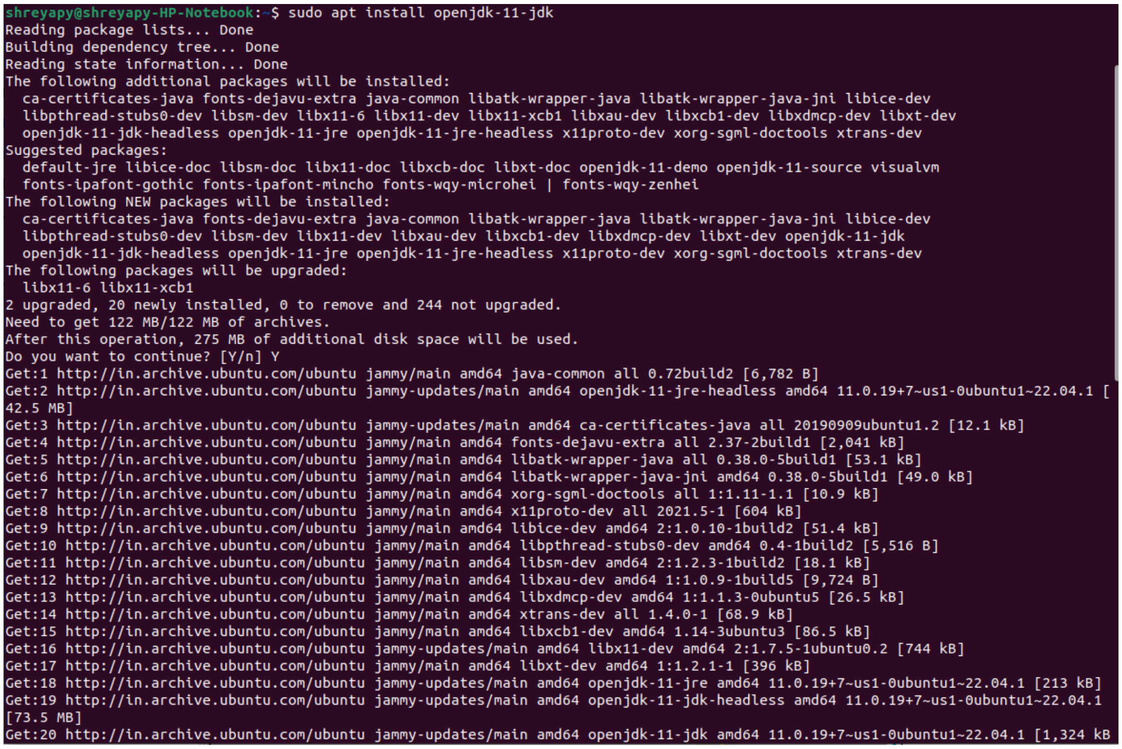
Before demonstrating the continuous integration and delivery by building a Docker Jenkins Pipeline, it is necessary to perform the installation of each program that needs to be used.

Because of that, it is necessary to install Jenkins, Docker and Git.

Java RunTime

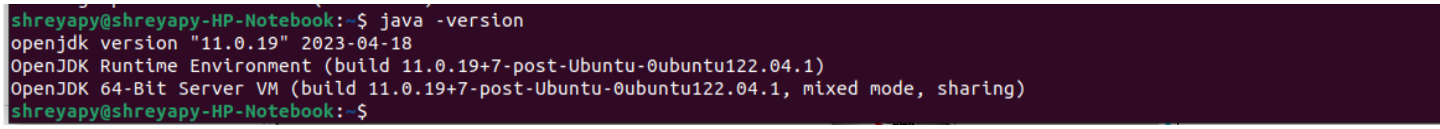
First important item about these processes, JRE is very important for all artifacts, because of this, this is the first step.

sudo apt-get update



```
shreyapy@shreyapy-HP-Notebook:~$ sudo apt install openjdk-11-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni libice-dev
  libpthread-stubs0-dev libsm-dev libx11-6 libx11-dev libx11-xcb1 libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
  openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc openjdk-11-demo openjdk-11-source visualvm
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java libatk-wrapper-java-jni libice-dev
  libpthread-stubs0-dev libsm-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-11-jdk
  openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless x11proto-dev xorg-sgml-doctools xtrans-dev
The following packages will be upgraded:
  libx11-6 libx11-xcb1
2 upgraded, 20 newly installed, 0 to remove and 244 not upgraded.
Need to get 122 MB/122 MB of archives.
After this operation, 275 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 java-common all 0.72build2 [6,782 B]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jre-headless amd64 11.0.19+7~us1-0ubuntu1~22.04.1 [
42.5 MB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ca-certificates-java all 20190909ubuntu1.2 [12.1 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2,041 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java-jni amd64 0.38.0-5build1 [49.0 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 xorg-sgml-doctools all 1:1.11-1.1 [10.9 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 x11proto-dev all 2021.5-1 [604 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libice-dev amd64 2:1.0.10-1build2 [51.4 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libpthread-stubs0-dev amd64 0.4-1build2 [5,516 B]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libsm-dev amd64 2:1.2.3-1build2 [18.1 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxau-dev amd64 1:1.0.9-1build5 [9,724 B]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxdmcp-dev amd64 1:1.1.3-0ubuntu5 [26.5 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 xtrans-dev all 1.4.0-1 [68.9 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxcb1-dev amd64 1.14-3ubuntu3 [86.5 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libx11-dev amd64 2:1.7.5-1ubuntu0.2 [744 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libxt-dev amd64 1:1.2.1-1 [396 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jre amd64 11.0.19+7~us1-0ubuntu1~22.04.1 [213 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jdk-headless amd64 11.0.19+7~us1-0ubuntu1~22.04.1
[73.5 MB]
Get:20 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openjdk-11-jdk amd64 11.0.19+7~us1-0ubuntu1~22.04.1 [1,324 kB
```

Is possible to verify what is the installed version of Java with code *java -version*



```
shreyapy@shreyapy-HP-Notebook:~$ java -version
openjdk version "11.0.19" 2023-04-18
OpenJDK Runtime Environment (build 11.0.19+7-post-Ubuntu-0ubuntu122.04.1)
OpenJDK 64-Bit Server VM (build 11.0.19+7-post-Ubuntu-0ubuntu122.04.1, mixed mode, sharing)
shreyapy@shreyapy-HP-Notebook:~$
```

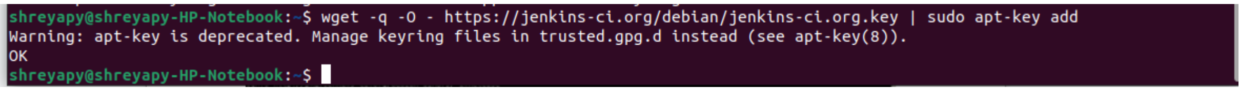
Jenkins Instalation

On the same terminal that executes the java installation, can be execute the Jenkins installation, with the command:

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

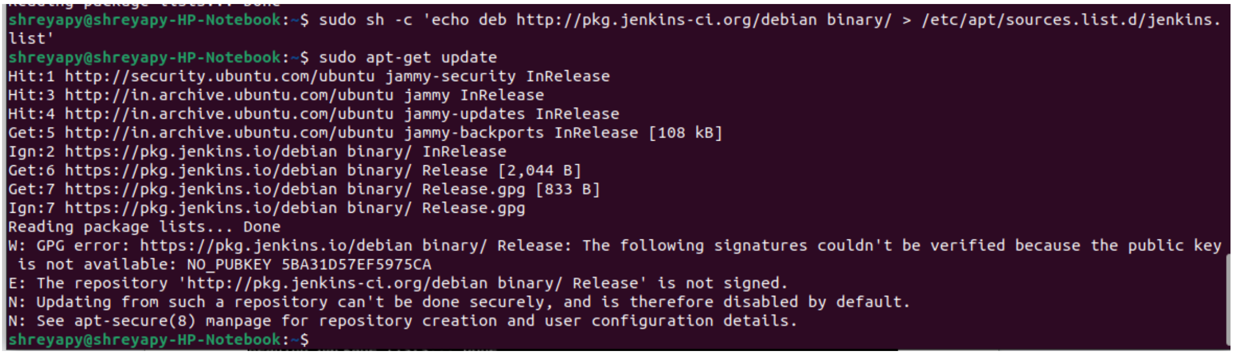
add if receives the message: *gpg: no valid OpenPGP data found.*

This problem might occur if you are behind corporate proxy and corporation uses its own certificate. Just add "--no-check-certificate" in the command. e.g. ***wget --no-check-certificate -qO - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add***



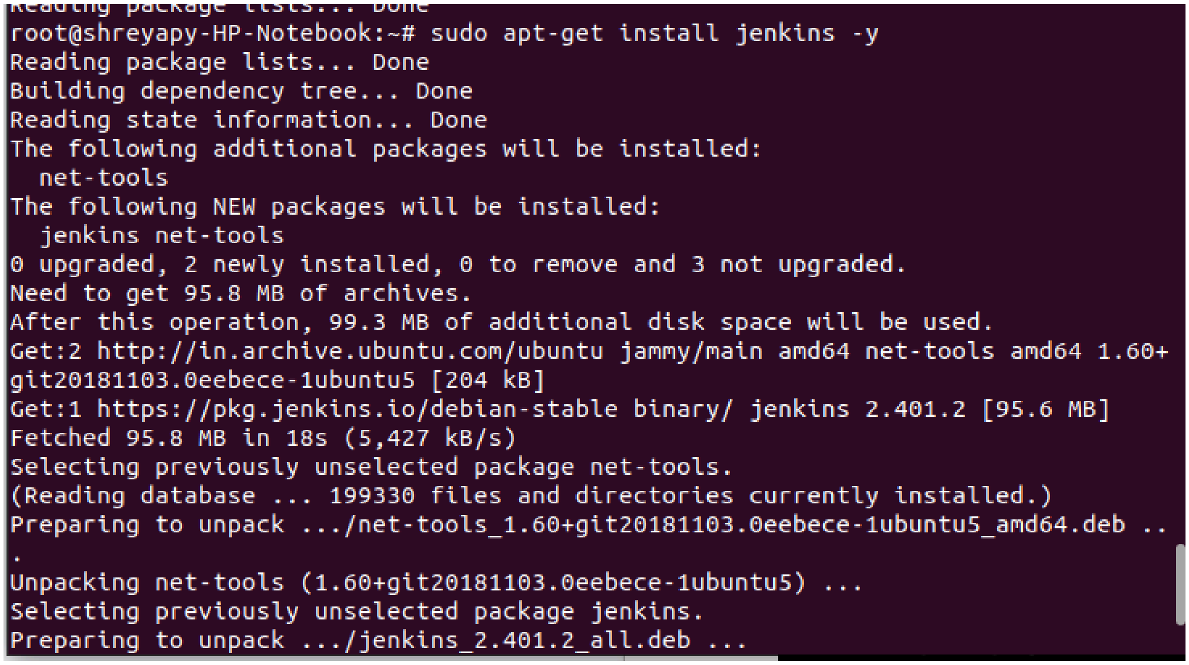
```
shreyapy@shreyapy-HP-Notebook:~$ wget -q -O - https://jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
shreyapy@shreyapy-HP-Notebook:~$ 
```

And run the command ***sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.list'*** *and after the command* ***sudo apt-get update***



```
shreyapy@shreyapy-HP-Notebook:~$ sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.
list'
shreyapy@shreyapy-HP-Notebook:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Ign:2 https://pkg.jenkins.io/debian binary/ InRelease
Get:6 https://pkg.jenkins.io/debian binary/ Release [2,044 B]
Get:7 https://pkg.jenkins.io/debian binary/ Release.gpg [833 B]
Ign:7 https://pkg.jenkins.io/debian binary/ Release.gpg
Reading package lists... Done
W: GPG error: https://pkg.jenkins.io/debian binary/ Release: The following signatures couldn't be verified because the public key
 is not available: NO_PUBKEY 5BA31D57EF5975CA
E: The repository 'http://pkg.jenkins-ci.org/debian binary/ Release' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
shreyapy@shreyapy-HP-Notebook:~$
```

Has installed all pre-requisites to install Jenkins, to install Jenkins execute the command ***sudo apt-get install jenkins***

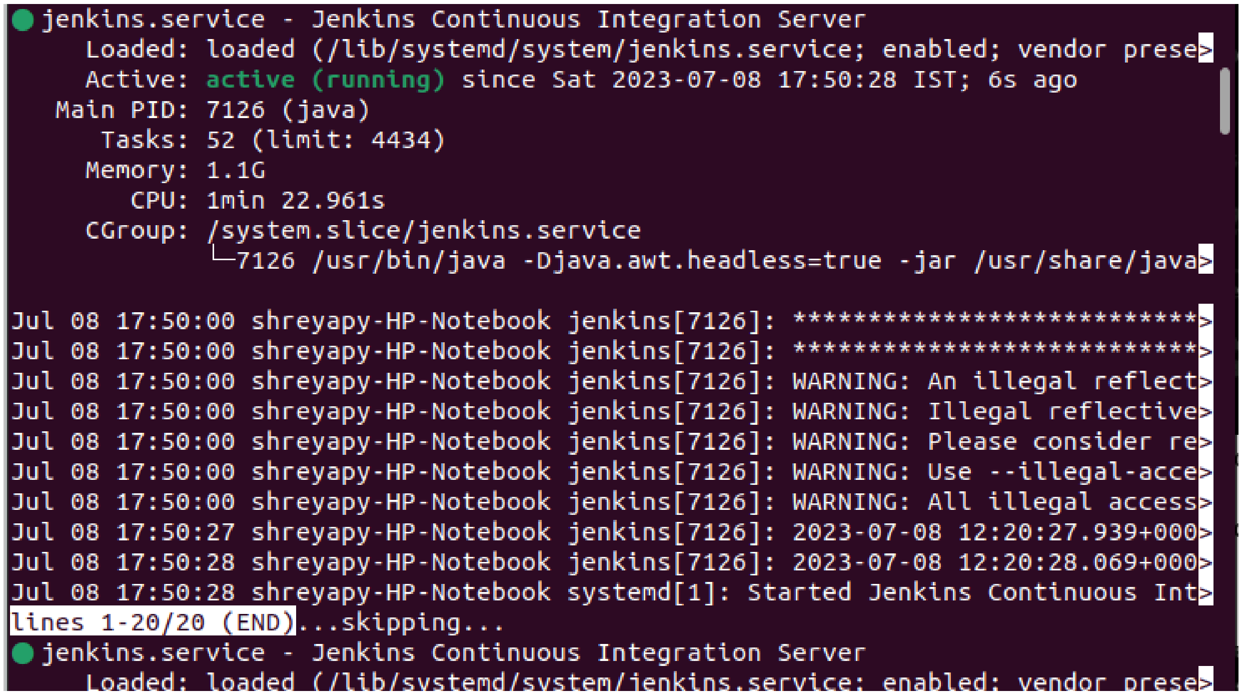


```
Reading package lists... Done
root@shreyapy-HP-Notebook:~# sudo apt-get install jenkins -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 95.8 MB of archives.
After this operation, 99.3 MB of additional disk space will be used.
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+
git20181103.0eebece-1ubuntu5 [204 kB]
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.401.2 [95.6 MB]
Fetched 95.8 MB in 18s (5,427 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 199330 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ..
.
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.401.2_all.deb ...
```

After Jenkins installation, is necessary to start the Jenkins, execute following command: ***sudo systemctl start jenkins***

To verify the start status of Jenkins, execute the following command

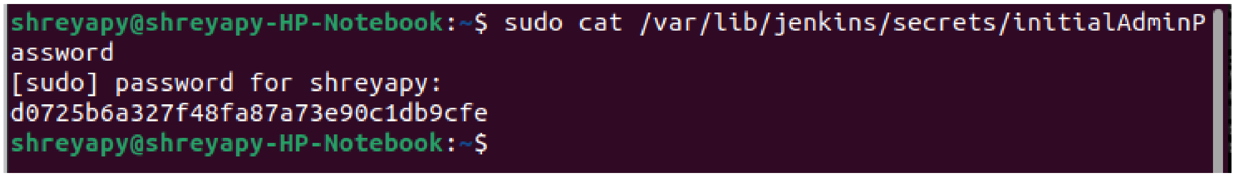
sudo systemctl status jenkins



```
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese>
     Active: active (running) since Sat 2023-07-08 17:50:28 IST; 6s ago
   Main PID: 7126 (java)
      Tasks: 52 (limit: 4434)
     Memory: 1.1G
        CPU: 1min 22.961s
     CGroup: /system.slice/jenkins.service
             └─7126 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java>

Jul 08 17:50:00 shreyapy-HP-Notebook jenkins[7126]: ***************************>
Jul 08 17:50:00 shreyapy-HP-Notebook jenkins[7126]: ***************************>
Jul 08 17:50:00 shreyapy-HP-Notebook jenkins[7126]: WARNING: An illegal reflect>
Jul 08 17:50:00 shreyapy-HP-Notebook jenkins[7126]: WARNING: Illegal reflective>
Jul 08 17:50:00 shreyapy-HP-Notebook jenkins[7126]: WARNING: Please consider re>
Jul 08 17:50:00 shreyapy-HP-Notebook jenkins[7126]: WARNING: Use --illegal-acce>
Jul 08 17:50:00 shreyapy-HP-Notebook jenkins[7126]: WARNING: All illegal access>
Jul 08 17:50:27 shreyapy-HP-Notebook jenkins[7126]: 2023-07-08 12:20:27.939+000>
Jul 08 17:50:28 shreyapy-HP-Notebook jenkins[7126]: 2023-07-08 12:20:28.069+000>
Jul 08 17:50:28 shreyapy-HP-Notebook systemd[1]: Started Jenkins Continuous Int>
lines 1-20/20 (END)...skipping...
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese>
```

When Jenkins has installed, Jenkins generated an initial admin password, because of that is necessary to copy this password after access Jenkins local website.

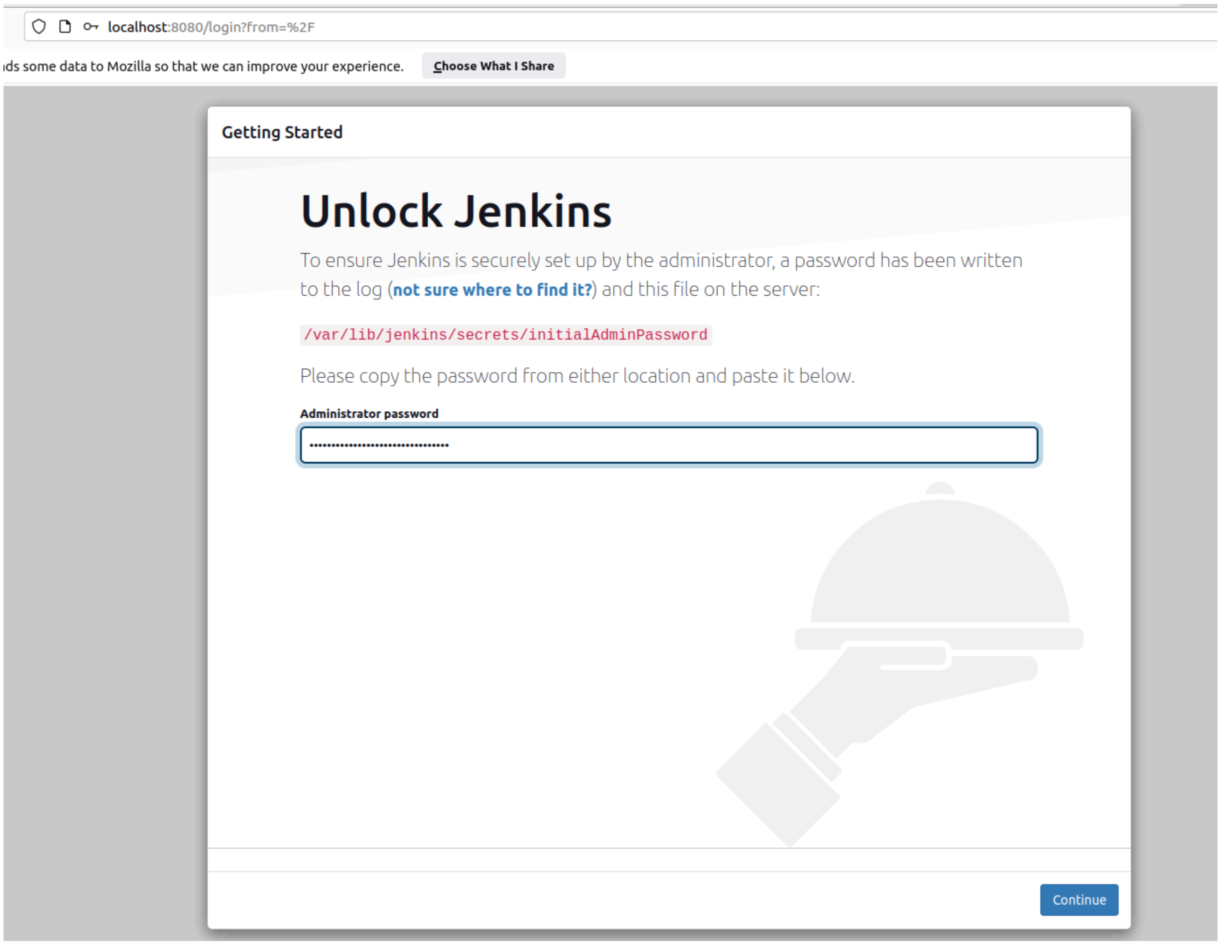
To verify Jenkins Initial admin password, is necessary to execute this command:
sudo cat /var/lib/jenkins/secrets/initialAdminPassword



```
shreyapy@shreyapy-HP-Notebook:~$ sudo cat /var/lib/jenkins/secrets/initialAdminP
assword
[sudo] password for shreyapy:
d0725b6a327f48fa87a73e90c1db9cfe
shreyapy@shreyapy-HP-Notebook:~$
```

And copy this initial password

Open a browser on a URL http://localhost:8080 and on field Administrator password, paste the copied value.


localhost:8080/login?from=%2F
ıds some data to Mozilla so that we can improve your experience.
Choose What I Share

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:

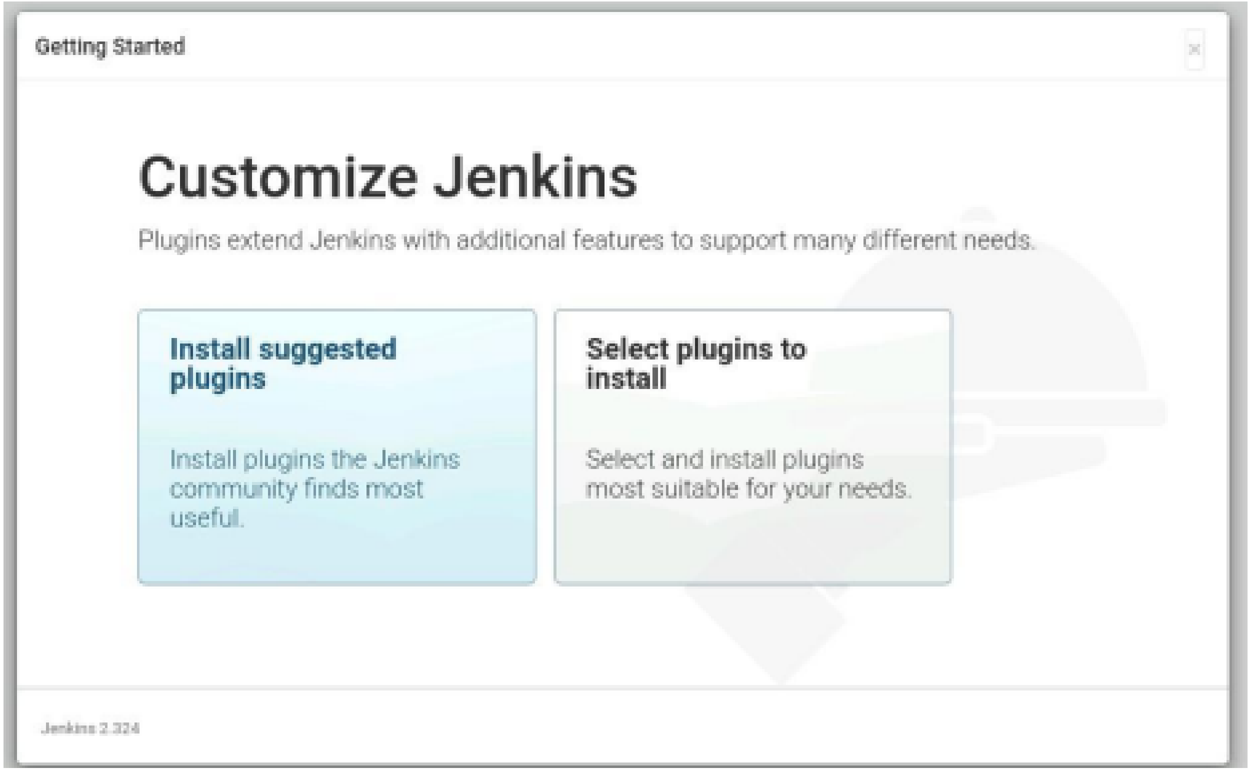
`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

••••••••••••••••••••••••••••••••

Click on Install suggested plugins



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

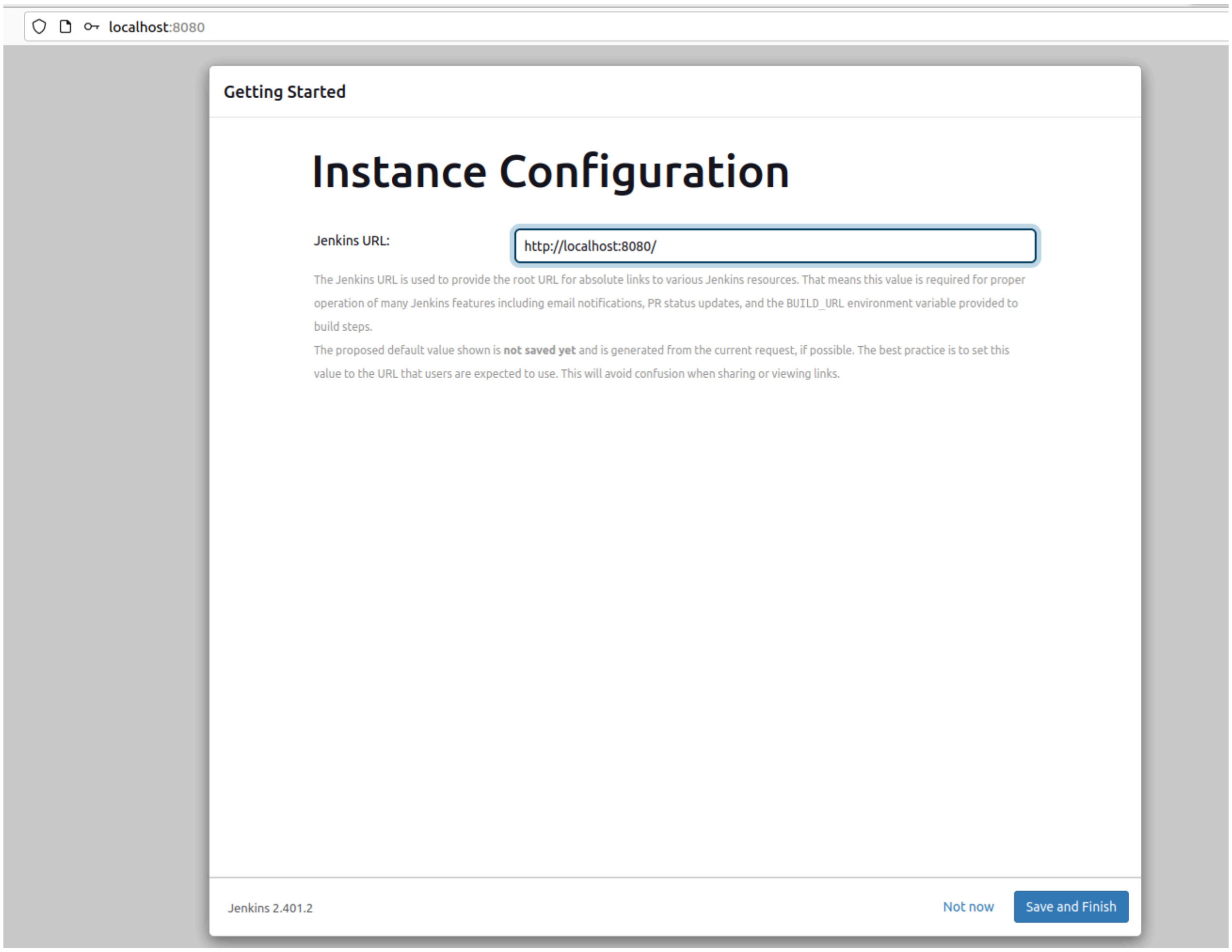
Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.324

To facility the currently process, we going to proceed with admin access

Is possible to change the Jenkins default URL, but at this moment we proceed with this url and port



localhost:8080

Getting Started

Instance Configuration

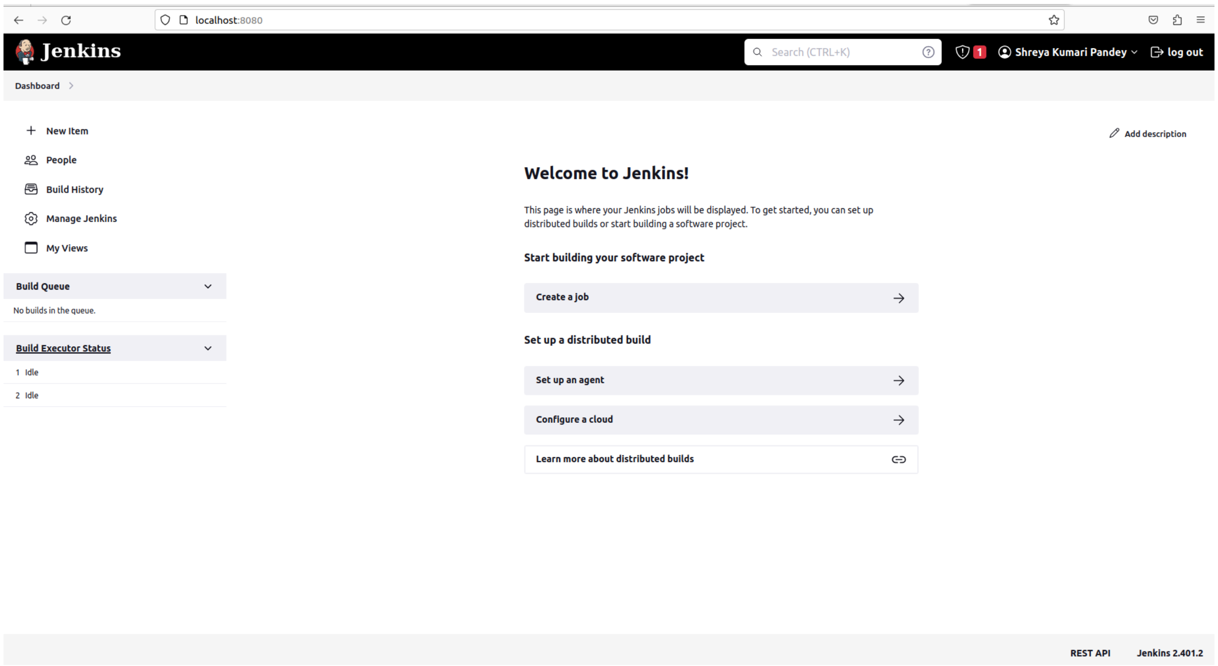
Jenkins URL: http://localhost:8080/

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 2.401.2

The default installation has executed



localhost:8080

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

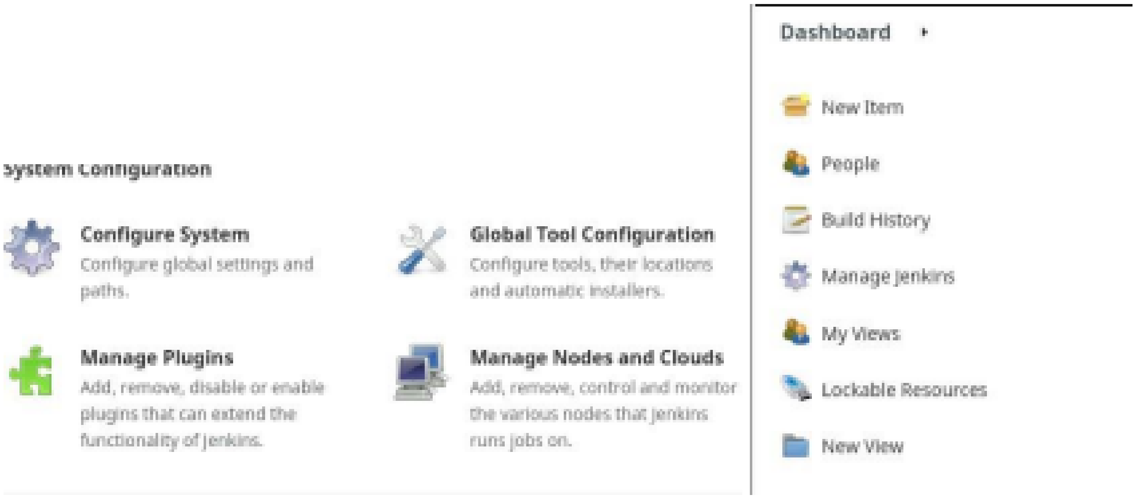
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

Jenkins 2.401.2

With Suggested plugins has installed the github plugin, but we also proceed with docker plugin, but is necessary to install docker plugin to perform our solution, because of this is necessary to click on Manage Jenkins > Manage Plugins



System Configuration

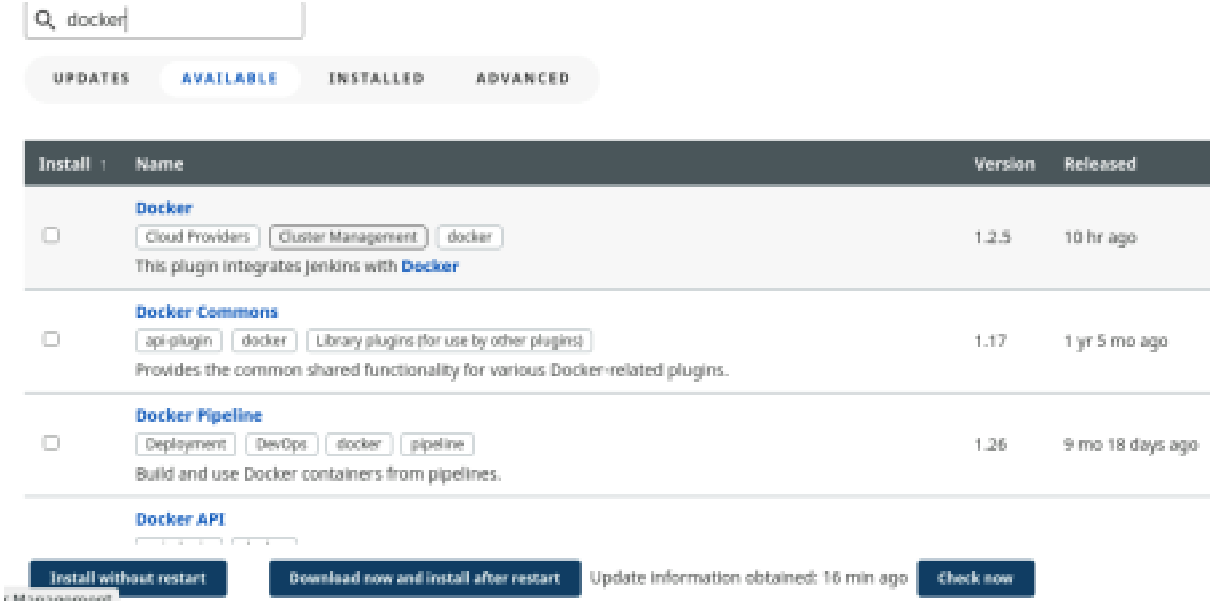
Configure System
Configure global settings and paths.

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

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

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Click on "***Available***" and Search for "***Docker***", select Docker and Download ***now and install after restart***



Install ↑	Name	Version	Released
□	Docker Cloud Providers · Cluster Management · docker This plugin integrates Jenkins with Docker	1.2.5	10 hr ago
□	Docker Commons api-plugin · docker · Library plugins (for use by other plugins) Provides the common shared functionality for various Docker-related plugins.	1.17	1 yr 5 mo ago
□	Docker Pipeline Deployment · DevOps · docker · pipeline Build and use Docker containers from pipelines.	1.26	9 mo 18 days ago
	Docker API		

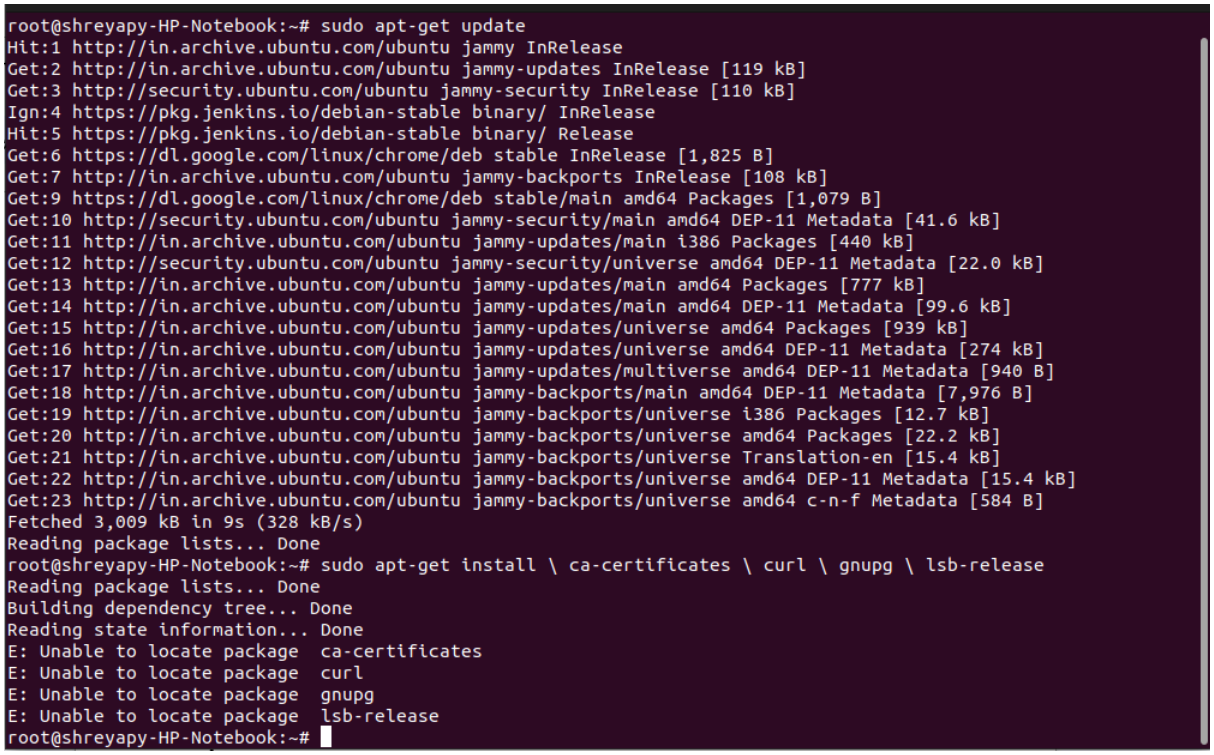
Update information obtained: 16 min ago

Docker Instalation

On terminal Update the apt package index and install packages to allow apt to use a repository over HTTPS:

sudo apt-get update

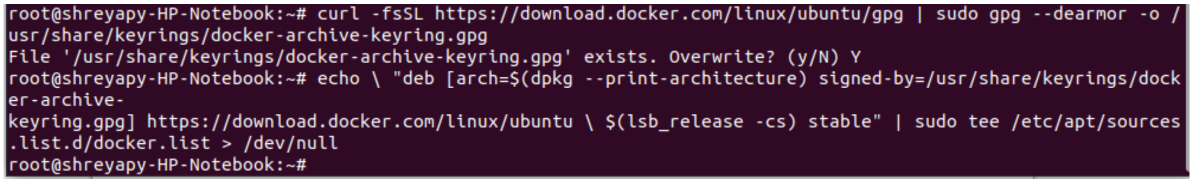
sudo apt-get install \ ca-certificates \ curl \ gnupg \ lsb-release



```
root@shreyapy-HP-Notebook:~# sudo apt-get update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian-stable binary/ Release
Get:6 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:9 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,079 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [41.6 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [440 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [22.0 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [777 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [99.6 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [939 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [274 kB]
Get:17 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:18 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [7,976 B]
Get:19 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [12.7 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [22.2 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [15.4 kB]
Get:22 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [15.4 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [584 B]
Fetched 3,009 kB in 9s (328 kB/s)
Reading package lists... Done
root@shreyapy-HP-Notebook:~# sudo apt-get install \ ca-certificates \ curl \ gnupg \ lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package  ca-certificates
E: Unable to locate package  curl
E: Unable to locate package  gnupg
E: Unable to locate package  lsb-release
root@shreyapy-HP-Notebook:~# 
```

Add Docker’s official GPG key

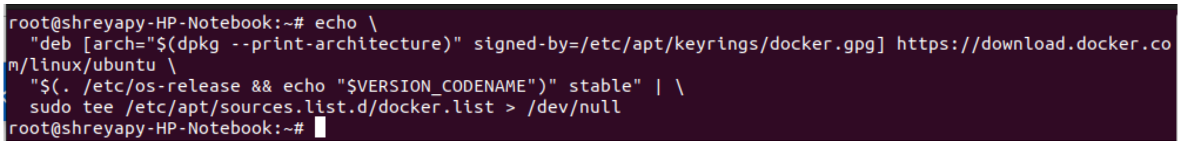
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg



```
root@shreyapy-HP-Notebook:~# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /
usr/share/keyrings/docker-archive-keyring.gpg
File '/usr/share/keyrings/docker-archive-keyring.gpg' exists. Overwrite? (y/N) Y
root@shreyapy-HP-Notebook:~# echo \ "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/dock
er-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) stable" | sudo tee /etc/apt/sources
.list.d/docker.list > /dev/null
root@shreyapy-HP-Notebook:~#
```

Use the following command to set up the stable repository. To add the nightly or test repository, add the word nightly or test (or both) after the word stable in the commands below. Learn about nightly and test channels.

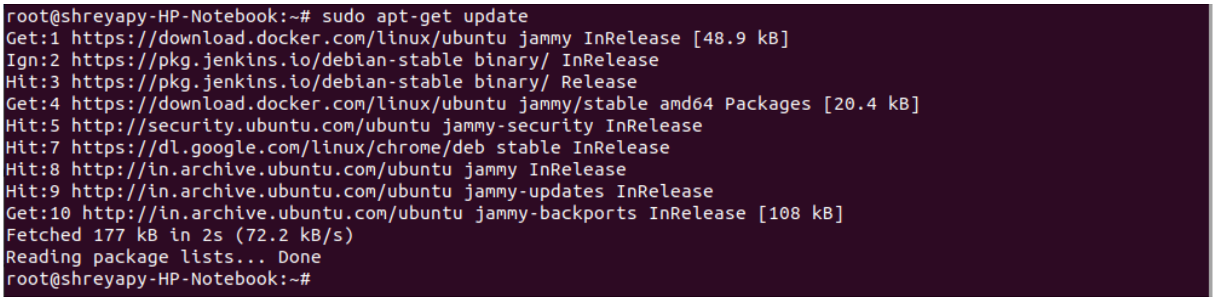
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive keyring.gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null



```
root@shreyapy-HP-Notebook:~# echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.co
m/linux/ubuntu \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
root@shreyapy-HP-Notebook:~# 
```

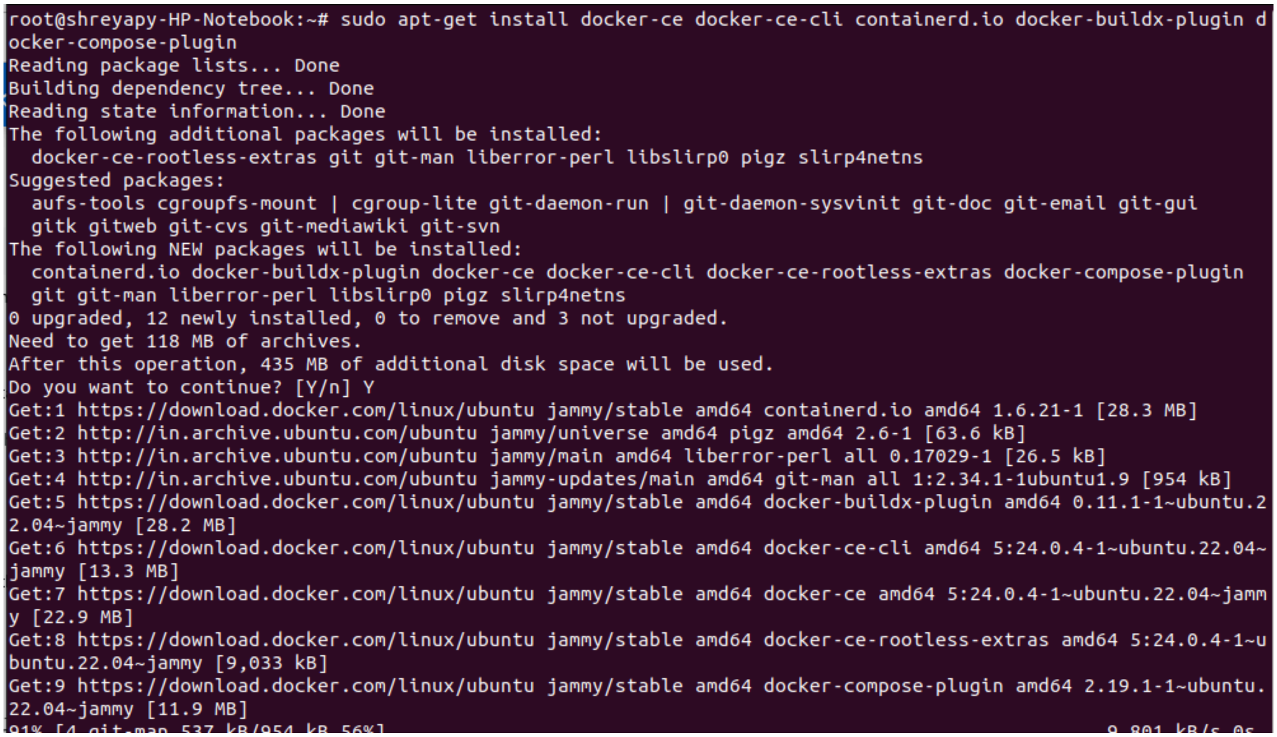
Update the apt package index, and install the latest version of Docker Engine and containerd, or go to the next step to install a specific version:

sudo apt-get update

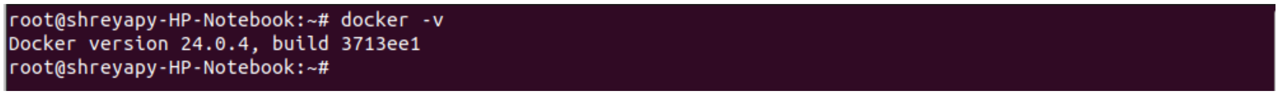


```
root@shreyapy-HP-Notebook:~# sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.9 kB]
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:3 https://pkg.jenkins.io/debian-stable binary/ Release
Get:4 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [20.4 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:7 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:8 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:9 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Fetched 177 kB in 2s (72.2 kB/s)
Reading package lists... Done
root@shreyapy-HP-Notebook:~#
```

Use the following commands to install the latest version of Docker CE and check the version: ***sudo apt-get install docker-ce***
docker --version

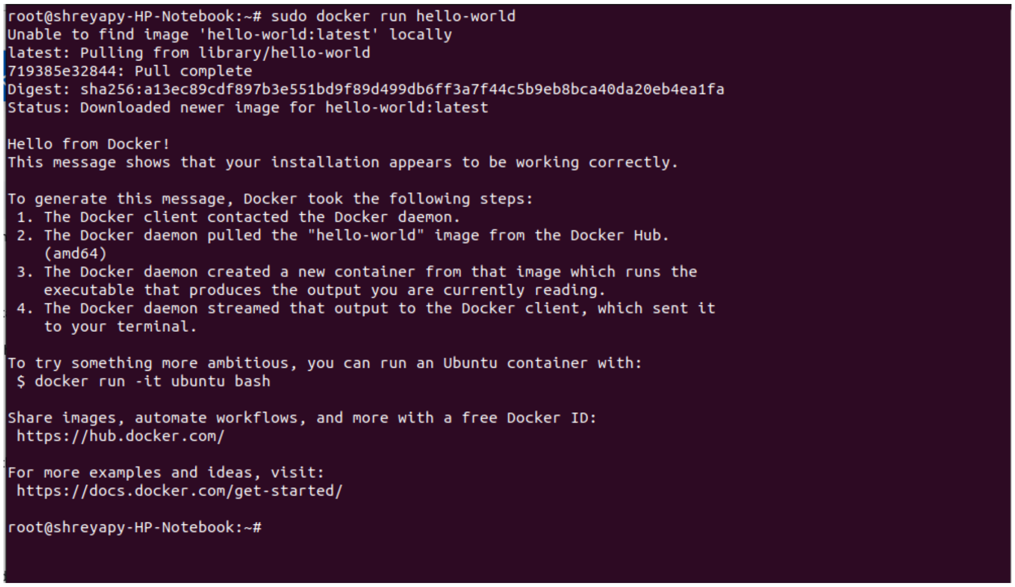


```
root@shreyapy-HP-Notebook:~# sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin d
ocker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui
  gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  git git-man liberror-perl libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 3 not upgraded.
Need to get 118 MB of archives.
After this operation, 435 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.21-1 [28.3 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.9 [954 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.11.1-1~ubuntu.2
2.04~jammy [28.2 MB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:24.0.4-1~ubuntu.22.04~
jammy [13.3 MB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:24.0.4-1~ubuntu.22.04~jamm
y [22.9 MB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:24.0.4-1~u
buntu.22.04~jammy [9,033 kB]
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.19.1-1~ubuntu.
22.04~jammy [11.9 MB]
91% [4 git-man 537 kB/954 kB 56%]                                                        9,801 kB/s 0s
```



```
root@shreyapy-HP-Notebook:~# docker -v
Docker version 24.0.4, build 3713ee1
root@shreyapy-HP-Notebook:~#
```

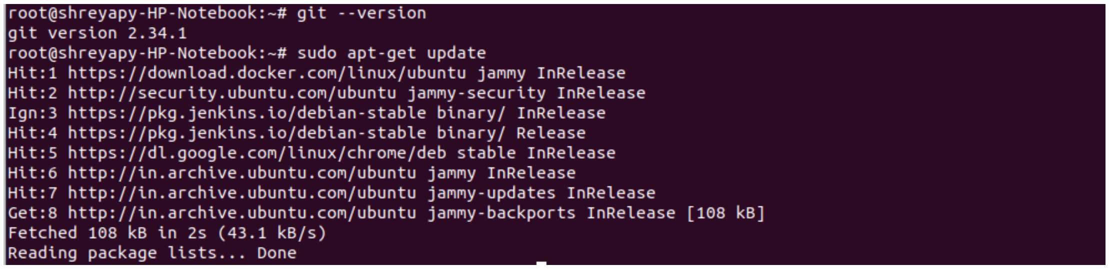
Test if docker is working running the default test repository from docker (hello-world)



```
root@shreyapy-HP-Notebook:~# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:a13ec89cdf897b3e551bd9f89d499db6ff3a7f44c5b9eb8bca40da20eb4ea1fa
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

root@shreyapy-HP-Notebook:~#
```

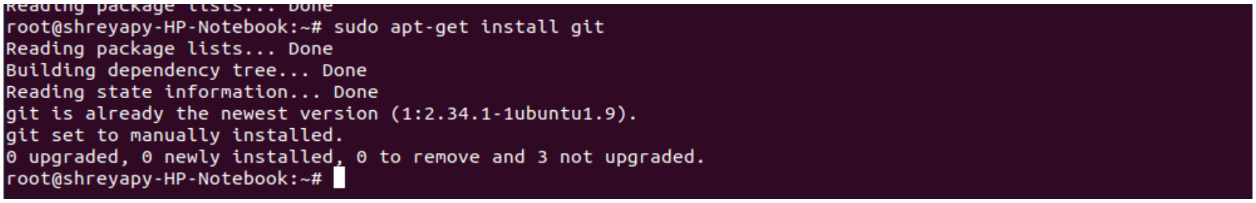
Git Installation

First is necessary to check if the git is installed, if it is not installed, doesn't have a problem, these following steps are also responsible for installing the git.
Case is installed, is only necessary execute these following commands:
sudo apt-get update



```
root@shreyapy-HP-Notebook:~# git --version
git version 2.34.1
root@shreyapy-HP-Notebook:~# sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:3 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:4 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:5 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:7 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Fetched 108 kB in 2s (43.1 kB/s)
Reading package lists... Done
```

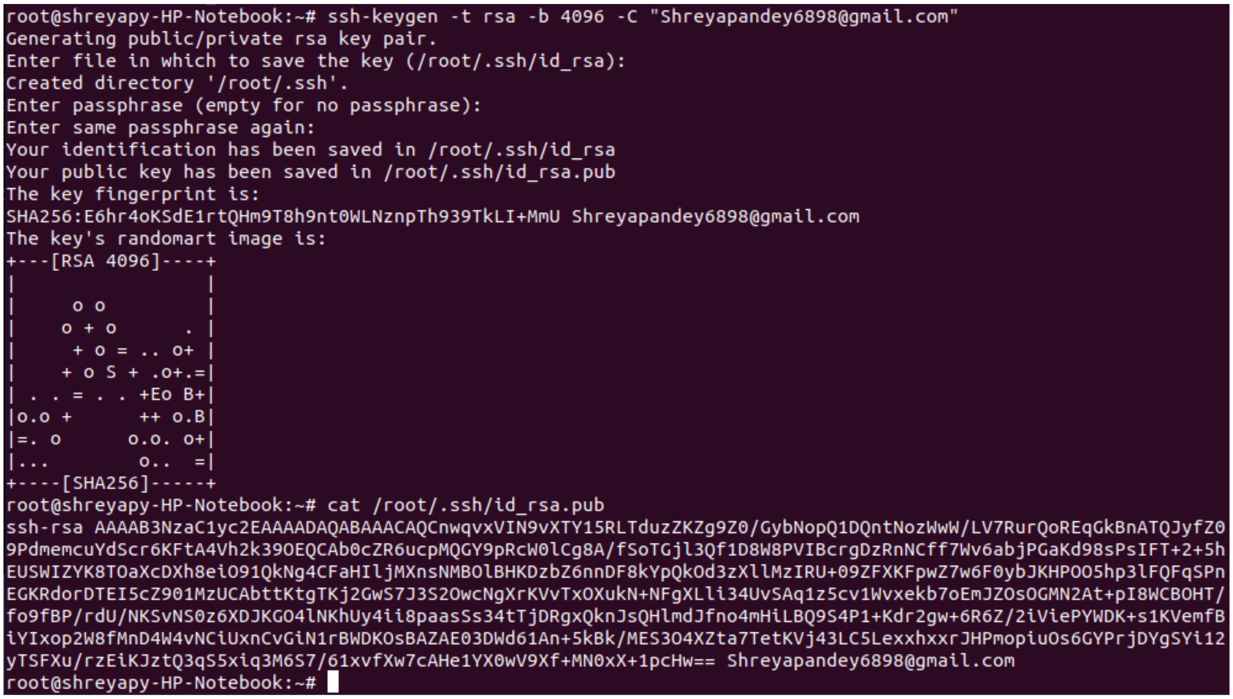
sudo apt-get install git



```
Reading package lists... Done
root@shreyapy-HP-Notebook:~# sudo apt-get install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.9).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
root@shreyapy-HP-Notebook:~# 
```

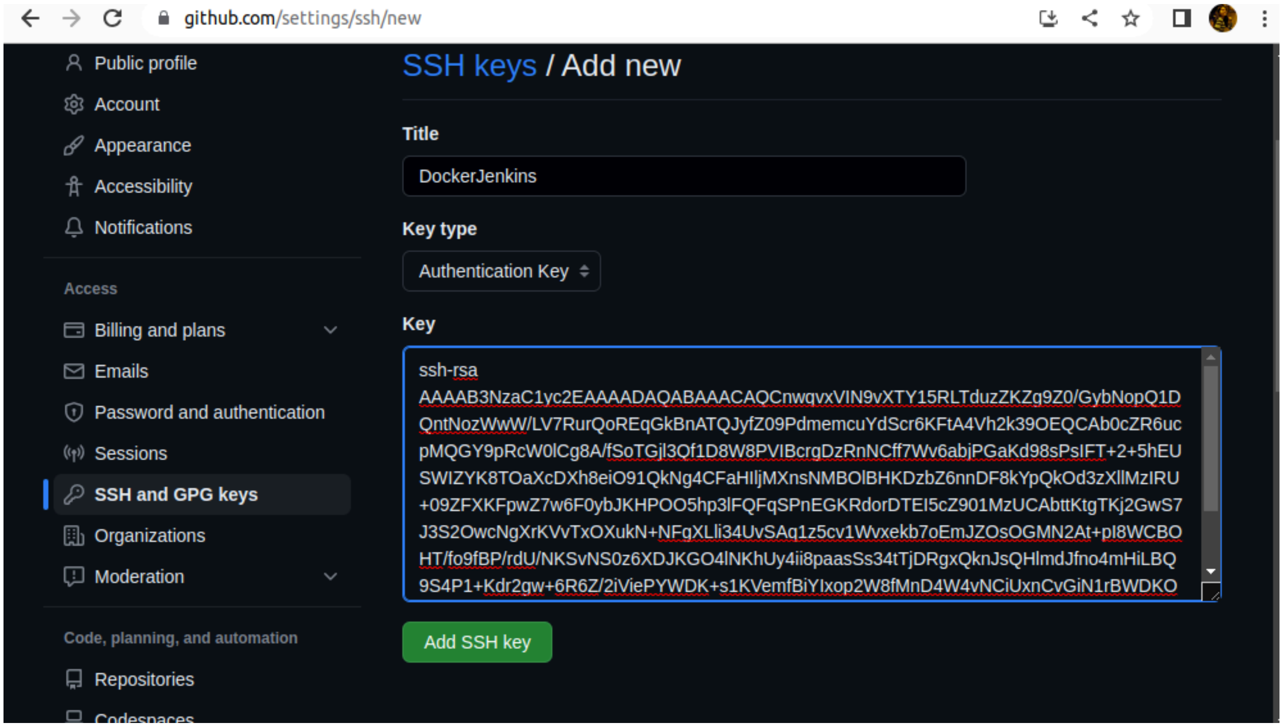
The last code (git –version) is to confer that git has been installed

On terminal, generate a new SSH key, use command ***ssh-keygen -t rsa -b 4096 -C "felandim@gmail.com"***. Enter to accept default filename and input passphase as needed.
Copy the generated ssh with command ***cat /home/felandimgmail/.ssh/id_rsa.pub***



```
root@shreyapy-HP-Notebook:~# ssh-keygen -t rsa -b 4096 -C "Shreyapandey6898@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:E6hr4oKSdE1rtQHm9T8h9nt0WLNznpTh939TkLI+MmU Shreyapandey6898@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|                 |
|     o o         |
|    o + o      . |
|     + o = .. o+ |
|    + o S + .o+.=|
| . . = . . +Eo B+|
|o.o +      ++ o.B|
|=. o      o.o. o+|
|...        o..  =|
+----[SHA256]-----+
root@shreyapy-HP-Notebook:~# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCnwqvxVIN9vXTY15RLTduzZKZg9Z0/GybNopQ1DQntNozWwW/LV7RurQoREqGkBnATQJyfZ0
9PdmemcuYdScr6KFtA4Vh2k39OEQCAb0cZR6ucpMQGY9pRcW0lCg8A/fSoTGjl3Qf1D8W8PVIBcrgDzRnNCff7Wv6abjPGaKd98sPsIFT+2+5h
EUSWIZYK8TOaXcDXh8eiO91QkNg4CFaHIljMXnsNMBOlBHKDzbZ6nnDF8kYpQkOd3zXllMzIRU+09ZFXKFpwZ7w6F0ybJKHPOO5hp3lFQFqSPn
EGKRdorDTEI5cZ901MzUCAbttKtgTKj2GwS7J3S2OwcNgXrKVvTxOXukN+NFgXLli34UvSAq1z5cv1Wvxekb7oEmJZOsOGMN2At+pI8WCBOHT/
fo9fBP/rdU/NKSvNS0z6XDJKGO4lNKhUy4ii8paasSs34tTjDRgxQknJsQHlmdJfno4mHiLBQ9S4P1+Kdr2gw+6R6Z/2iViePYWDK+s1KVemfB
iYIxop2W8fMnD4W4vNCiUxnCvGiN1rBWDKOsBAZAE03DWd61An+5kBk/MES3O4XZta7TetKVj43LC5LexxhxxrJHPmopiuOs6GYPrjDYgSYi12
yTSFXu/rzEiKJztQ3qS5xiq3M6S7/61xvfXw7cAHe1YX0wV9Xf+MN0xX+1pcHw== Shreyapandey6898@gmail.com
root@shreyapy-HP-Notebook:~# 
```

On GitHub portal access Profile >> SSH Keys >> Add New and paste the copied value



SSH keys / Add new

Title

DockerJenkins

Key type

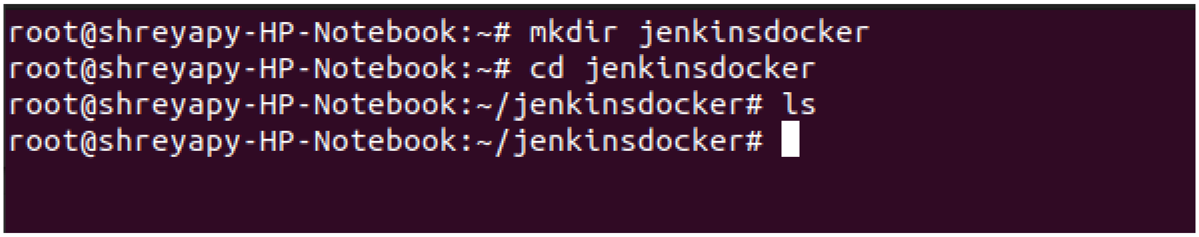
Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQCnwqvxVIN9vXTY15RLTduzZKZg9Z0/GybNopQ1D
QntNozWwW/LV7RurQoREqGkBnATQJyfZ09PdmemcuYdScr6KFtA4Vh2k39OEQCAb0cZR6uc
pMQGY9pRcW0lCg8A/fSoTGjl3Qf1D8W8PVIBcrgDzRnNCff7Wv6abjPGaKd98sPsIFT+2+5hEU
SWIZYK8TOaXcDXh8eiO91QkNg4CFaHIljMXnsNMBOlBHKDzbZ6nnDF8kYpQkOd3zXllMzIRU
+09ZFXKFpwZ7w6F0ybJKHPOO5hp3lFQFqSPnEGKRdorDTEI5cZ901MzUCAbttKtgTKj2GwS7
J3S2OwcNgXrKVvTxOXukN+NFgXLli34UvSAq1z5cv1Wvxekb7oEmJZOsOGMN2At+pl8WCBO
HT/fo9fBP/rdU/NKSvNS0z6XDJKGO4lNKhUy4ii8paasSs34tTjDRgxQknJsQHlmdJfno4mHiLBQ
9S4P1+Kdr2gw+6R6Z/2iViePYWDK+s1KVemfBiYIxop2W8fMnD4W4vNCiUxnCvGiN1rBWDKO
```

GitHub Repository

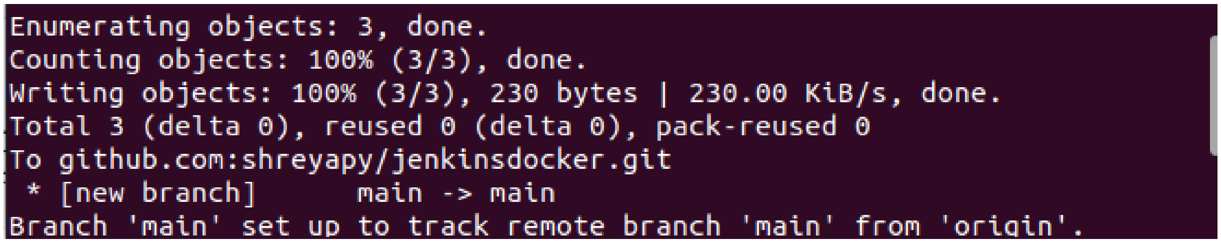
Create a new Jenkins repository on GitHub WebSite



```
root@shreyapy-HP-Notebook:~# mkdir jenkinsdocker
root@shreyapy-HP-Notebook:~# cd jenkinsdocker
root@shreyapy-HP-Notebook:~/jenkinsdocker# ls
root@shreyapy-HP-Notebook:~/jenkinsdocker#
```

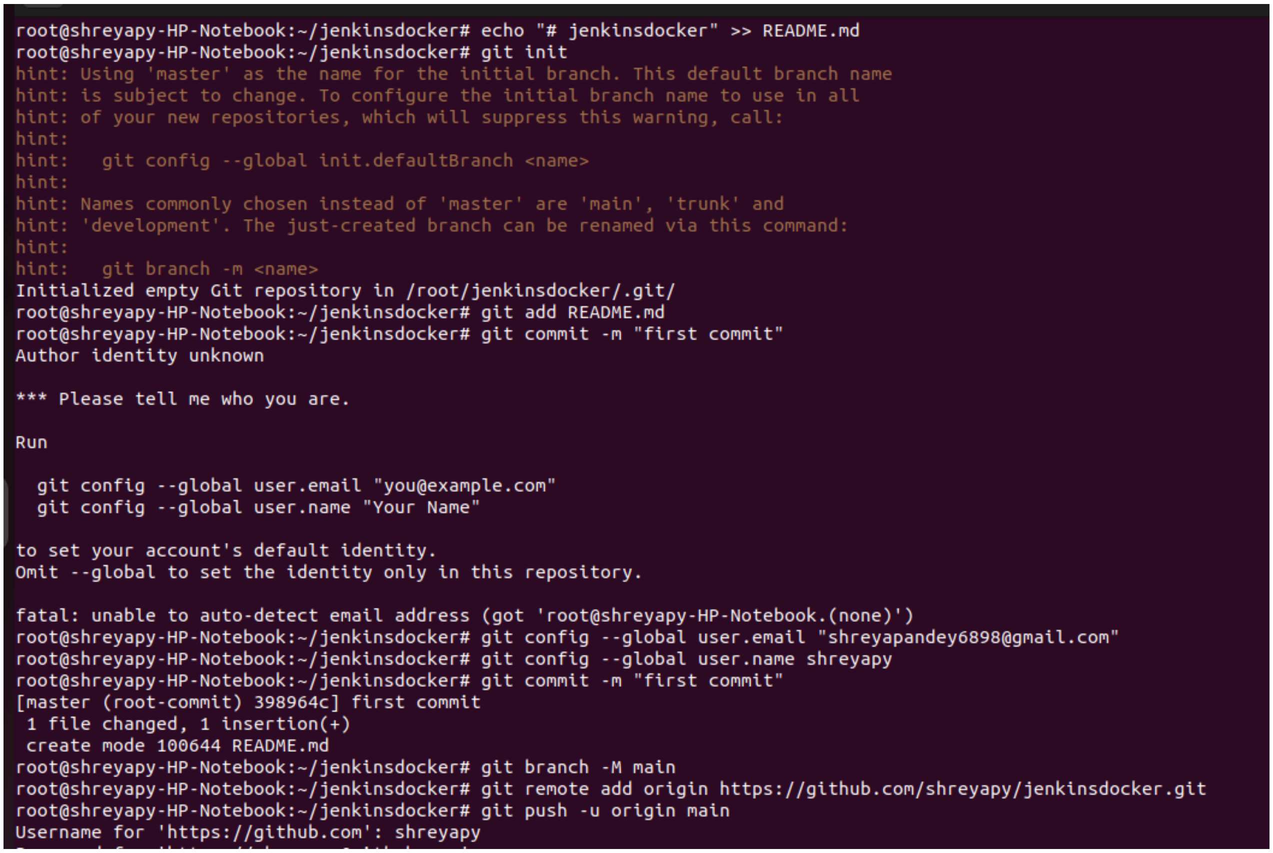
Create a directory named Jenkinsdocker
Execute these following commands

```
echo "# jenkinsdocker" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin
git@github.com:lipelandim/jenkinsdocker.git git push -u origin
main
```

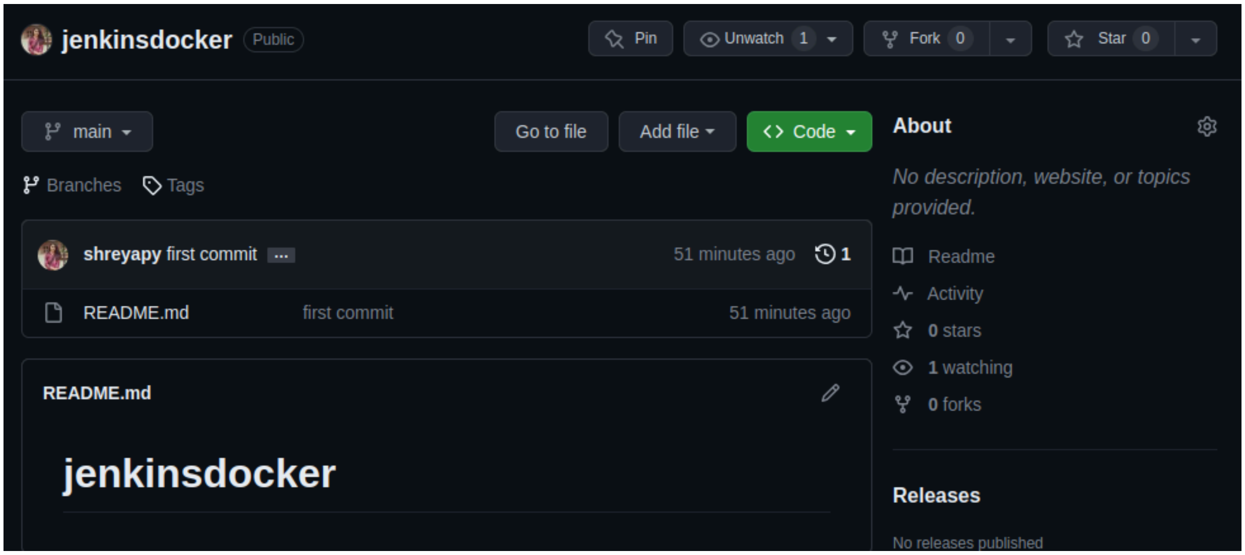


```
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 230 bytes | 230.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:shreyapy/jenkinsdocker.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

The github repo is working



```
root@shreyapy-HP-Notebook:~/jenkinsdocker# echo "# jenkinsdocker" >> README.md
root@shreyapy-HP-Notebook:~/jenkinsdocker# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/jenkinsdocker/.git/
root@shreyapy-HP-Notebook:~/jenkinsdocker# git add README.md
root@shreyapy-HP-Notebook:~/jenkinsdocker# git commit -m "first commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'root@shreyapy-HP-Notebook.(none)')
root@shreyapy-HP-Notebook:~/jenkinsdocker# git config --global user.email "shreyapandey6898@gmail.com"
root@shreyapy-HP-Notebook:~/jenkinsdocker# git config --global user.name shreyapy
root@shreyapy-HP-Notebook:~/jenkinsdocker# git commit -m "first commit"
[master (root-commit) 398964c] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
root@shreyapy-HP-Notebook:~/jenkinsdocker# git branch -M main
root@shreyapy-HP-Notebook:~/jenkinsdocker# git remote add origin https://github.com/shreyapy/jenkinsdocker.git
root@shreyapy-HP-Notebook:~/jenkinsdocker# git push -u origin main
Username for 'https://github.com': shreyapy
```



No description, website, or topics provided.

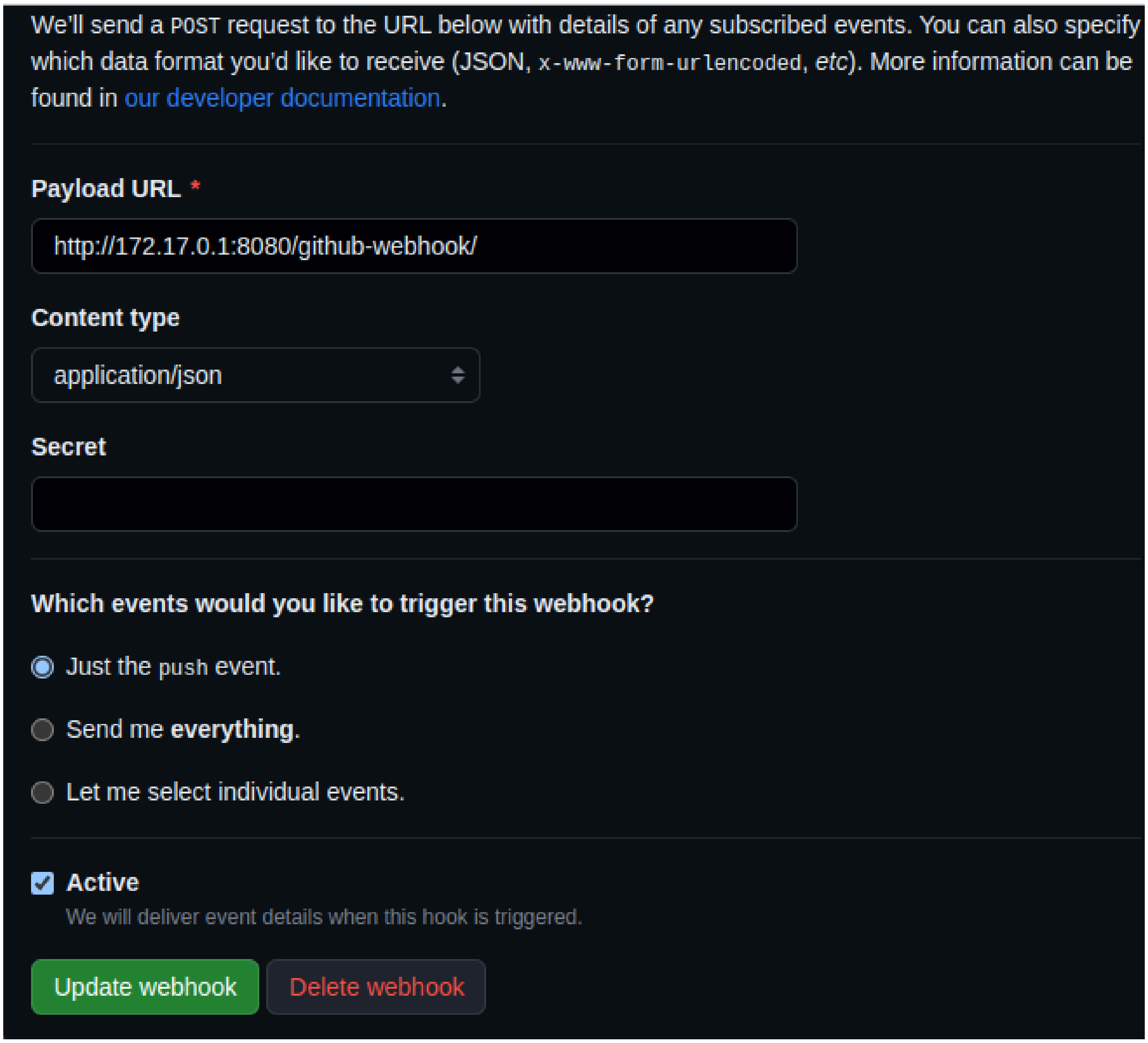
GitHub webhook

To configure the integration with GitHub.

Create a webhook inside the fork you created:

- Select **Settings**, then select **Webhooks** on the left-hand side.
- Choose **Add webhook**, then enter *Jenkins* in filter box.
- For the **Payload URL**, enter `http://<publicIps>:8080/github-webhook/`. Make sure you include the trailing /

- For **Content type**, select *application/x-www-form-urlencoded*.
- For **Which events would you like to trigger this webhook?**, select *Just the push event.*
- Set **Active** to checked.
- Click **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://172.17.0.1:8080/github-webhook/

Content type

application/json

Secret

Which events would you like to trigger this webhook?

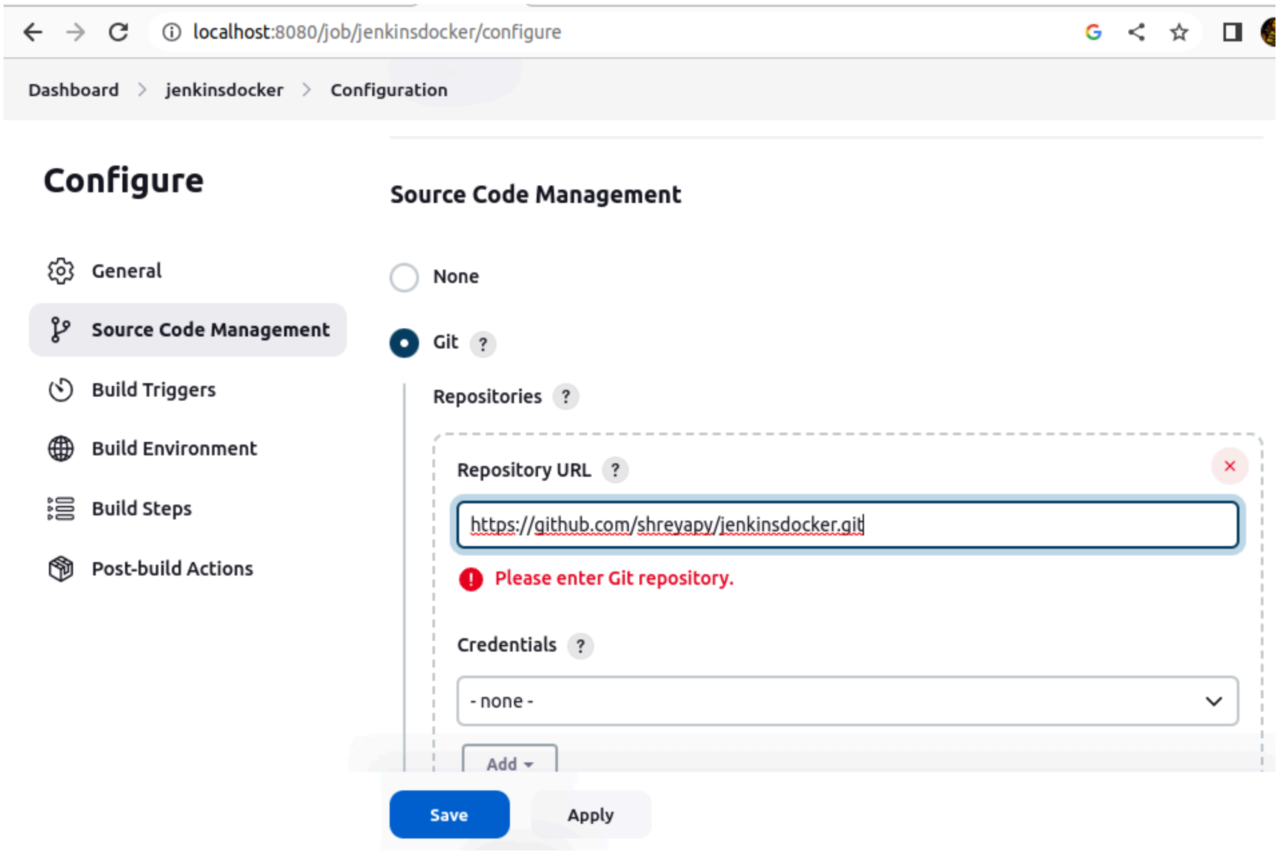
● Just the `push` event.

○ Send me **everything**.

○ Let me select individual events.

☑ **Active**

We will deliver event details when this hook is triggered.



Source Code Management

○ None

● Git

Repositories

Repository URL: https://github.com/shreyapy/jenkinsdocker.git

Please enter Git repository.

Credentials: - none -

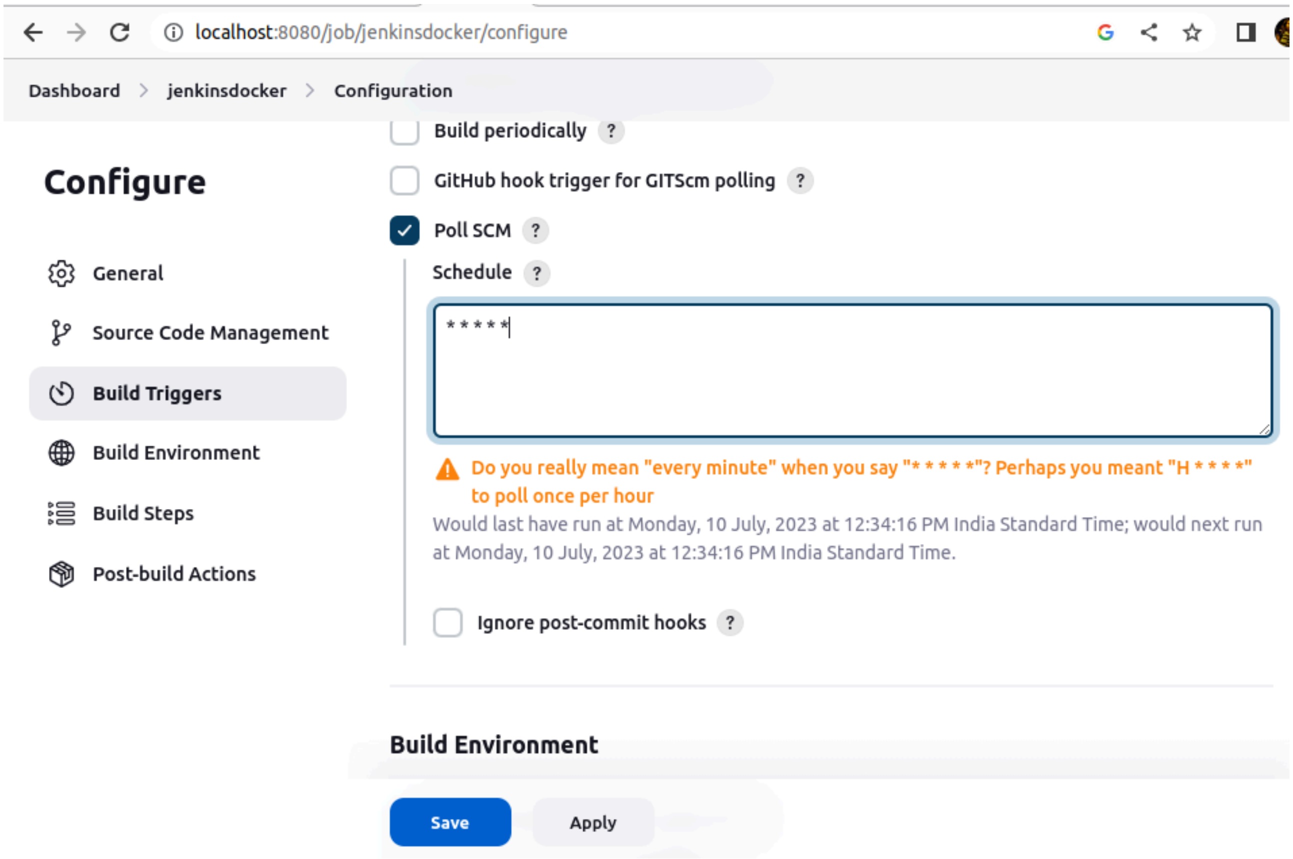
Integrating Jenkins and GitHub

To have Jenkins respond to an event in GitHub such as committing code, create a Jenkins job.

On Jenkins portal click on New Item

Select FreeStyle Project and put a name for your project >> Ok

On Source Code Management put the Repository URL On Build Trigger select Pool SCM and type * * * * *



localhost:8080/job/jenkinsdocker/configure

Dashboard > jenkinsdocker > Configuration

□ Build periodically ?

□ GitHub hook trigger for GITScm polling ?

☑ Poll SCM ?

Schedule ?

```
* * * * *
```

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Monday, 10 July, 2023 at 12:34:16 PM India Standard Time; would next run at Monday, 10 July, 2023 at 12:34:16 PM India Standard Time.

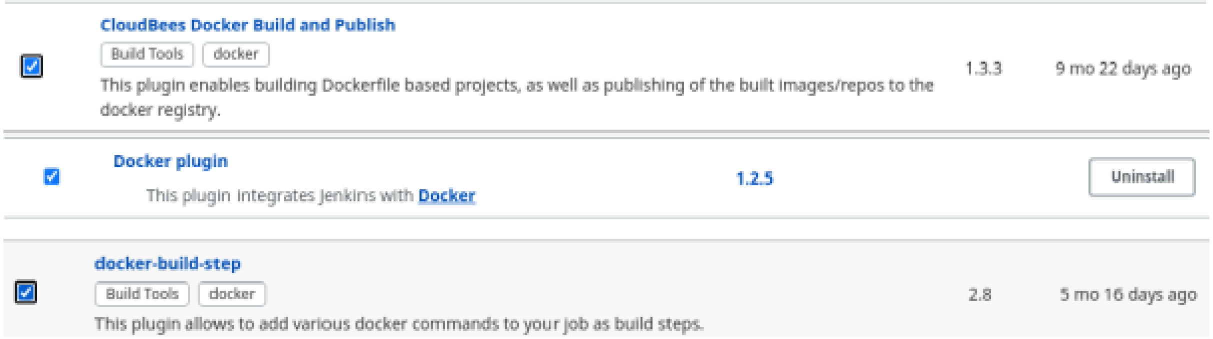
□ Ignore post-commit hooks ?

Build Environment

And Save changes.
If you change your code on github the process will starts automatically on Jenkins, the CI process is working fine.

Now is necessary to install Docker Plugin

Click on Manage Jenkins
Click on Manage Plugins
Install these plugins

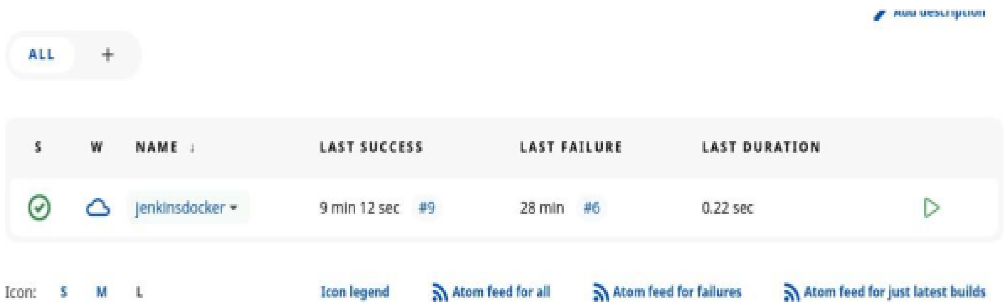


☑ CloudBees Docker Build and Publish — Build Tools, docker — 1.3.3 — 9 mo 22 days ago
This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker registry.

☑ Docker plugin — 1.2.5 — Uninstall
This plugin integrates Jenkins with Docker

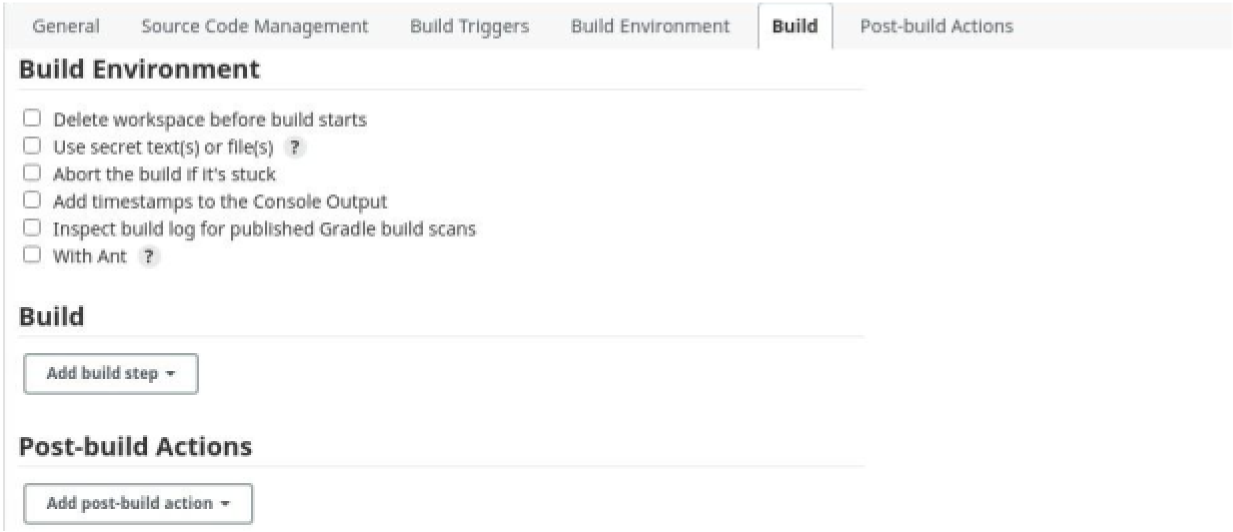
☑ docker-build-step — Build Tools, docker — 2.8 — 5 mo 16 days ago
This plugin allows to add various docker commands to your job as build steps.

Click on Project



S	W	NAME ↓	LAST SUCCESS	LAST FAILURE	LAST DURATION	
✓	☁	jenkinsdocker	9 min 12 sec #9	28 min #6	0.22 sec	▷

Click on Configure > Build



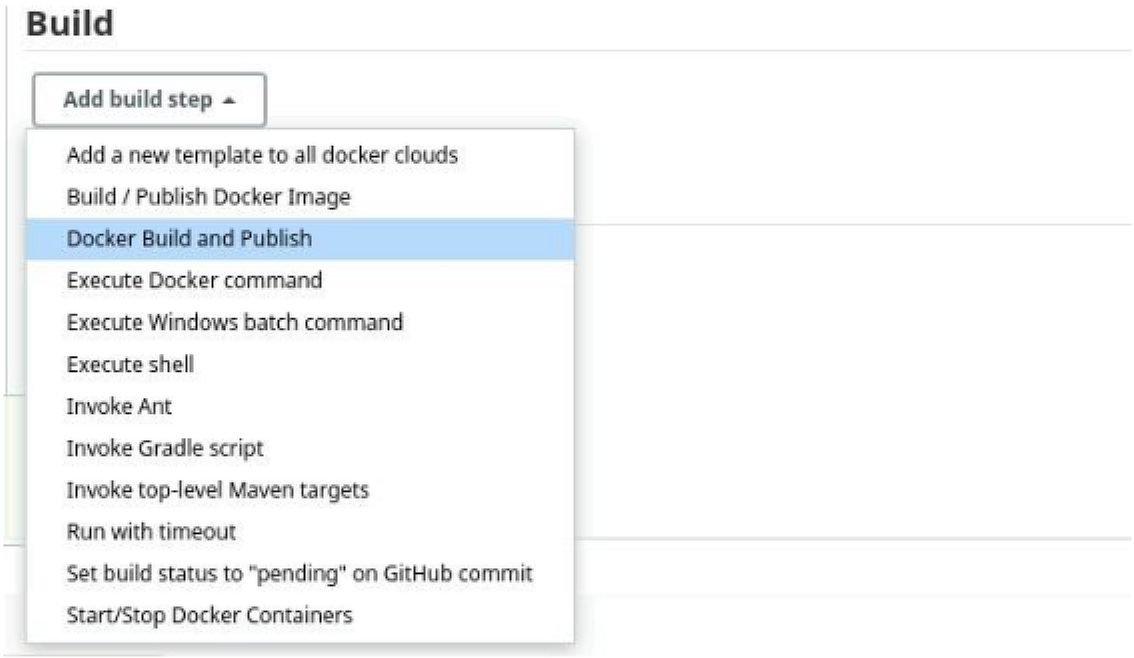
Build Environment

- ☐ 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 ?

Build

Post-build Actions

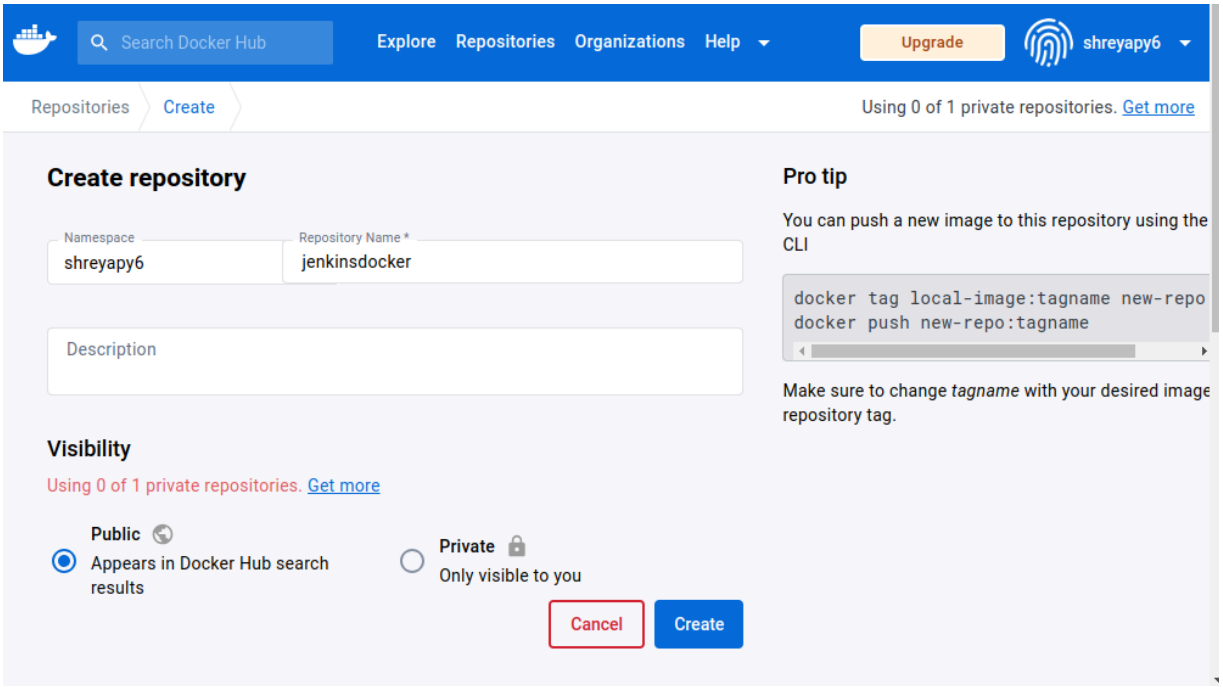
And on build button is possible to find a Docker Build and Publish


Build
Add build step ▴
Add a new template to all docker clouds
Build / Publish Docker Image
Docker Build and Publish
Execute Docker command
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit
Start/Stop Docker Containers

Add the credentials

Docker Image

Create docker repository on docker hub



Using 0 of 1 private repositories. Get more

Create repository

Namespace: shreyapy6

Repository Name *: jenkinsdocker

Description

Visibility

Using 0 of 1 private repositories. Get more

● Public – Appears in Docker Hub search results

○ Private – Only visible to you

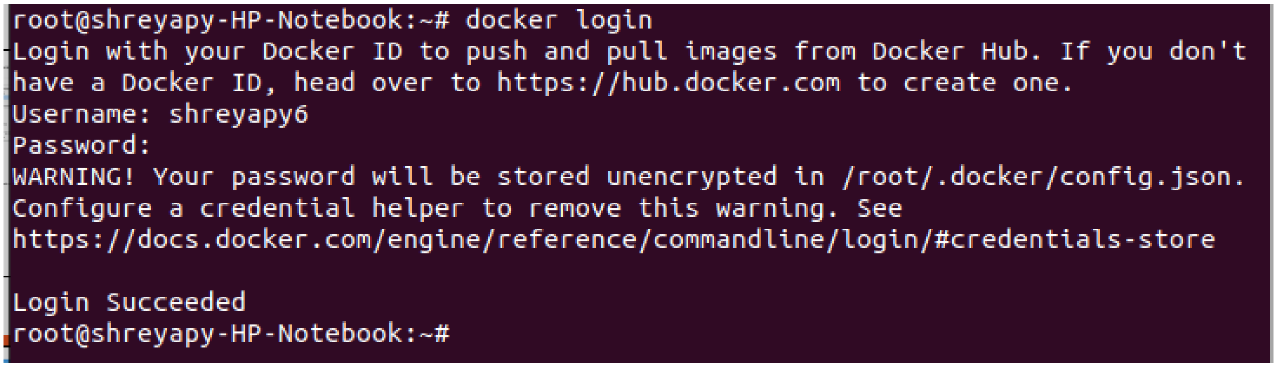
Pro tip

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

Create docker login



```
root@shreyapy-HP-Notebook:~# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username: shreyapy6
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@shreyapy-HP-Notebook:~#
```

Execute the creation of image from file dockerjenkins.