CLOUD TRAIN

ACCELERATE YOUR GROWTH

JENKINS RUNBOOK

DevOps Workshop

Contact us

TO ACCELERATE YOUR CAREER GROWTH

For questions and more details:

please call @ +91 98712 72900, or

visit https://www.thecloudtrain.com/, or

email at join@thecloudtrain.com, or

WhatsApp us @ +91 98712 72900

Exercise 1: Install Jenkins of GCP server and make sure its enabled and running once installed successfully.

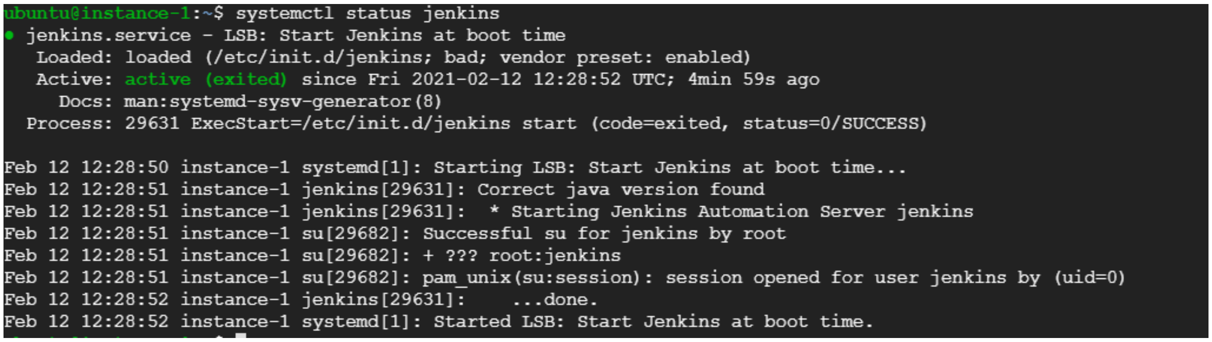
Solution:

1. Create one GCP Ubuntu 18.04 instance and run below commands one by one. Make sure each command executed successfully before running next command:

```
sudo apt update
sudo apt install openjdk-11-jdk
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo a
pt-key add -

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/
apt/sources.list.d/jenkins.list'

sudo apt update
sudo apt install jenkins
systemctl status jenkins
```



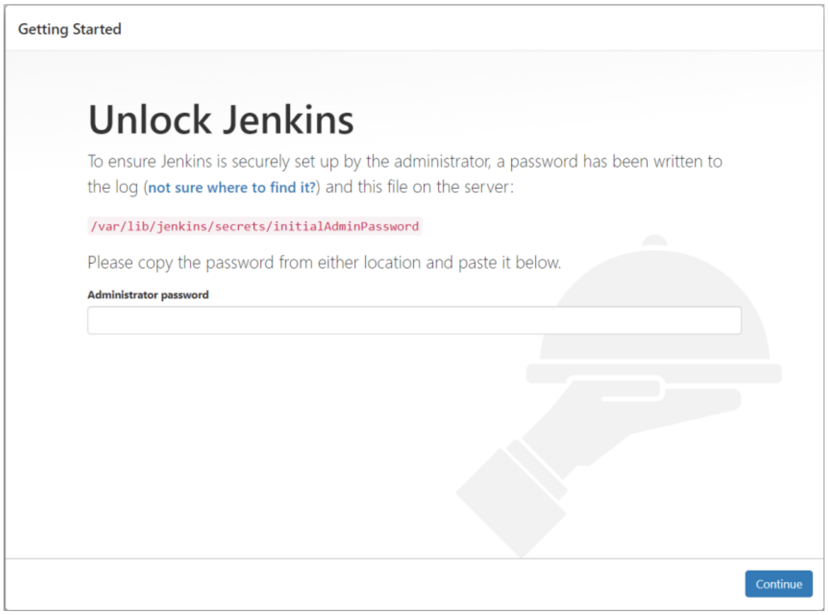
```
ubuntu@instance-1:~$ systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)
   Active: active (exited) since Fri 2021-02-12 12:28:52 UTC; 4min 59s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 29631 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)

Feb 12 12:28:50 instance-1 systemd[1]: Starting LSB: Start Jenkins at boot time...
Feb 12 12:28:51 instance-1 jenkins[29631]: Correct java version found
Feb 12 12:28:51 instance-1 jenkins[29631]:  * Starting Jenkins Automation Server jenkins
Feb 12 12:28:51 instance-1 su[29682]: Successful su for jenkins by root
Feb 12 12:28:51 instance-1 su[29682]: + ??? root:jenkins
Feb 12 12:28:51 instance-1 su[29682]: pam_unix(su:session): session opened for user jenkins by (uid=0)
Feb 12 12:28:52 instance-1 jenkins[29631]:    ...done.
Feb 12 12:28:52 instance-1 systemd[1]: Started LSB: Start Jenkins at boot time.
```

Note: In case you face any issue, refer to the tutorial recording

Exercise 2: Complete below tasks as part of this exercise:

1. Create one more GCP server and configure it as slave node to Jenkins you installed in exercise 1:
   Solution:
   a) Configure Jenkins
      i. Open browser and enter masterIP:8080.
      ii. You should land on a page like this:



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:

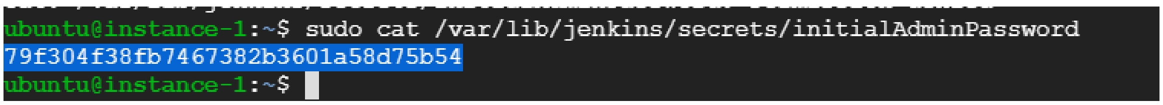
`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

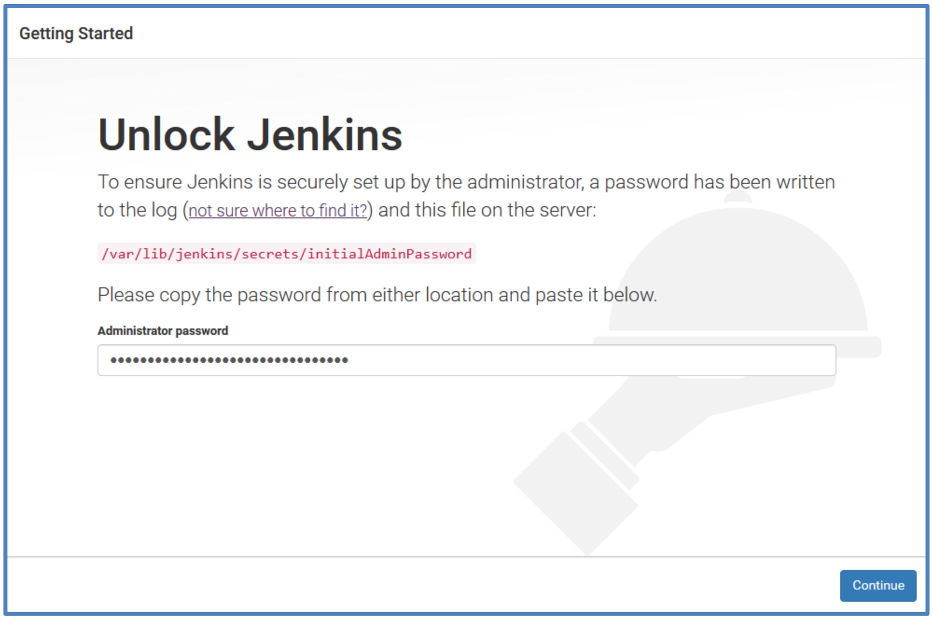
iii. Copy the path mentioned in the page and perform cat operation in master terminal.

```
sudo cat <path>
```



```
ubuntu@instance-1:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
79f304f38fb7467382b3601a58d75b54
ubuntu@instance-1:~$
```

This will give us the password which we will use to unlock our Jenkins.
Copy the password from there and paste it on the Jenkins Server page.



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:

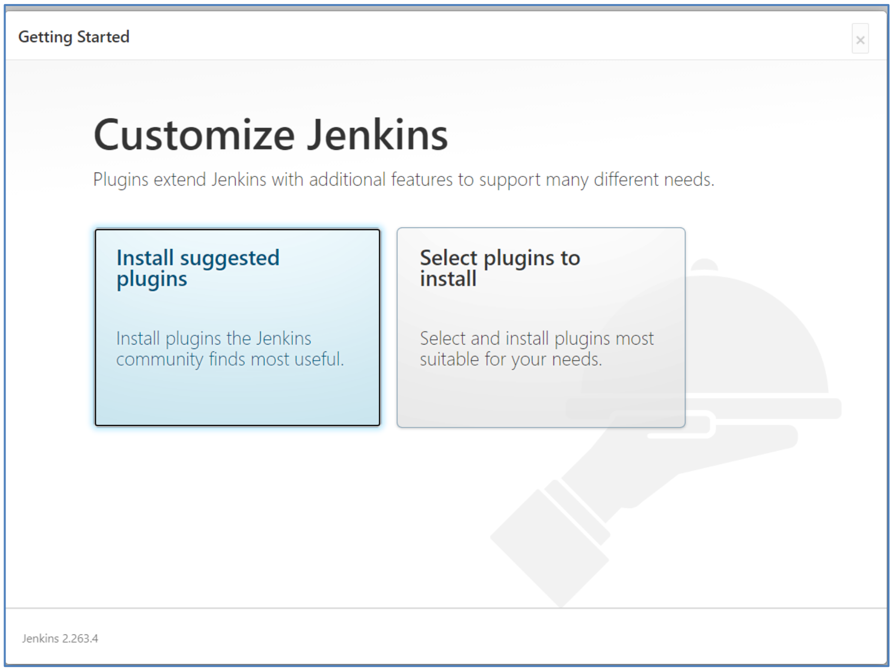
`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

••••••••••••••••••••••••••••••••

iv. Now click on continue. Then click on Install Suggested Plugins.

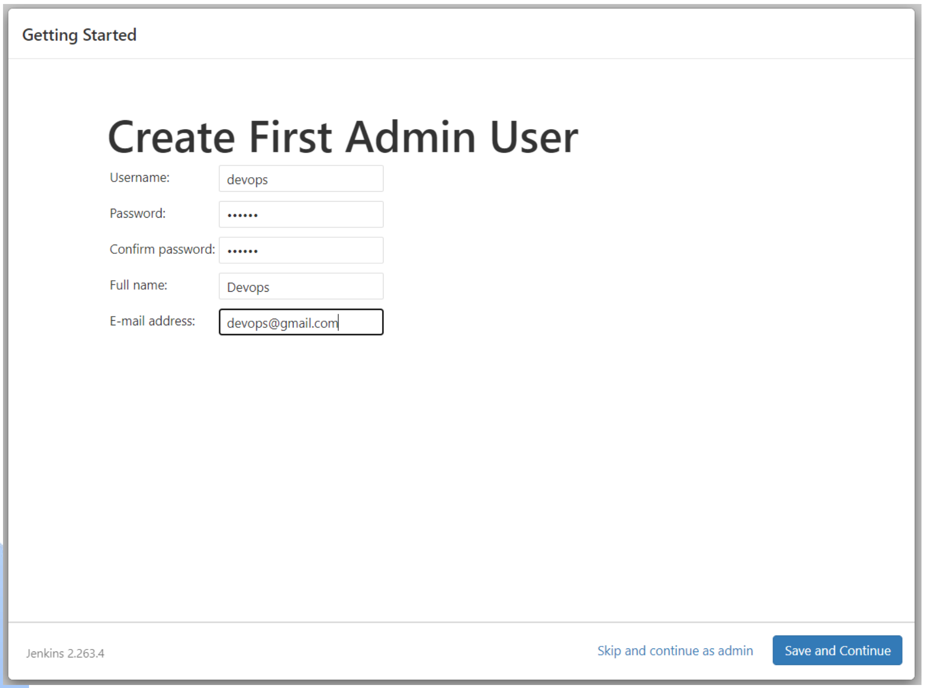

Getting Started

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 2.263.4

v. Once done, enter the Admin User details.


Getting Started

Create First Admin User

Username: devops

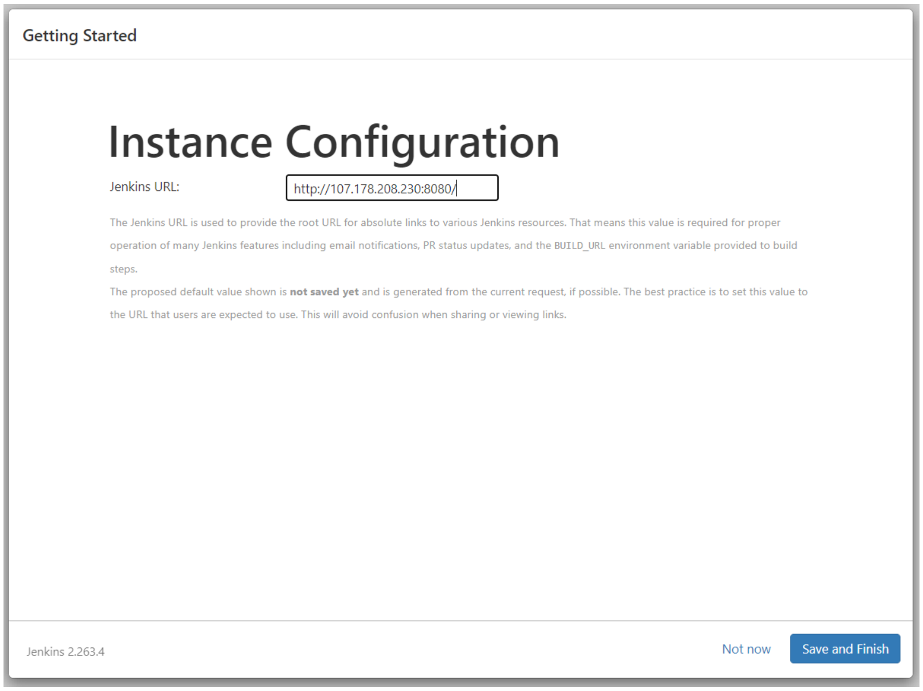
Password: ••••••

Confirm password: ••••••

Full name: Devops

E-mail address: devops@gmail.com

Then click on Save and Continue.



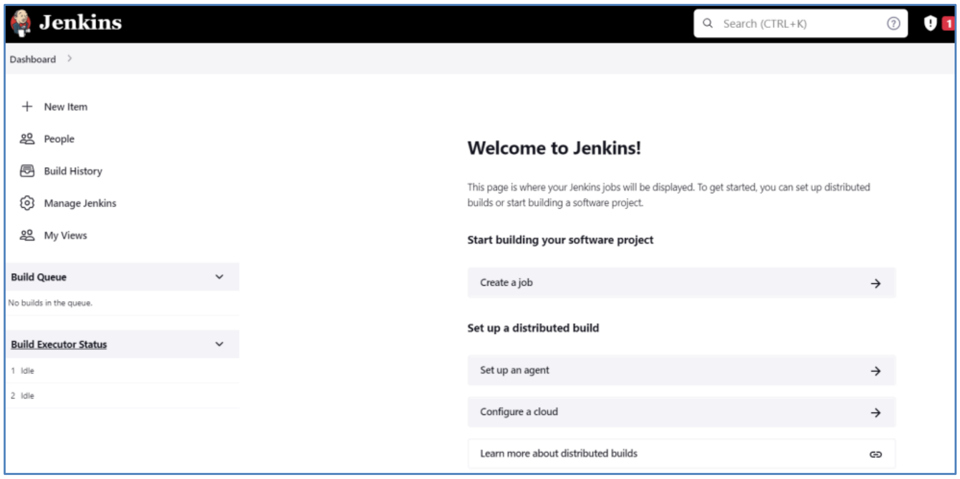
Instance Configuration

Jenkins URL: http://107.178.208.230:8080/

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.

Again, click Save and Finish. Click on Install Suggested Plugins. Once it's done, we will land on a page as shown below.



No builds in the queue.

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.

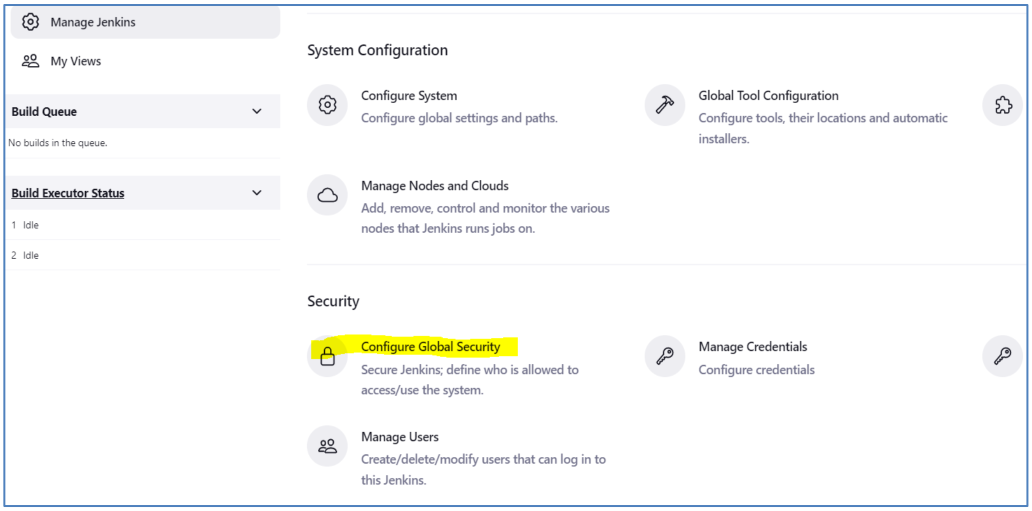
Start building your software project

Set up a distributed build

This is our Jenkins Dashboard.

b) Configure Slave nodes

i. Go to Manage Jenkins. Click on Configure Global Security.



Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

System Configuration

Configure System
Configure global settings and paths.

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

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

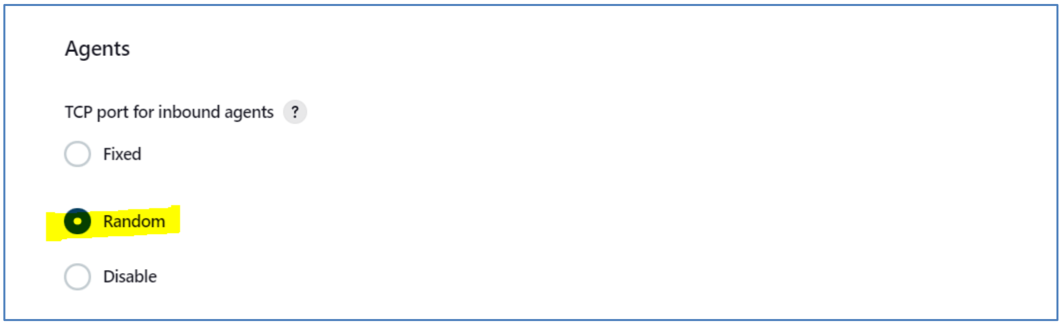
Security

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

Manage Credentials
Configure credentials

Manage Users
Create/delete/modify users that can log in to this Jenkins.

ii. Change the Agents to Random. Then click on Save.



Agents

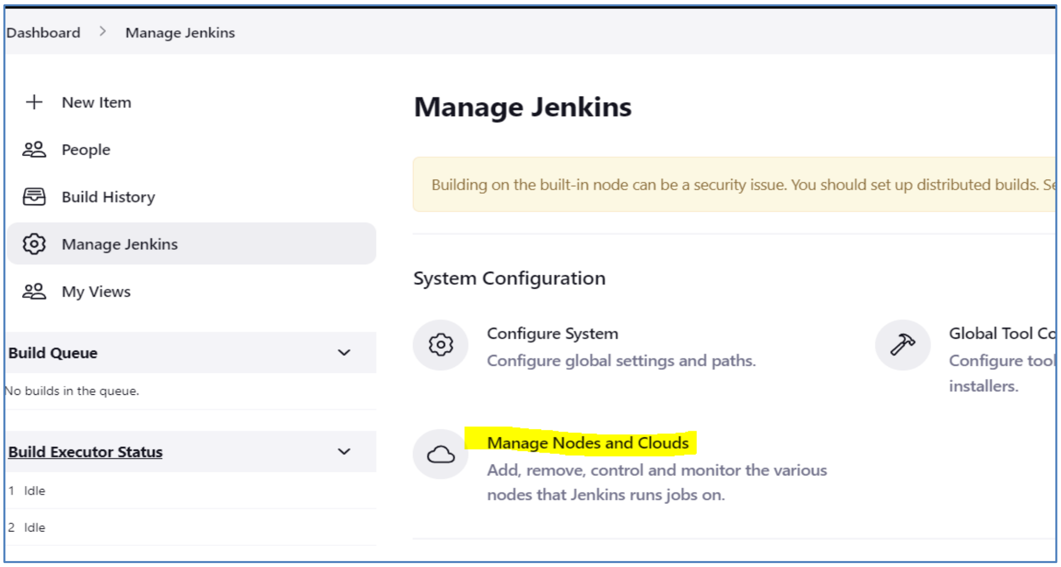
TCP port for inbound agents ?

○ Fixed

● Random

○ Disable

iii. Now go to Manage Nodes.



Dashboard > Manage Jenkins

New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. Se

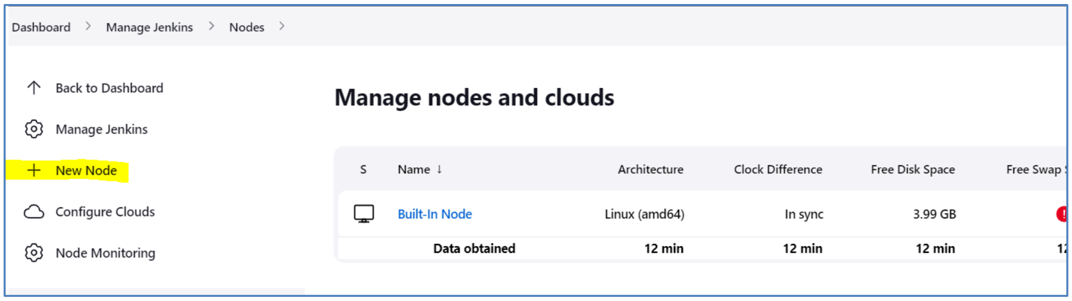
System Configuration

Configure System
Configure global settings and paths.

Global Tool Co
Configure tool
installers.

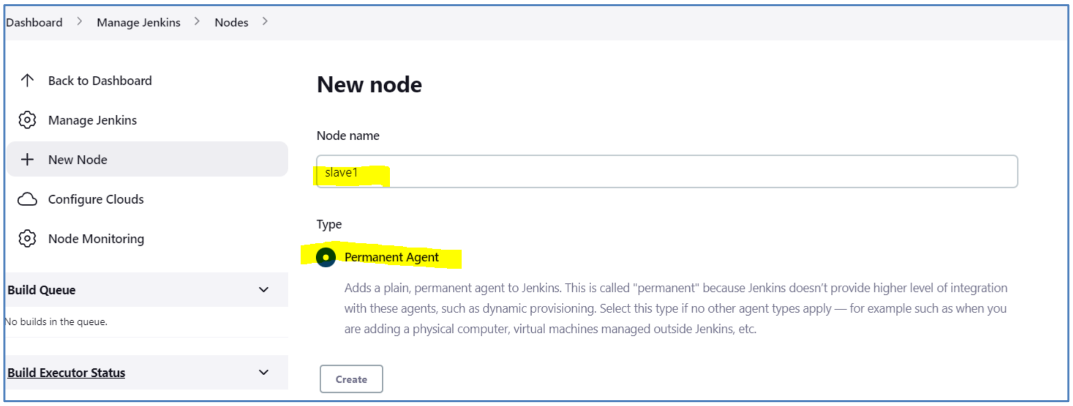
Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

iv. Click on New Node. Add Slave1 as new node and make Permanent Agent. Click on ok.



Manage nodes and clouds

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap
	Built-In Node	Linux (amd64)	In sync	3.99 GB	!
	Data obtained	12 min	12 min	12 min	12



New node

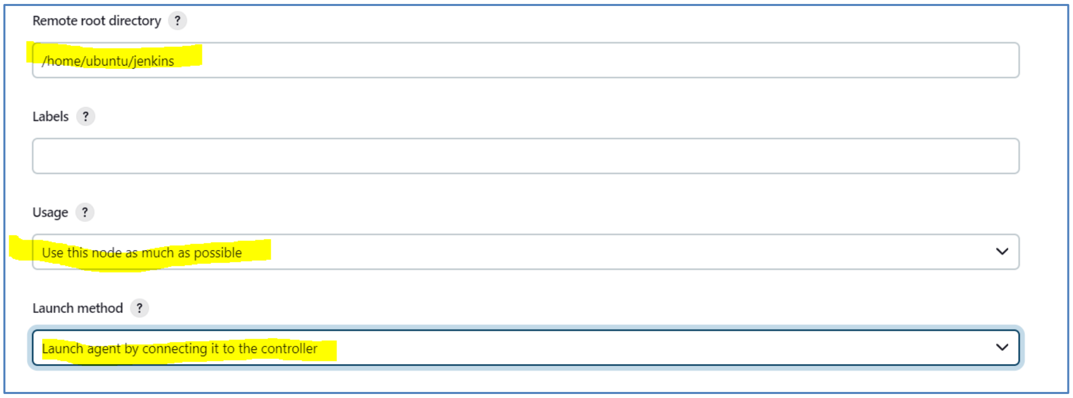
Node name: slave1

Type

● Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

v. Go to Launch method change it to **Launch agent by connecting it to the controller**.



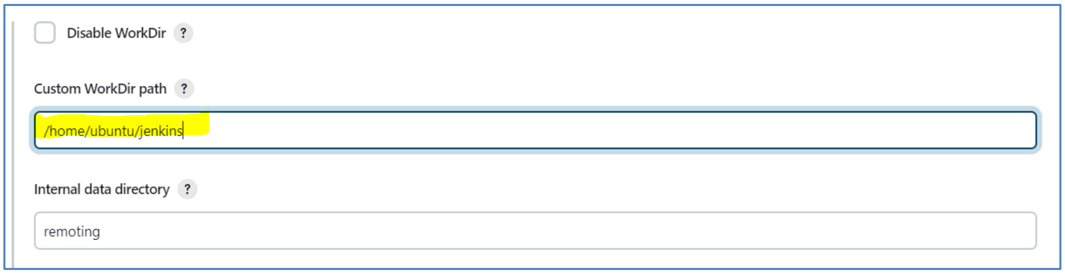
Remote root directory ?: /home/ubuntu/jenkins

Labels ?:

Usage ?: Use this node as much as possible

Launch method ?: Launch agent by connecting it to the controller

vi. Then add the current working directory path to **/home/ubuntu/jenkins**. Then click on Save.



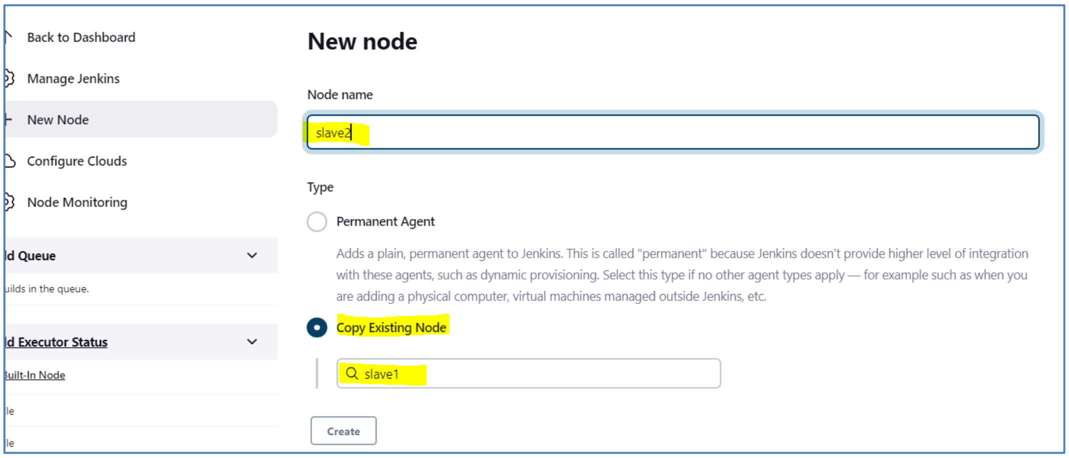
□ Disable WorkDir ?

Custom WorkDir path ?

/home/ubuntu/jenkins

Internal data directory ?

remoting



New node

Node name

slave2

Type

○ Permanent Agent

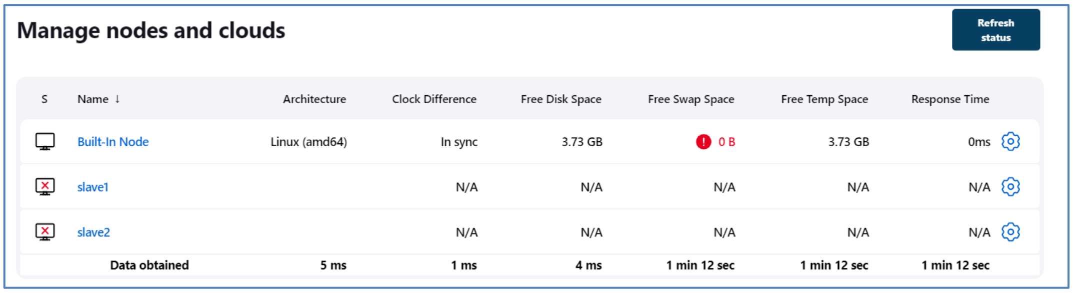
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

● Copy Existing Node

slave1

Create

vii. Then click ok. You can see the list of nodes that we have on the Jenkins Dashboard.

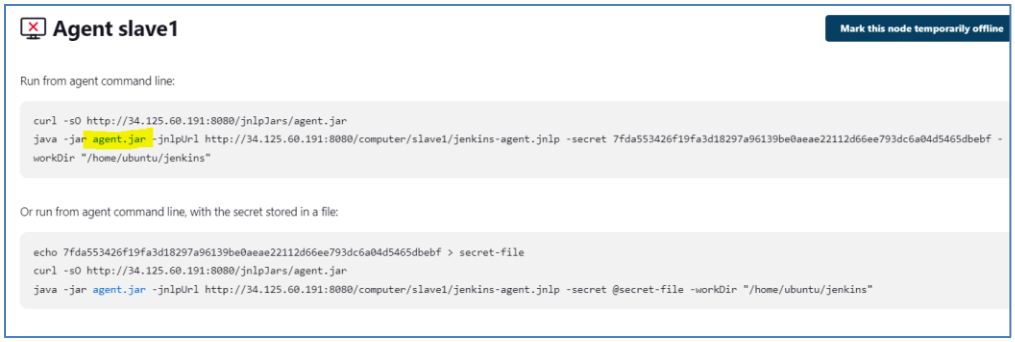


Manage nodes and clouds

Refresh status

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.73 GB	0 B	3.73 GB	0ms
	slave1		N/A	N/A	N/A	N/A	N/A
	slave2		N/A	N/A	N/A	N/A	N/A
	Data obtained	5 ms	1 ms	4 ms	1 min 12 sec	1 min 12 sec	1 min 12 sec

viii. Go to the Jenkins Dashboard, Click on Slave1. Download the Agent.jar file by clicking on it.



Agent slave1

Run from agent command line:

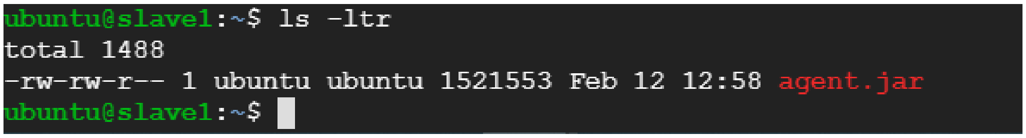
```
curl -sO http://34.125.60.191:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://34.125.60.191:8080/computer/slave1/jenkins-agent.jnlp -secret 7fda553426f19fa3d18297a96139be0aeae22112d66ee793dc6a04d5465dbebf -
workDir "/home/ubuntu/jenkins"
```

Or run from agent command line, with the secret stored in a file:

```
echo 7fda553426f19fa3d18297a96139be0aeae22112d66ee793dc6a04d5465dbebf > secret-file
curl -sO http://34.125.60.191:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://34.125.60.191:8080/computer/slave1/jenkins-agent.jnlp -secret @secret-file -workDir "/home/ubuntu/jenkins"
```

ix. Now download agent.jar file and upload on the slave ubuntu server using scp, winscp or filezilla.

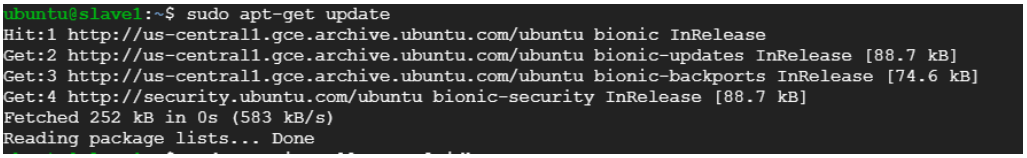
x. Let us verify if the file has been transferred to Slave1 or not. Open a new session of server. Connect to slave1. Run ls command.



```
ubuntu@slave1:~$ ls -ltr
total 1488
-rw-rw-r-- 1 ubuntu ubuntu 1521553 Feb 12 12:58 agent.jar
ubuntu@slave1:~$ 
```

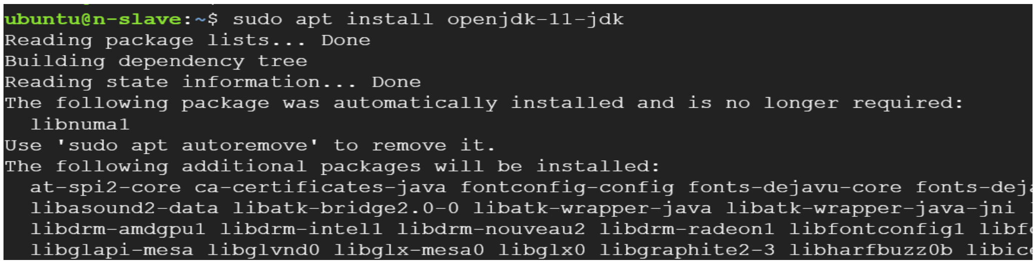
As you can see the agent.jar file appears there, which means our file has been successfully transferred to Slave1.

xi. Before moving ahead install OpenJDK on both Slave1



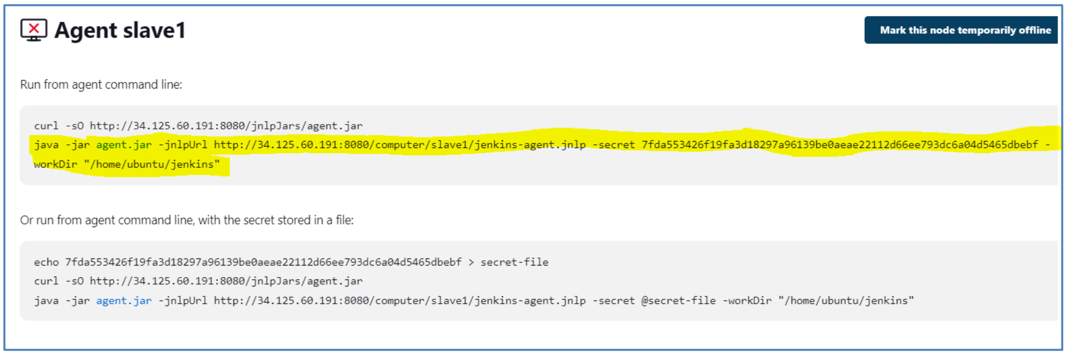
```
ubuntu@slave1:~$ sudo apt-get update
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 252 kB in 0s (583 kB/s)
Reading package lists... Done
```

xii. Now install OpenJDK on the server



```
ubuntu@n-slave:~$ sudo apt install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libnuma1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  at-spi2-core ca-certificates-java fontconfig-config fonts-dejavu-core fonts-dej
  libasound2-data libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni 
  libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libfontconfig1 libf
  libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libharfbuzz0b libic
```

xiii. Connect Slave1 to the Jenkins Server. Go to the Jenkins Dashboard, Click on Slave1, Copy the command line as shown.



Agent slave1

Run from agent command line:

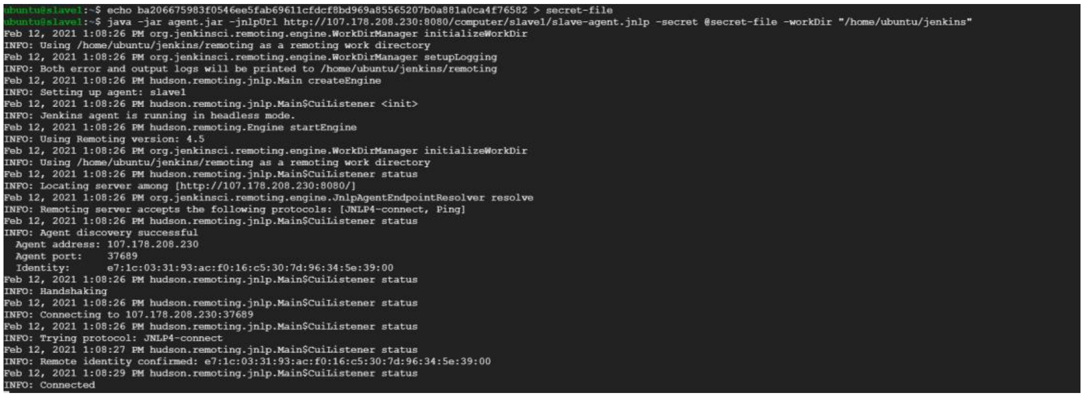
```
curl -sO http://34.125.60.191:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://34.125.60.191:8080/computer/slave1/jenkins-agent.jnlp -secret 7fda553426f19fa3d18297a96139be0aeae22112d66ee793dc6a04d5465dbebf -
workDir "/home/ubuntu/jenkins"
```

Or run from agent command line, with the secret stored in a file:

```
echo 7fda553426f19fa3d18297a96139be0aeae22112d66ee793dc6a04d5465dbebf > secret-file
curl -sO http://34.125.60.191:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://34.125.60.191:8080/computer/slave1/jenkins-agent.jnlp -secret @secret-file -workDir "/home/ubuntu/jenkins"
```

xiv. Run the command on the slave node as shown below.

Note: *If you face any issue with port while connecting agent to master, then open that port on master server by updating the firewall attached to it.*

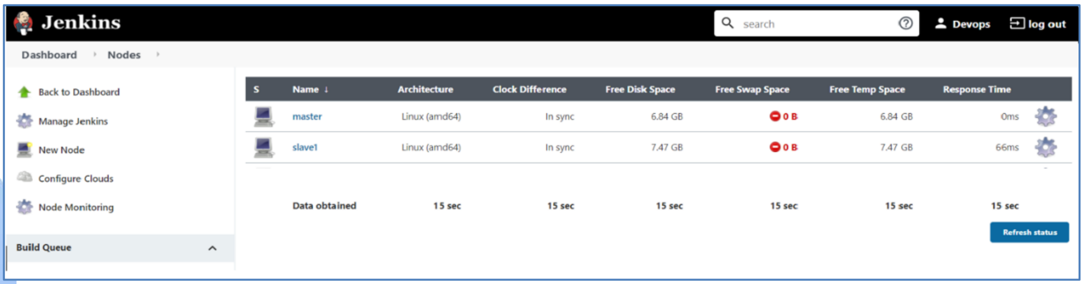


```
ubuntu@slave1:~$ echo ba206675983f0546ee5fab69611cfdcf8bd969a85565207b0a881a0ca4f76582 > secret-file
ubuntu@slave1:~$ java -jar agent.jar -jnlpUrl http://107.178.208.230:8080/computer/slave1/slave-agent.jnlp -secret @secret-file -workDir "/home/ubuntu/jenkins"
Feb 12, 2021 1:08:26 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Feb 12, 2021 1:08:26 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/jenkins/remoting
Feb 12, 2021 1:08:26 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: slave1
Feb 12, 2021 1:08:26 PM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Feb 12, 2021 1:08:26 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.5
Feb 12, 2021 1:08:26 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Feb 12, 2021 1:08:26 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://107.178.208.230:8080/]
Feb 12, 2021 1:08:26 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Feb 12, 2021 1:08:26 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
  Agent address: 107.178.208.230
  Agent port:    37689
  Identity:      e7:1c:03:31:93:ac:f0:16:c5:30:7d:96:34:5e:39:00
Feb 12, 2021 1:08:26 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Feb 12, 2021 1:08:26 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 107.178.208.230:37689
Feb 12, 2021 1:08:26 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Feb 12, 2021 1:08:27 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: e7:1c:03:31:93:ac:f0:16:c5:30:7d:96:34:5e:39:00
Feb 12, 2021 1:08:29 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

It shows “Connected”.

Important Note: Don’t end the Sessions that we just Connected. To perform further operations on Slave1 duplicate the slave server session.

So now that our Slave1 has been connected to Jenkins Server, it look similar to this.



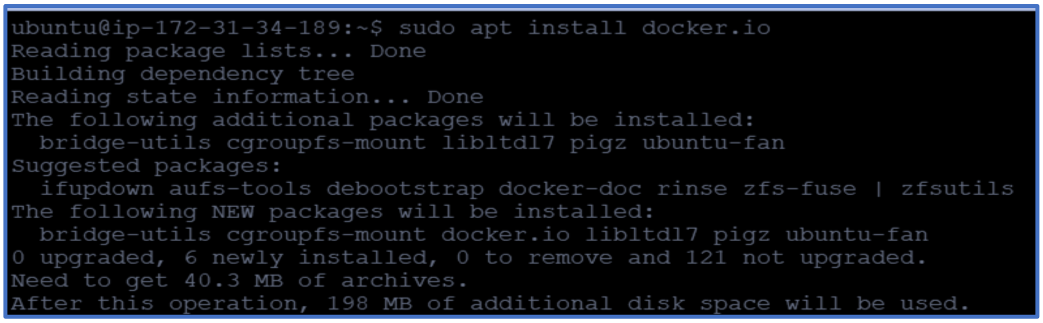
S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	6.84 GB	⊖ 0 B	6.84 GB	0ms
	slave1	Linux (amd64)	In sync	7.47 GB	⊖ 0 B	7.47 GB	66ms
	Data obtained	15 sec	15 sec	15 sec	15 sec	15 sec	15 sec

2. **Create a Jenkins job to clone repo https://github.com/vistasunil/devopsIQ and deploy the website inside it the slave instance in container.**
   Solution:
   i. Open your GitHub account and import the below given repository.

```
https://github.com/vistasunil/devopsIQ
```

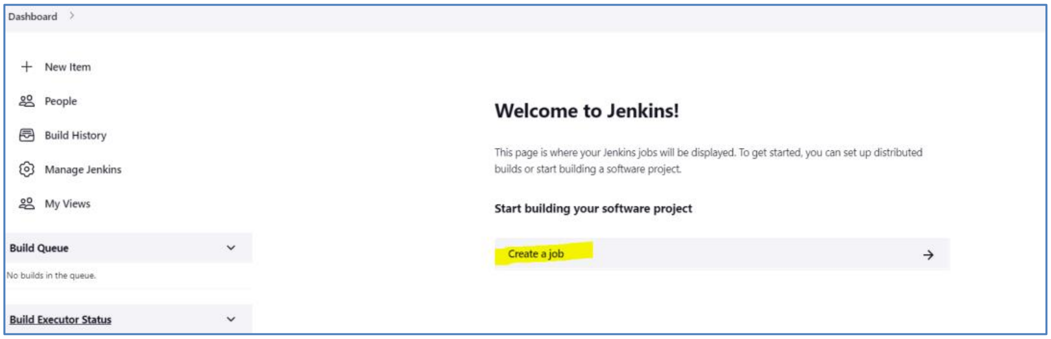
   ii. Install docker on both Slave1

```
sudo apt install docker.io
```



```
ubuntu@ip-172-31-34-189:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount libltdl7 pigz ubuntu-fan
Suggested packages:
  ifupdown aufs-tools debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount docker.io libltdl7 pigz ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 121 not upgraded.
Need to get 40.3 MB of archives.
After this operation, 198 MB of additional disk space will be used.
```

   iii. Open Jenkins Dashboard. Create a new job (Freestyle Project) for Slave1.



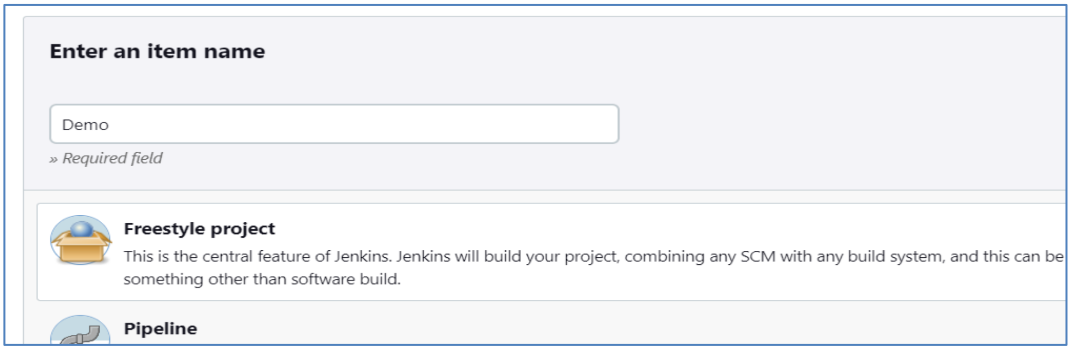
Welcome to Jenkins!

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

No builds in the queue.

   iv. Name the Project as Demo, Select Freestyle Project option.



Enter an item name

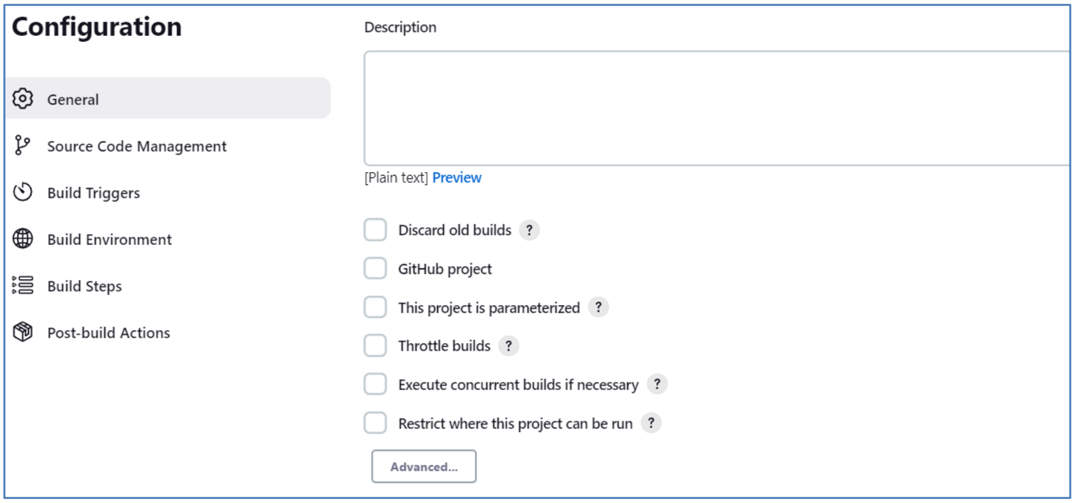
Demo

» 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 something other than software build.

Pipeline

Then click on Ok.
You should land on a page like this.



Configuration

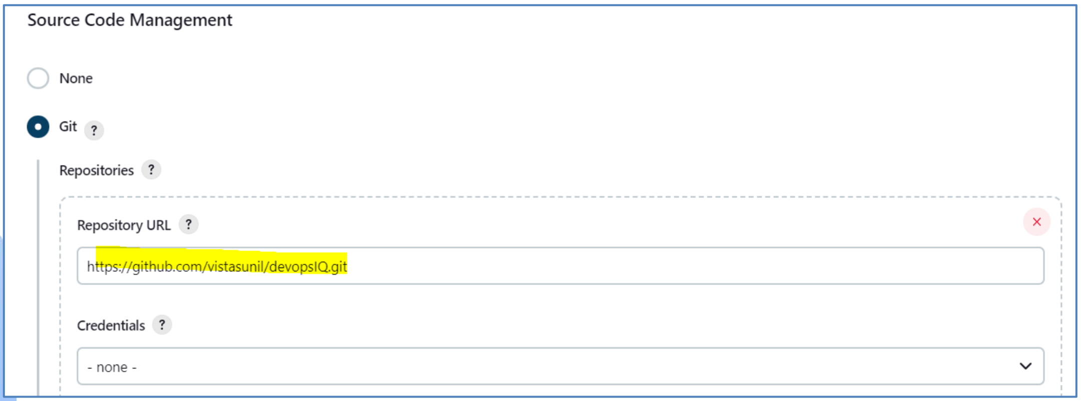
Description

[Plain text] Preview

□ Discard old builds ?
□ GitHub project
□ This project is parameterized ?
□ Throttle builds ?
□ Execute concurrent builds if necessary ?
□ Restrict where this project can be run ?

Advanced...

v. Place your git repository link as shown below.



Source Code Management

○ None
● Git ?

Repositories ?

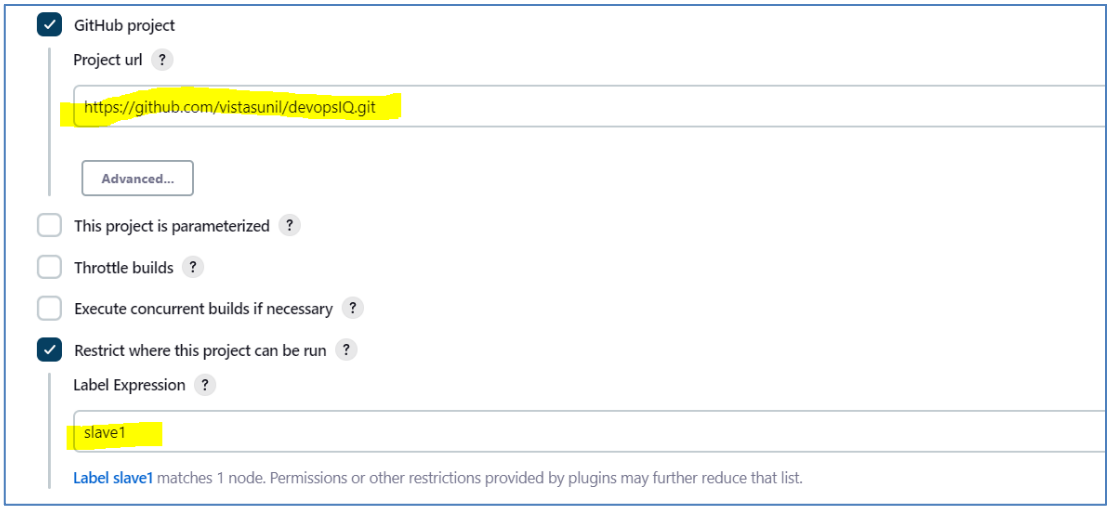
Repository URL ?

https://github.com/vistasunil/devopsIQ.git

Credentials ?

- none -

vi. Click on Restrict where this project can be run. Add Slave1 there.



☑ GitHub project

Project url ?

https://github.com/vistasunil/devopsIQ.git

□ This project is parameterized ?

□ Throttle builds ?

□ Execute concurrent builds if necessary ?

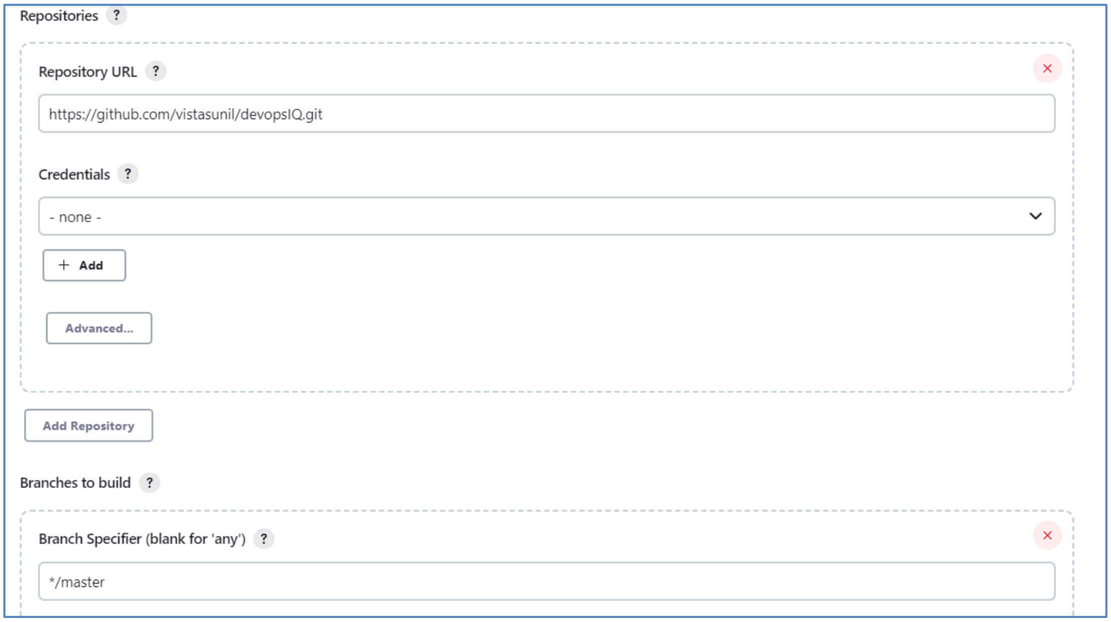
☑ Restrict where this project can be run ?

Label Expression ?

slave1

Label slave1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

vii. Go to Source Code Management, click on git, add the git repository link there as well.



Repositories ?

Repository URL ?

https://github.com/vistasunil/devopsIQ.git

Credentials ?

- none -

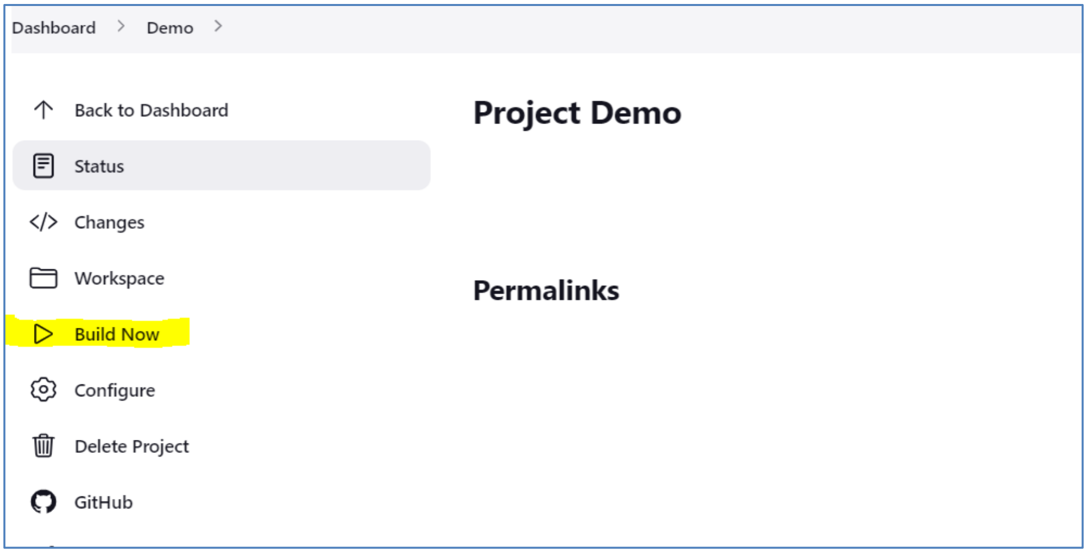
Branches to build ?

Branch Specifier (blank for 'any') ?

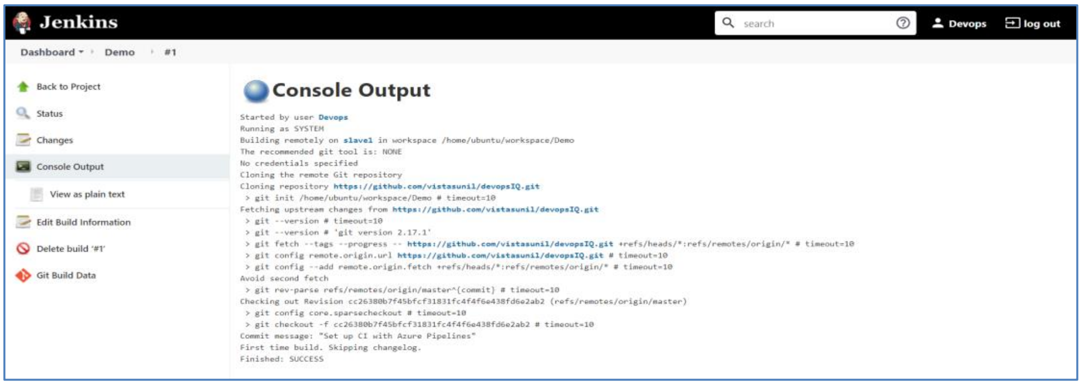
*/master

Click on Save.

viii. Click on Build Now, if the building is done without any error there will be blue circle in the building history.



ix. Click on the blue circle of build #1.

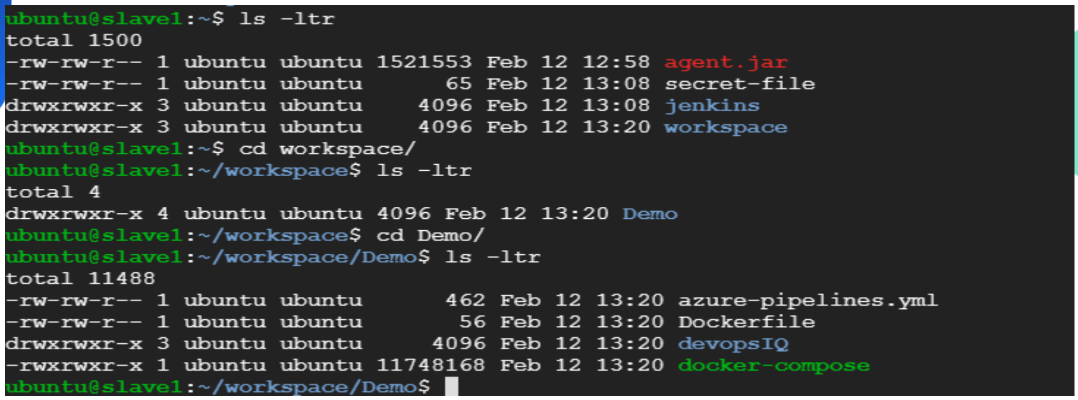


```
Started by user Devops
Running as SYSTEM
Building remotely on slave1 in workspace /home/ubuntu/workspace/Demo
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/vistasunil/devopsIQ.git
 > git init /home/ubuntu/workspace/Demo # timeout=10
Fetching upstream changes from https://github.com/vistasunil/devopsIQ.git
 > git --version # timeout=10
 > git --version # 'git version 2.17.1'
 > git fetch --tags --progress -- https://github.com/vistasunil/devopsIQ.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/vistasunil/devopsIQ.git # 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 cc26380b7f45bfcf31831fc4f4f6e438fd6e2ab2 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f cc26380b7f45bfcf31831fc4f4f6e438fd6e2ab2 # timeout=10
Commit message: "Set up CI with Azure Pipelines"
First time build. Skipping changelog.
Finished: SUCCESS
```

You can see it has been built successfully. Let us verify that.

x. Go to slave1.

```
$ ls -ltr
$ cd workspace
$ ls -ltr
$ cd Demo
$ ls -ltr
```

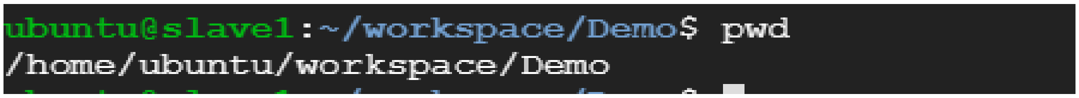


```
ubuntu@slave1:~$ ls -ltr
total 1500
-rw-rw-r-- 1 ubuntu ubuntu 1521553 Feb 12 12:58 agent.jar
-rw-rw-r-- 1 ubuntu ubuntu      65 Feb 12 13:08 secret-file
drwxrwxr-x 3 ubuntu ubuntu    4096 Feb 12 13:08 jenkins
drwxrwxr-x 3 ubuntu ubuntu    4096 Feb 12 13:20 workspace
ubuntu@slave1:~$ cd workspace/
ubuntu@slave1:~/workspace$ ls -ltr
total 4
drwxrwxr-x 4 ubuntu ubuntu 4096 Feb 12 13:20 Demo
ubuntu@slave1:~/workspace$ cd Demo/
ubuntu@slave1:~/workspace/Demo$ ls -ltr
total 11488
-rw-rw-r-- 1 ubuntu ubuntu      462 Feb 12 13:20 azure-pipelines.yml
-rw-rw-r-- 1 ubuntu ubuntu       56 Feb 12 13:20 Dockerfile
drwxrwxr-x 3 ubuntu ubuntu     4096 Feb 12 13:20 devopsIQ
-rwxrwxr-x 1 ubuntu ubuntu 11748168 Feb 12 13:20 docker-compose
ubuntu@slave1:~/workspace/Demo$
```

You can see the repository files there. This means the git repository has been successfully cloned into the Demo job.

Now we will deploy the website that we have stored in our repository.

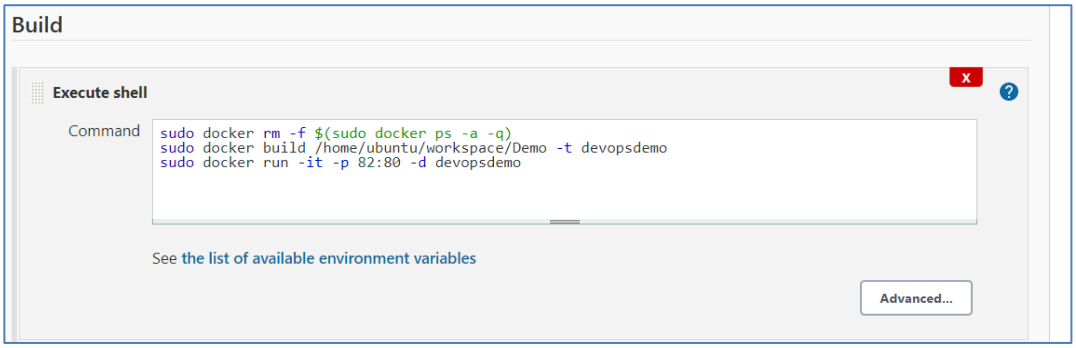
xi. To run the Dockerfile we have to check the copy the present working directory.



```
ubuntu@slave1:~/workspace/Demo$ pwd
/home/ubuntu/workspace/Demo
```

xii. Now go back to configuring the job. Click on Build, then go to Execute shell

```
sudo docker rm -f $(sudo docker ps -a -q)
sudo docker build /home/ubuntu/workspace/Demo -t devopsdemo
sudo docker run -it -p 82:80 -d devopsdemo
```



Command

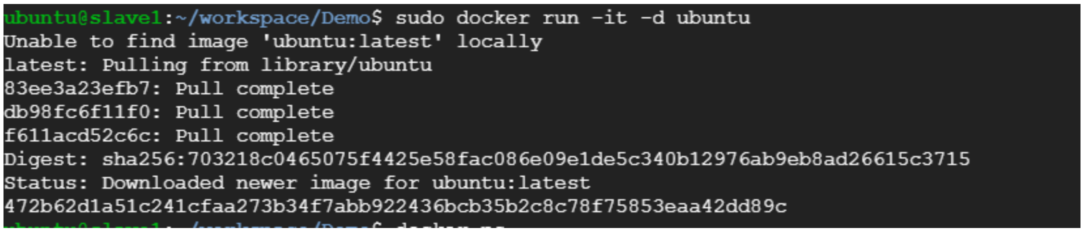
```
sudo docker rm -f $(sudo docker ps -a -q)
sudo docker build /home/ubuntu/workspace/Demo -t devopsdemo
sudo docker run -it -p 82:80 -d devopsdemo
```

See the list of available environment variables

Click on save.

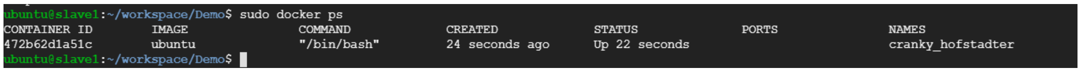
xiii. Before building our job again we must add one arbitrary container in slave1. Add container by performing the following command.

```
$ sudo docker run -it -d ubuntu
```



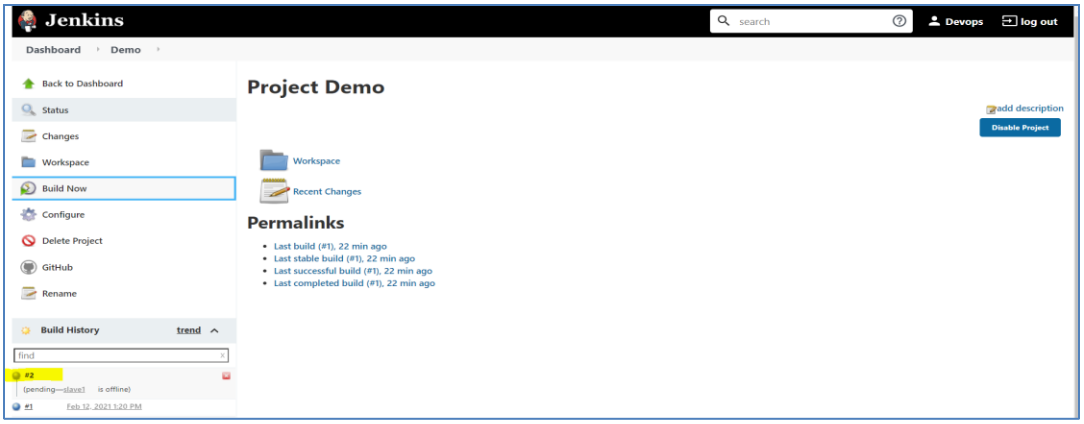
```
ubuntu@slave1:~/workspace/Demo$ sudo docker run -it -d ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
83ee3a23efb7: Pull complete
db98fc6f11f0: Pull complete
f611acd52c6c: Pull complete
Digest: sha256:703218c0465075f4425e58fac086e09e1de5c340b12976ab9eb8ad26615c3715
Status: Downloaded newer image for ubuntu:latest
472b62d1a51c241cfaa273b34f7abb922436bcb35b2c8c78f75853eaa42dd89c
```

Now we have added a container as below



```
ubuntu@slave1:~/workspace/Demo$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
472b62d1a51c        ubuntu              "/bin/bash"         24 seconds ago      Up 22 seconds                           cranky_hofstadter
ubuntu@slave1:~/workspace/Demo$
```

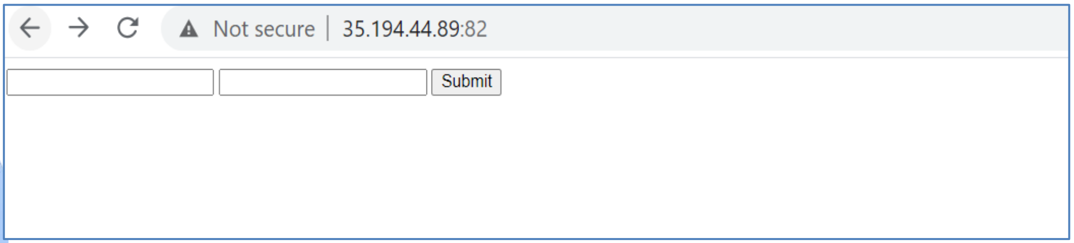
xiv. Now open Jenkins Dashboard and build the project.



- Last build (#1), 22 min ago
- Last stable build (#1), 22 min ago
- Last successful build (#1), 22 min ago
- Last completed build (#1), 22 min ago

Building was successful.

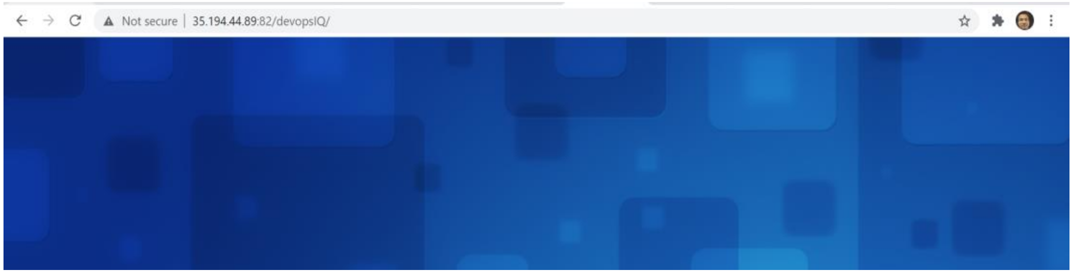
xv. Now open browser and enter Slave1 Public IP:82. You can get the IP against server name in GCP console



Not secure | 35.194.44.89:82

This is the apache page that means our container is working perfectly.

xvi. Now enter slave1 Public IP:82/devopsIQ/ in the browser.



Website is accessible successfully.