

Setting up jenkins master-slave architecture for testing and production servers

1. Setting up master and slave servers in aws.

=>setup 3 nodes of type t2.micro of ubuntu 20.04 os with security group instructions to permit all traffic and keep all other details as default.

=>NOTE : we will be installing Jenkins only on master nodes and monitor slave nodes using master nodes.

=>On master node execute these instructions

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 3

Purchasing option: Request Spot Instances

Network: vpc-b917d2d2 (default)

Subnet: No preference (default subnet in any Availability Zone)

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory

IAM role: None

Shutdown behavior: Stop

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring

Buttons: Cancel, Previous, Review and Launch (highlighted in blue), Next: Add Storage

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The 'Instances' section is currently selected and expanded, showing three instances: 'Master', 'Slave1', and 'Slave2'. Each instance has a status of 'Running'. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. At the bottom of the page, there are links for Feedback, English (US), and a footer note about copyright.

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 DNS |
|--------|---------------------|----------------|---------------|--------------|--------------|-------------------|-----------------------|
| Master | i-036ef866e7e014e67 | Running | t2.micro | Initializing | No alarms | ap-south-1b | ec2-3-7-254-118.ap... |
| Slave1 | i-00f4643d005d8f411 | Running | t2.micro | Initializing | No alarms | ap-south-1b | ec2-15-207-21-112 |
| Slave2 | i-05e8fcff6c7fecb2 | Running | t2.micro | Initializing | No alarms | ap-south-1b | ec2-65-2-6-190.ap... |

COMMANDS

```

sudo apt install openjdk-8-jdk
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo
apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'
sudo apt update
sudo apt install jenkins

```

=>On completing the above instructions, access this server on the browser using a public ipv4 address with port number 8080 i.e ipaddress:8080.

=>after opening the above jenkins page login to jenkins to proceed further

Instance details | EC2 Management C | DevOps/jenkins-masterSlave at main | Setup Wizard [Jenkins] | +

Not secure | http://3.7.254.118:8080

Spinbot.com - Artic... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Jenkins

Getting Started

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

Jenkins 2.303.3

The screenshot shows the Jenkins Setup Wizard's 'Getting Started' page. A dependency tree is overlaid on the sidebar, listing various Jenkins components and their dependencies. The tree includes 'GitHub API', 'GitHub', 'GitHub Branch Source', 'Pipeline: GitHub Groovy Libraries', 'Pipeline: Stage View', 'Git', 'SSH Build Agents', 'Matrix Authorization Strategy', 'PAM Authentication', and 'LDAP'. A note at the bottom of the tree states '** - required dependency'.

Instance details | EC2 Management C | DevOps/jenkins-masterSlave at main | Setup Wizard [Jenkins] | +

Not secure | http://3.7.254.118:8080

Spinbot.com - Artic... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Jenkins

Create First Admin User

Username: Team12

Password:

Confirm password:

Full name: LaxmiNarayana K

E-mail address: laxminarayana33317@gmail.com

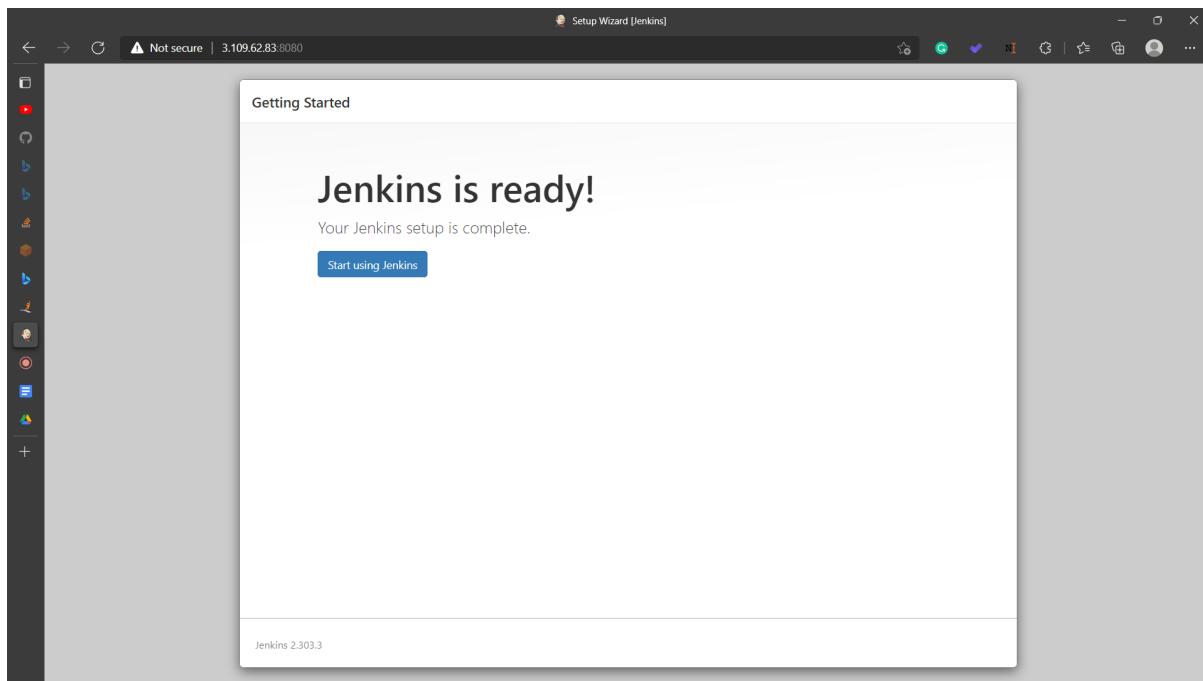
Manage...

Jenkins 2.303.3

Skip and continue as admin

Save and Continue

The screenshot shows the 'Create First Admin User' step of the Jenkins Setup Wizard. It requires the user to enter a username ('Team12'), password, and confirm password. Below these fields, there are input fields for 'Full name' (containing 'LaxmiNarayana K') and 'E-mail address' (containing 'laxminarayana33317@gmail.com'). A 'Manage...' button is also present. At the bottom, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'.



2.configuring jenkins

=>Goto manage jenkins -> configure global settings -> Agents -> random -> click on save.
=>Goto manage jenkins -> manage nodes -> new node.

A screenshot of a web browser window titled "Dashboard [Jenkins]". The address bar shows "Not secure | http://3.7.254.118:8080". The main content area is titled "Welcome to Jenkins!". It includes a search bar, user information (LaxmiNarayana), and a log out link. On the left, there is a sidebar with links like "New Item", "People", "Build History", "Manage Jenkins", "My Views", "Lockable Resources", and "New View". The main area has sections for "Start building your software project" (with a "Create a job" button) and "Set up a distributed build" (with "Set up an agent" and "Configure a cloud" buttons). At the bottom, there are sections for "Build Queue" (empty) and "Build Executor Status" (two idle executors). A footer at the bottom right includes links for "REST API" and "Jenkins 2.303.3".

=>Create 2 slave nodes named slave-1 and slave-2 respectively with permanent agent option enabled and with the launch method as configure when master is triggered.Also add custom workdir path to be /home/ubuntu/jenkins and click on save.

| S | Name ↓ | Architecture | Clock Difference | Free Disk Space | Free Swap Space | Free Temp Space | Response Time |
|---|--------|---------------|------------------|-----------------|-----------------|-----------------|---------------|
| | master | Linux (amd64) | In sync | 5.38 GB | 0 B | 5.38 GB | 0ms |
| | Slave1 | | N/A | N/A | N/A | N/A | N/A |
| | Slave2 | | N/A | N/A | N/A | N/A | N/A |

=>click on respective agents and download their agent.jar executable file.

=>Now these agent.jar files are to be sent to the respective slave nodes that were created from above points.For ubuntu os run the following command.

```
scp -i.< pemfileofmasternode> ./agent.jar <username>@< ipv4addressofslav enode>:<destinationfolderofslavenode>
```

3. Connect to slave nodes from master nodes.

The Jenkins Slave1 configuration page shows the following details:

- Agent Slave1**: A section for connecting the agent to Jenkins via browser or command line.
- Build Executor Status**: Shows "None" under Build Executor Status.
- Projects tied to Slave1**: Shows "None" under Projects tied to Slave1.

The AWS FileZilla interface shows a file transfer between two EC2 instances:

- Local site:** C:\Users\Anu\Downloads\DevOps\
- Remote site:** /home/ubuntu
- Selected files:** agent.jar (1,507,813 bytes)
- Transfer statistics:** 5 files and 2 directories. Total size: 1,512,611 bytes.

=> Goto manage jenkins -> manage nodes -> click on respective slave and copy paste the code under Run from agent command line.

=> Now copy paste the code in slave nodes.

COMMANDS

sudo apt-get update

sudo apt install openjdk-8-jdk

```

Nov 18, 2021 3:56:18 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Nov 18, 2021 3:56:18 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/jenkins/remoting
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: Slave2
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Nov 18, 2021 3:56:19 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.10.1
Nov 18, 2021 3:56:19 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/jenkins/remoting as a remoting work directory
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://3.7.254.118:8080/]
Nov 18, 2021 3:56:19 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
    Agent address: 3.7.254.118
    Agent port: 37103
    Identity: a4:17:92:03:c1:90:2e:56:24:06:79:5d:6d:66:04:ab
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to 3.7.254.118:37103
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Nov 18, 2021 3:56:19 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Nov 18, 2021 3:56:19 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: a4:17:92:03:c1:90:2e:56:24:06:79:5d:6d:66:04:ab
Nov 18, 2021 3:56:21 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected

```

i-05e8fcff62c7fecb2 (Slave2)

Public IPs: 65.2.6.190 Private IPs: 172.31.0.192

Slave-1 [Jenkins]

Not secure | http://3.109.62.83:8080/computer/Slave-1/

Jenkins

Dashboard > Nodes > Slave-1

Agent Slave-1

Agent is connected.

Projects tied to Slave-1

None

Build Executor Status

1 Idle

REST API Jenkins 2.303.3

| S | Name | Architecture | Clock Difference | Free Disk Space | Free Swap Space | Free Temp Space | Response Time |
|---|----------------------|---------------|------------------|-----------------|-----------------|-----------------|----------------|
| | master | Linux (amd64) | In sync | 5.38 GB | 0 B | 5.38 GB | 0ms |
| | Slave1 | Linux (amd64) | In sync | 5.73 GB | 0 B | 5.73 GB | 24ms |
| | Slave2 | Linux (amd64) | In sync | 5.74 GB | 0 B | 5.74 GB | 74ms |
| | Data obtained | | 7.5 sec | 7.5 sec | 7.4 sec | 7.4 sec | 7.4 sec |

Build Executor Status

| Node | Status |
|--------|--------|
| master | 1 Idle |
| Slave1 | 2 Idle |
| Slave2 | 1 Idle |

=>Now verify that the master and slave nodes are connected

4. Configuring Jenkins to build a project on slave-1(Test server) if successful build on slave-2(Production server).

=>NOTE : make sure to duplicate sessions in the slave node so that connection doesn't terminate.

=>Install Docker on both the slaves.

commands

```
sudo apt-get install docker.io
```

Enter an item name

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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

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

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

Organization Folder

=>Head Back to jenkins. create new job -> make it new freestyle project -> name it slave-1.Now under configure in the general section click on github project and copy paste the url of github project.Also enable the Restrict where this project can be run and type in slave-1 under label expression section.Go to source code management and enable git.

=>Under the build section click Add build step -> Execute shell.Now run the below commands.

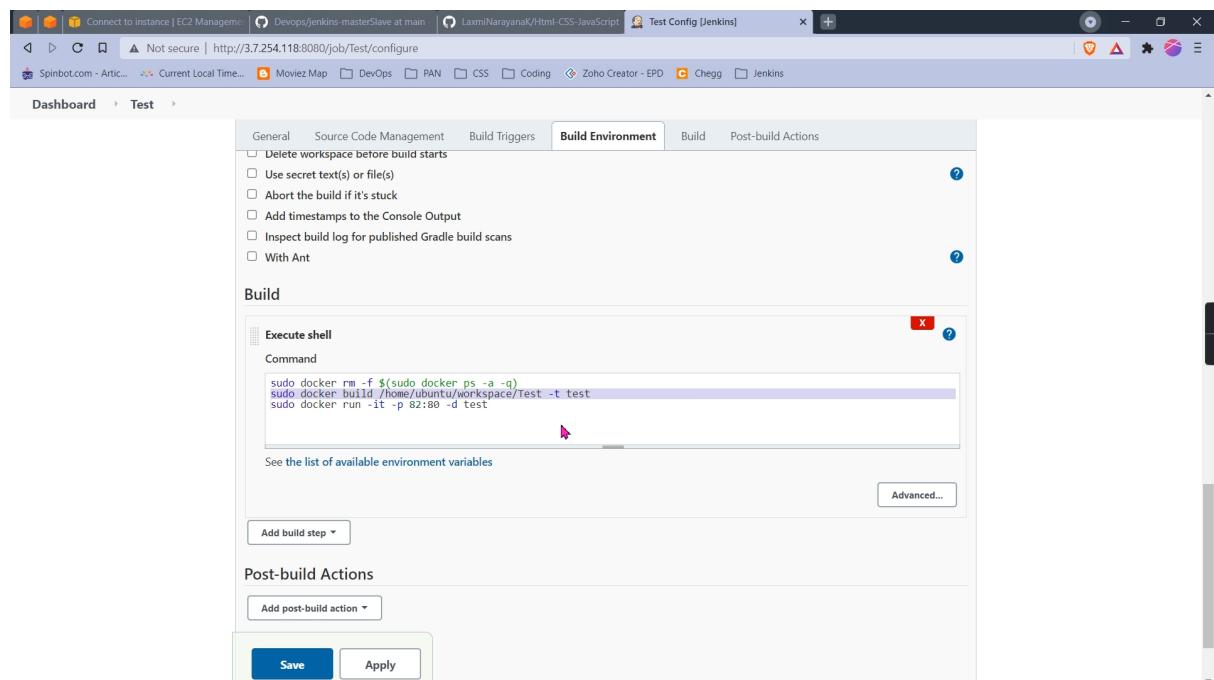
COMMANDS

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

=>NOTE : Make sure to run a custom container on the slave node before executing the above commands.The above steps are to be executed for both test and production server.

=>Now click on build.

=>To check the above build step headover to :/



Screenshot of a Jenkins job page titled "Build #3 (Nov 14, 2021 12:38:28 PM)".

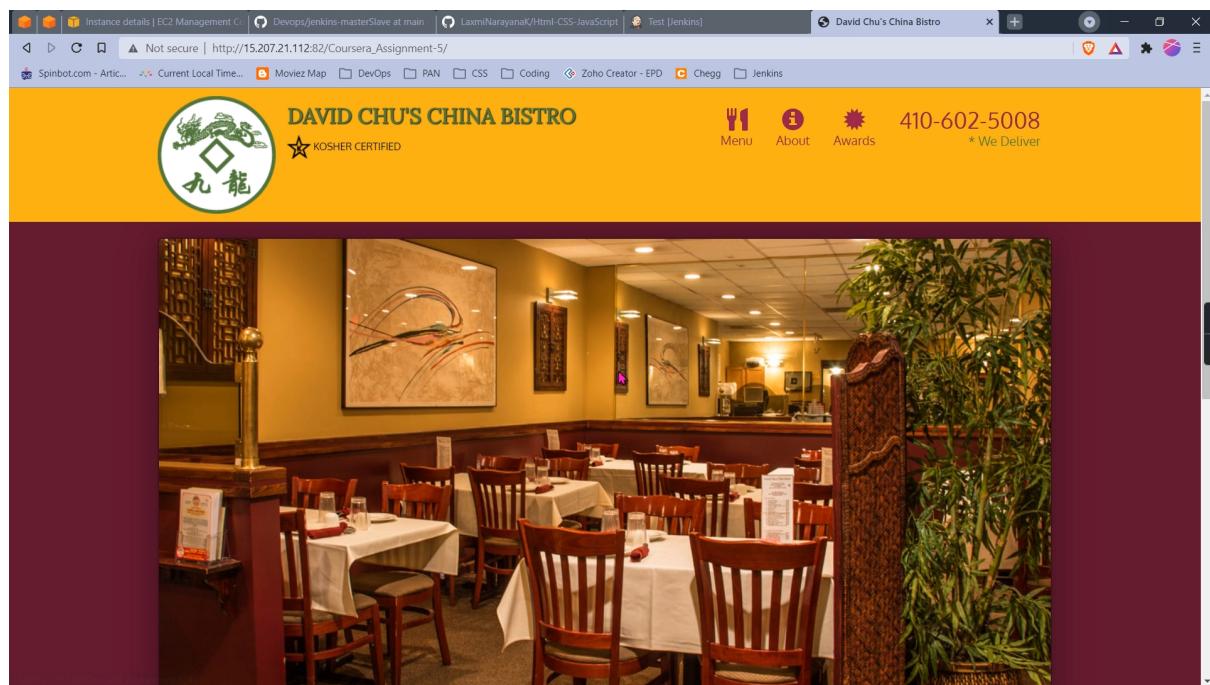
The page shows the following details:

- Status: No changes.
- Started by user: jaganmohan reddy
- Revision: 6be43b10a8dafde9e2f0da9f674682e1e12968c
- Repository: <https://github.com/LaxmiNarayanaK/web-app-test.git>
- refs/remotes/origin/master

Buttons at the top right include "Keep this build forever" and "add description". A note indicates the build took 16 sec on Slave-1.

Navigation links on the left include: Back to Project, Status, Changes, Console Output, Edit Build Information, Delete build '#3', Git Build Data, and Previous Build.

Bottom right corner shows "REST API" and "Jenkins 2.303.3".



```

es: ubuntu@ip-172-31-10-219: ~/workspace
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@ip-172-31-10-219: $ client_loop: send disconnect: Connection reset by peer
And@And-MINGW64 ~ /Downloads/DevOps
$ ssh -i "Devops.pem" ubuntu@ec2-15-207-21-112.ap-south-1.compute.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

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

System information as of Thu Nov 18 16:05:12 UTC 2021

System load: 0.08 Processes: 109
Usage of /: 31.4% of 7.69GB Users logged in: 1
Memory usage: 39% IPv4 address for docker0: 172.17.0.1
Swap usage: 0% IPv4 address for eth0: 172.31.10.219

34 updates can be applied immediately.
22 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Nov 18 15:57:42 2021 from 157.47.30.87
ubuntu@ip-172-31-10-219: $ sudo docker build ./home/ubuntu/workspace/Test -t test
Sending build context to Docker daemon 54.93MB
Step 1/2 : FROM hshar/webapp
latest: Pulling from hshar/webapp
a48c500ed24e: Pull complete
le1de0ff7e71: Pull complete
0330ca45a200: Pull complete
471dab38c9fbf: Pull complete
0b1aa0a48707: Pull complete
c293eae0f5f7: Pull complete
1b3d4a75273: Pull complete
4c2c0ff3ce8: Pull complete
Digest: sha256:3cfcac1a26c01410dcc9cbc57252b50d9ed2f31a2dc24e3f066c61b88e839b
Status: Downloaded newer image for hshar/webapp:latest
--> 0bc1f535ed5
Step 2/2 : ADD ./Coursera_Assignment-5 /var/www/html/Coursera_Assignment-5
--> 0038e6f76661
Successfully built 0038e6f76661
Successfully tagged test:latest
ubuntu@ip-172-31-10-219: $ ls
agent.jar Jenkins workspace
ubuntu@ip-172-31-10-219: $ cd workspace
ubuntu@ip-172-31-10-219: ~/workspace$ -

```

Dashboard > Prod >

- General
- Source Code Management
- Build Triggers**
- Build Environment
- Build
- Post-build Actions

build periodically

GitHub hook trigger for GITScm polling

Poll SCM

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

Execute shell

Command

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

See the list of available environment variables

Post-build Actions

Add build step > Save Apply

```

ubuntu@ip-172-31-0-192:~/jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Processing triggers for systemd (245,4-4ubuntu3.13) ...
Processing triggers for man-db (2,9.1-1) ...
Processing triggers for dbus (1,12,16-2ubuntu2.1) ...
Processing triggers for libc-bin (2,31-0ubuntu9.2) ...
ubuntu@ip-172-31-0-192:~$ client_loop: send disconnect: Connection reset by peer
[41@Ant1 MINIGW64 ~/Downloads/DevOps]
$ ssh -i "Devops.pem" ubuntu@ec2-65-2-6-190.ap-south-1.compute.amazonaws.com
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1020-aws x86_64)

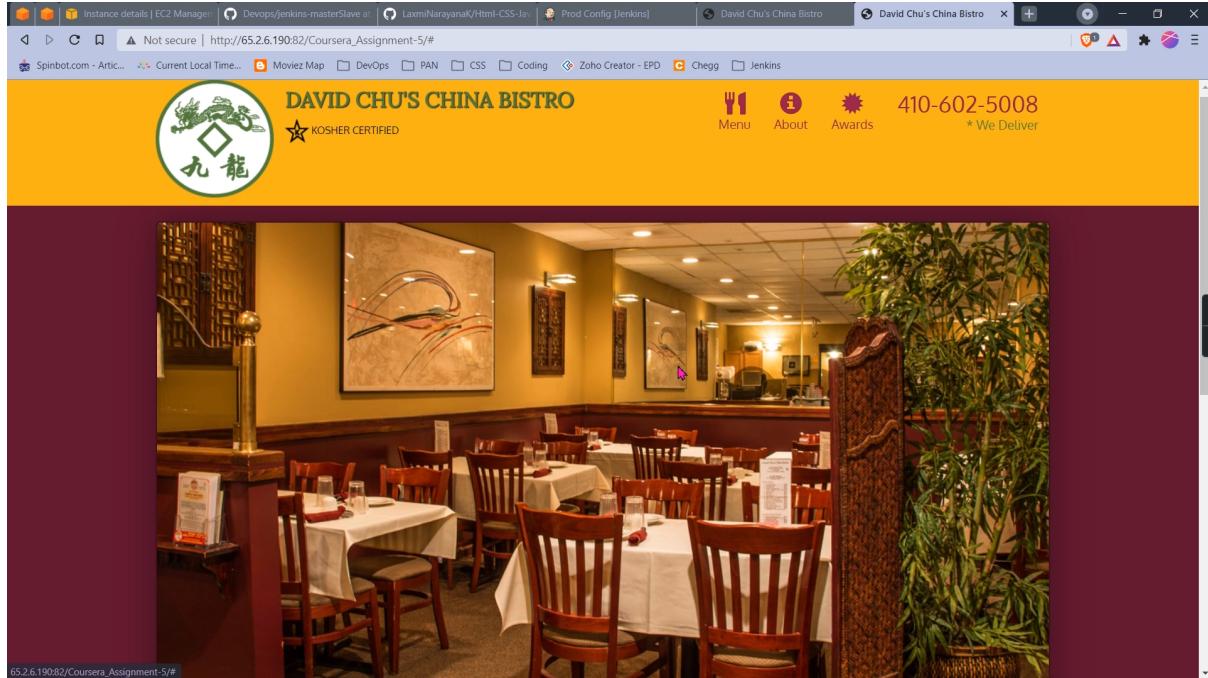
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Thu Nov 18 16:07:58 UTC 2021

 System load: 0.0      Processes:          109
 Usage of `/': 31.3% of 7.69GB  Users logged in: 1
 Memory usage: 39%          IPv4 address for docker0: 172.17.0.1
 Swap usage:  0%          IPv4 address for eth0: 172.31.0.192

34 updates can be applied immediately.
22 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Nov 18 15:58:26 2021 from 157.47.30.87
ubuntu@ip-172-31-0-192:~$ sudo docker build /home/ubuntu/jenkins/workspace/Prod -t production
Sending build context to Docker daemon 54.93MB
Step 1/2 : FROM hshar/webapp
latest: Pulling from hshar/webapp
a48c500ed24e: Pull complete
fe1de00ff7e1: Pull complete
0030ca45a200: Pull complete
4711404ff: Pull complete
Digest: sha256:3cfcba1a26c01410dcc9cbc57252b50d9ed2f31a2dc24e3f066c61b88e839b
Status: Downloaded newer image for hshar/webapp:latest
--> 0b6cf1535ed5
Step 2/2 : ADD ./Coursera_Assignment-5 /var/www/html/Coursera_Assignment-5
Successfully built 3ffbf331e05
Successfully tagged production:latest
ubuntu@ip-172-31-0-192:~$ ls
agent.jar  jenkins
ubuntu@ip-172-31-0-192:~$ cd jenkins
ubuntu@ip-172-31-0-192:~/jenkins$
```



=>Now we need to configure the servers in such a way that after testing production is built.To enable this Goto the test(freestyle project of slave-1) and under the post build actions click on build other projects and enter production(freestyle project of slave-2).

5. Build in the ci-cd pipeline for test and production servers.

=>Headover to manage jenkins -> manage plugins -> available.Now search for build pipeline,install the plugin.

=>In the jenkins home page click on the + sign near the All.

=>Now click the Build pipeline view, name the view as CiCd.Also under Build pipeline view title give CiCd and with other options as default and click save.

Dashboard > Plugin Manager

Available Plugins [Jenkins] | David Chu's China Bistro | David Chu's China Bistro

Not secure | http://3.7.254.118:8080/pluginManager/available

Spinbot.com - Artic... Current Local Time... Moviez Map DevOps PAN CSS Coding Zoho Creator - EPD Chegg Jenkins

Jenkins

search log out

Back to Dashboard Manage Jenkins Update Center

Updates Available Installed Advanced

Install Name Version Released

Build Pipeline

Build Tools Other Post-Build Actions User Interface

This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.

Warning: This plugin version may not be safe to use. Please review the following security notices:

- Stored XSS vulnerability

| | | | |
|--|---|------------------|-----------------|
| <input type="checkbox"/> Webhook Step | Allows build pipelines to wait for notification from an external system before continuing. | 80.v6737a5fd857b | 2 days 5 hr ago |
| <input type="checkbox"/> Pipeline timeline | An interactive build timeline to help you visualize your build pipeline and identify bottlenecks. | 1.0.3 | 2 yr 9 mo ago |

Install without restart Download now and install after restart Update information obtained: 29 min ago Check now

REST API Jenkins 2.303.3

Dashboard > Test

General Source Code Management Build Triggers Build Environment Build Post-build Actions

With Ant

Build

Execute shell

Command

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

See the list of available environment variables Advanced...

Add build step ▾

Post-build Actions

Build other projects

Projects to build

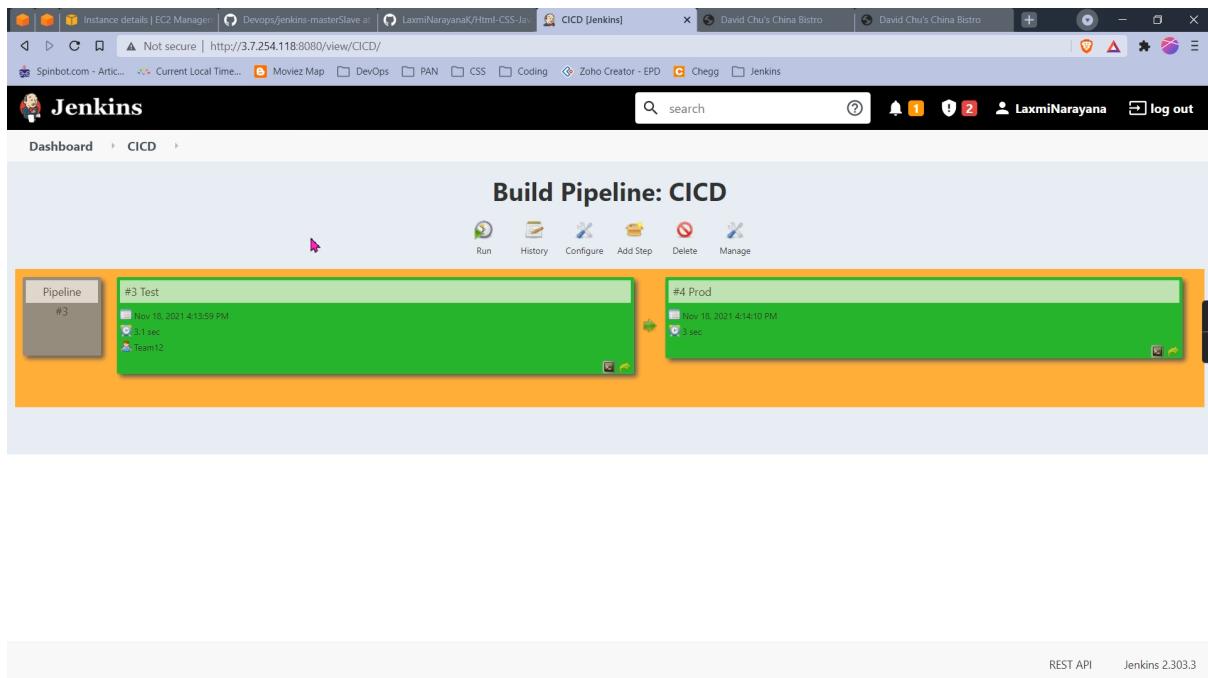
Prod. No such project 'pro'. Did you mean 'Prod.'?

Trigger only if build is stable

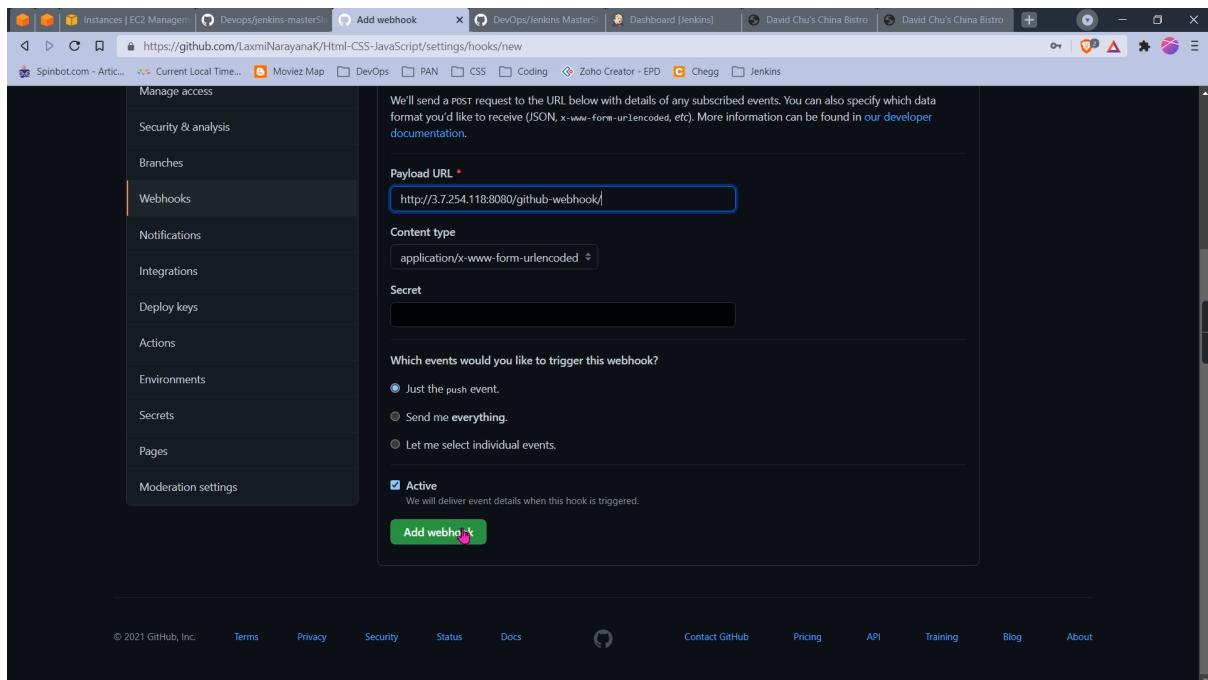
Save Apply

=>Under the Configure section of the CiCd pipeline,make Pipeline flow -> select initial job -> test.

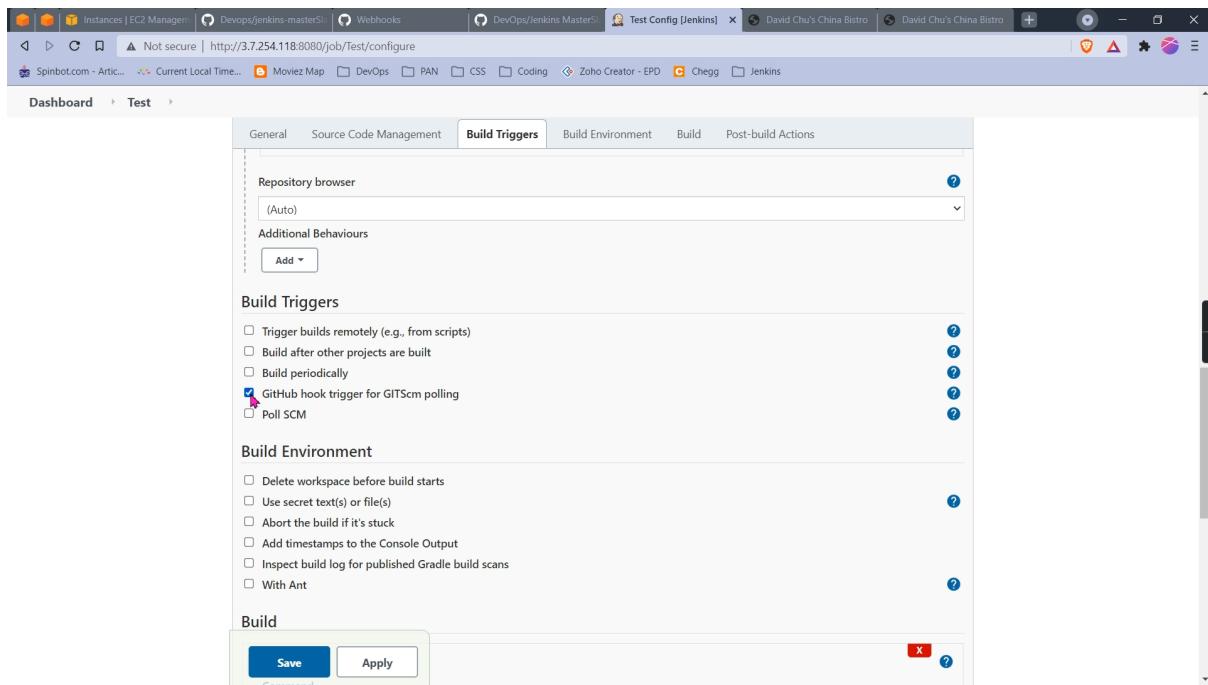
=>Now click on Run to run the respective jobs.



6. Using Github-Webhook to reflect changes made to github projects in the test and production servers.



=>Goto jenkins dashboard -> test -> configure -> build triggers. And enable the github hook trigger for GIT-Scm polling option and click save.



=>Headover to your github webapp project and goto settings -> Webhooks -> Add webhook. Now copy paste the ip address of your jenkins server(master node) under the payload url section and click add webhook.

=>NOTE : Verify that you get a tick mark under the webhook section for the url that you copy pasted.

=>Now lets trigger a build by committing port changes to the github webapp project and see the changes made to the test and production server.

COMMANDS

```
git clone <github url>
cd <projectfolder>
ls
nano index.html // make changes to this file
git add .
git commit -m "message"
git push origin master
```

```
es: ubuntu@ip-172-31-12-220:~/Html-CSS-JavaScript/Coursera_Assignment-5
36 updates can be applied immediately.
22 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Nov 18 15:38:42 2021 from 157.47.30.87
ubuntu@ip-172-31-12-220:~$ ls
ubuntu@ip-172-31-12-220:~$ git clone https://github.com/LaxmiNarayanaK/Html-CSS-JavaScript.git
Cloning into 'Html-CSS-JavaScript'...
remote: Enumerating objects: 358, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 358 (delta 18), reused 3 (delta 1), pack-reused 319
Receiving objects: 100% (358/358), 21.51 MiB | 19.36 MiB/s, done.
Resolving deltas: 100% (56/56), done.
ubuntu@ip-172-31-12-220:~$ ls
Html-CSS-JavaScript
ubuntu@ip-172-31-12-220:~$ cd Html-CSS-JavaScript
ubuntu@ip-172-31-12-220:~/Html-CSS-JavaScript$ ls
Coursera_Assignment-2 Coursera_Assignment-4 Dockerfile azure-pipelines.yml
Coursera_Assignment-3 Coursera_Assignment-5 README.md docker-compose
ubuntu@ip-172-31-12-220:~/Html-CSS-JavaScript$ cd Coursera_Assignment-5
ubuntu@ip-172-31-12-220:~/Html-CSS-JavaScript/Coursera_Assignment-5$ ls
css fonts images index.html js readme.md snippets
ubuntu@ip-172-31-12-220:~/Html-CSS-JavaScript/Coursera_Assignment-5$ -
```

```
index.html
Modified
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>IIIT DHARWAD</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/styles.css">
    <link href="https://fonts.googleapis.com/css?family=Oxygen:400,300,700" rel="stylesheet" type="text/css">
    <link href="https://fonts.googleapis.com/css?family=Lora" rel="stylesheet" type="text/css">
  </head>
  <body>
    <nav id="header-nav" class="navbar navbar-default">
      <div class="container">
        <div class="navbar-header">
          <a href="index.html" class="pull-left visible-md visible-lg">
            <div id="logo-img" alt="Logo image"></div>
          </a>
          <div class="navbar-brand">
            <a href="index.html">TEAM 12 RESTAURANT</a>
            <p>
              
              Kosher Certified
            </p>
          </div>
        <button id="navberToggle" type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#collapsable-nav" aria-expanded="false">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
      </div>
      <div id="collapsable-nav" class="collapse navbar-collapse">
        <ul id="nav-list" class="nav navbar-nav navbar-right">
          <li id="navHomeButton" class="visible-xs active">
            <a href="index.html">
              <span class="glyphicon glyphicon-home"></span> Home
            </a>
          </li>
          <li id="navMenuButton">
            <a href="#" onclick="$dc.loadMenuCategories();">
              <span class="glyphicon glyphicon-cutlery"></span> Menu
            </a>
          </li>
        </ul>
      </div>
    </nav>
    <div>
```

```

ubuntu@ip-172-31-12-220:~/Html-CSS-JavaScript/Coursera_Assignment-5
ubuntu@ip-172-31-12-220: $ ls
ubuntu@ip-172-31-12-220: $ git clone https://github.com/LaxmiNarayanaK/Html-CSS-JavaScript.git
Cloning into 'Html-CSS-JavaScript'...
remote: Enumerating objects: 358, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (37/37), done.
remote: Total 358 (delta 0), reused 3 (delta 1), pack-reused 319
Receiving objects: 100% (358/358), 21.51 MiB | 19.36 MiB/s, done.
Resolving deltas: 100% (56/56), done.
ubuntu@ip-172-31-12-220: $ ls
Html-CSS-JavaScript
ubuntu@ip-172-31-12-220: $ cd Html-CSS-JavaScript
-bash: cd: Html-CSS-JavaScript: No such file or directory
ubuntu@ip-172-31-12-220: $ cd Html-CSS-JavaScript
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript$ ls
Coursera_Assignment_4 Coursera_Assignment_4 Dockerfile azure-pipelines.yml
Coursera_Assignment_3 Coursera_Assignment_3 README.md docker-compose
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript$ cd Coursera_Assignment_5
-bash: cd: Coursera_Assignment_5: No such file or directory
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript$ cd Coursera_Assignment_5
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript/Coursera_Assignment_5$ ls
css fonts images index.html js readme.md snippets
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript/Coursera_Assignment_5$ nano index.html
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript/Coursera_Assignment_5$ git add .
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript/Coursera_Assignment_5$ git commit -m "Title Changed"
[master 490d7f8] Title changed
Committer: Ubuntu <ubuntu@ip-172-31-12-220.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
git config --global --edit
After doing this, you may fix the identity used for this commit with:
git commit --amend --reset-author
1 file changed, 2 insertions(+), 2 deletions(-)
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript/Coursera_Assignment_5$ git push origin master
Username for 'https://github.com': LaxmiNarayanaK
Password for 'https://LaxmiNarayanaK@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 409 bytes | 409.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/LaxmiNarayanaK/Html-CSS-JavaScript.git
  958006f..490d7f8  master -> master
ubuntu@ip-172-31-12-220: ~/Html-CSS-JavaScript/Coursera_Assignment_5$
```

Build Pipeline: CICD

Run History Configure Add Step Delete Manage

Pipeline #4

#4 Test

Nov 18, 2021 4:24:05 PM
1.6 sec and counting

Prod

N/A N/A

REST API Jenkins 2.303.3

Jenkins

Dashboard > CICD >

Build Pipeline: CICD

Pipeline #4 #5 Prod

#4 Test

Nov 18, 2021 4:24:05 PM 45 sec

#5 Prod

Nov 18, 2021 4:24:15 PM 47 sec

REST API Jenkins 2.303.3

Testing Server

IIIT DHARWAD

David Chu's China Bistro

TEAM 12 RESTAURANT

KOSHER CERTIFIED

410-602-5008 * We Deliver

Waiting for davids-restaurant.herokuapp.com...

Production Server

