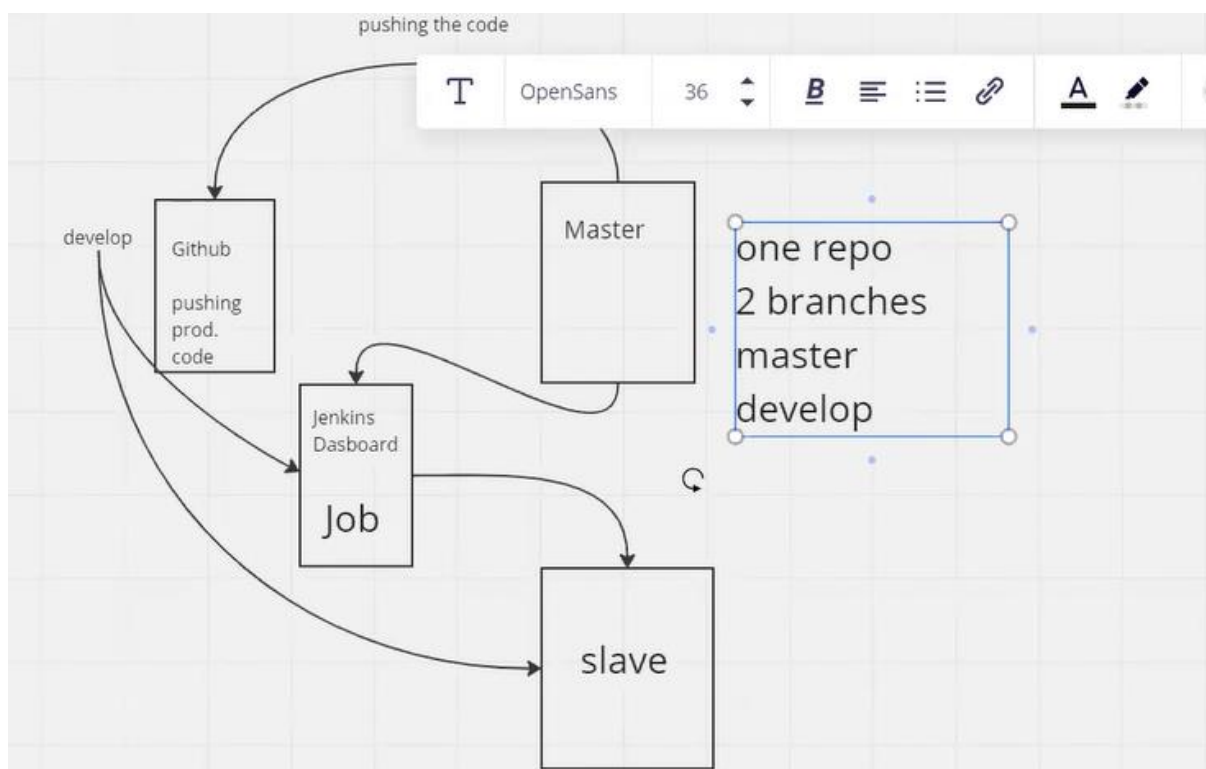


Module-8: Jenkins Assignment - 1

You have been asked to:

- Trigger a pipeline using Git when push on Develop branch
- Pipeline should pull git content to a folder



Launch Instances

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive) All states

running Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	Jenkins-Master	i-0483966327c6df06c	Running	t2.micro	Initializing	View alarms +	us-east-1d
<input type="checkbox"/>	Jenkins-Slave	i-0e411f3e540f5272a	Running	t2.micro	Initializing	View alarms +	us-east-1d

i-0483966327c6df06c (Jenkins-Master)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary Info

Instance ID i-0483966327c6df06c (Jenkins-Master)	Public IPv4 address 54.197.64.173 open address	Private IPv4 addresses 172.31.80.163
---	---	---

```
C:\WINDOWS\system32\cmd. X Command Prompt X + v
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-197-64-173.compute-1.amazonaws.com
```

```
C:\WINDOWS\system32\cmd. X Command Prompt X + v
ubuntu@ip-172-31-80-163:~$ sudo hostnamectl set-hostname Jenkins-Master
ubuntu@ip-172-31-80-163:~$ exit
logout
Connection to ec2-54-197-64-173.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-197-64-173.compute-1.amazonaws.com
```

```
ubuntu@Jenkins-Master: ~ X Command Prompt X + v
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-166-161-27.compute-1.amazonaws.com
```

Change Hostname

```
ubuntu@Jenkins-Master: ~ X Command Prompt X + v
ubuntu@ip-172-31-95-230:~$ sudo hostnamectl set-hostname Jenkins-Slave
ubuntu@ip-172-31-95-230:~$ exit
logout
Connection to ec2-54-166-161-27.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-166-161-27.compute-1.amazonaws.com
```

Install Jenkins

```
ubuntu@Jenkins-Master: ~ X ubuntu@Jenkins-Slave: ~ X +
ubuntu@Jenkins-Master:~$ nano jenkins_install.sh
```

```
GNU nano 6.2 jenkins_install.sh *
sudo apt update -y
sudo apt install openjdk-17-jre -y
java --version

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install jenkins -y
```

Execute

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Master:~$ nano jenkins_install.sh  
ubuntu@Jenkins-Master:~$ bash jenkins_install.sh  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]  
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]  
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]  
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]  
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]  
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]  
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1734 kB]
```

Installed

```
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
ubuntu@Jenkins-Master:~$ |
```

Jenkins --version

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Master:~$ jenkins --version  
2.452.2  
ubuntu@Jenkins-Master:~$
```

Update package in slave

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave:~$ sudo apt update -y
```

Install java in slave

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave:~$ sudo apt install openjdk-17-jre -y
```

After that check the version whether its installed

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave: ~$ java --version  
openjdk 17.0.11 2024-04-16  
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-122.04.1)  
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-122.04.1, mixed mode, sharing)  
ubuntu@Jenkins-Slave:~$
```

Copy master public IP and paste it on to browser and :8080

<input checked="" type="checkbox"/>	Jenkins-Master	i-0483966327c6df06c	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d
<input type="checkbox"/>	Jenkins-Slave	i-0e411f3e540f5272a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d

i-0483966327c6df06c (Jenkins-Master)	
Instance summary Info	Public IPv4 address copied
Instance ID i-0483966327c6df06c (Jenkins-Master)	Private IPv4 addresses 54.197.64.173 open address 172.31.80.163

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

Continue

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave: ~  
ubuntu@Jenkins-Master:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
2aaae6901b724cbdb2be0a74e2782a03  
ubuntu@Jenkins-Master:~$
```

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

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

Administrator password

.....

Continue

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.

Create First Admin User

Username

salmanfarsi

Password

.....

Confirm password

.....

Full name

SalmanFarsi

E-mail address

shahsalmanfarsi80@gmail.com

Instance Configuration

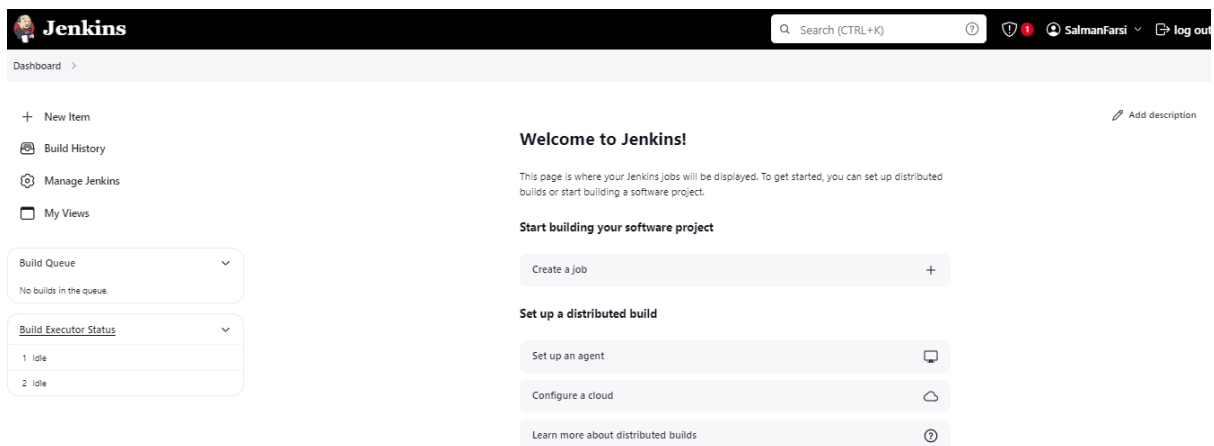
Jenkins URL:

<http://54.197.64.173:8080/>

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.

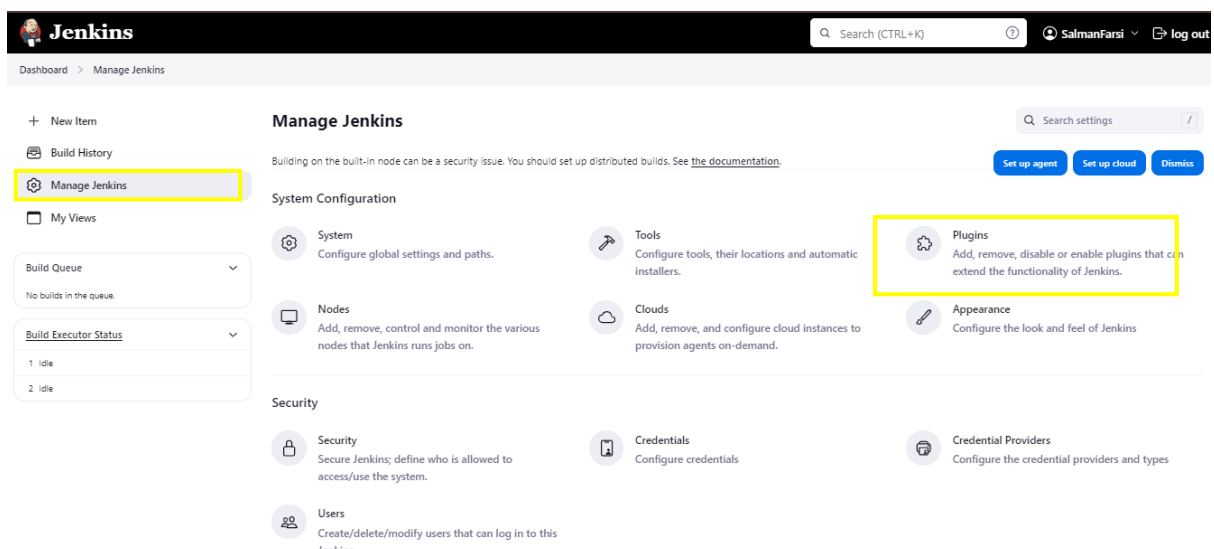
The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins dashboard



The screenshot shows the Jenkins dashboard. The top navigation bar includes the Jenkins logo, a search bar, and the user name 'SalmanFarsi'. The left sidebar contains links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The main content area has a 'Welcome to Jenkins!' message and a 'Start building your software project' section with buttons for 'Create a job', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. There are also status boxes for 'Build Queue' and 'Build Executor Status'.

Go to Manage Jenkins



The screenshot shows the 'Manage Jenkins' page. The left sidebar is the same as the dashboard. The main content area is titled 'Manage Jenkins' and contains a warning about security. Below the warning are three buttons: 'Set up agent', 'Set up cloud', and 'Dismiss'. The page is divided into sections: 'System Configuration' (System, Tools, Plugins, Nodes, Clouds, Appearance) and 'Security' (Security, Credentials, Credential Providers, Users). The 'Plugins' section is highlighted with a yellow box.

Go to Plugins and install SSH Agent

The screenshot shows the Jenkins 'Plugins' page. The search bar at the top contains 'ssh agent'. The left sidebar has 'Available plugins' highlighted. The main table lists installed plugins with their status. At the bottom, there are links to go back to the top page and a checkbox to restart Jenkins.

Install	Name	Released
<input checked="" type="checkbox"/>	SSH Agent 367.vf9076cd4ee21 This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.	2 mo 10 days ago

Plugin	Status
Pipeline Graph Analysis	Success
Metrics	Success
Pipeline Graph View	Success
Git	Success
EDDSA API	Success
Trilead API	Success
SSH Build Agents	Success
Matrix Authorization Strategy	Success
PAM Authentication	Success
LDAP	Success
Email Extension	Success
Mailer	Success
Theme Manager	Success
Dark Theme	Success
Loading plugin extensions	Success
SSH Agent	Success
Loading plugin extensions	Success

→ [Go back to the top page](#)
(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

Set up agent

The screenshot shows the Jenkins 'Welcome to Jenkins!' page. The 'Set up a distributed build' section is highlighted, showing options to 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

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

Create a job

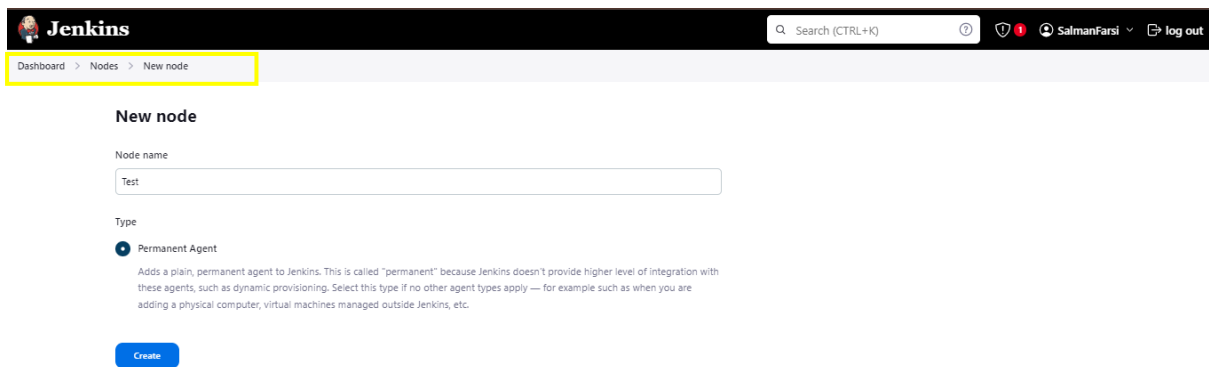
Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

Name of Node



The screenshot shows the Jenkins 'New node' configuration page. The 'Node name' field is set to 'Test'. The 'Type' is set to 'Permanent Agent'. A description explains that this type is for plain, permanent agents. A 'Create' button is at the bottom.

Jenkins Search (CTRL+K) SalmanFaris log out

Dashboard > Nodes > New node

New node

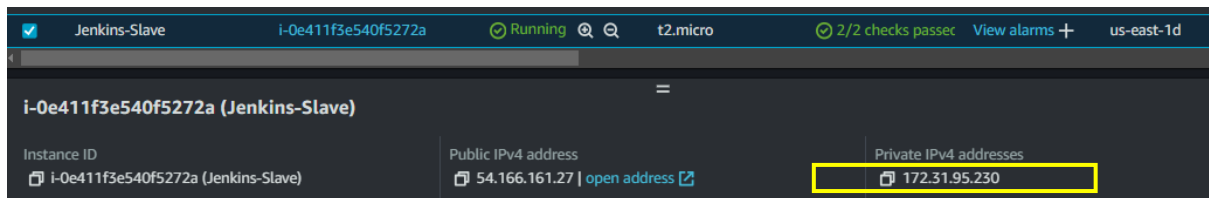
Node name
Test

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.

Create

Copy Private Ip of slave node



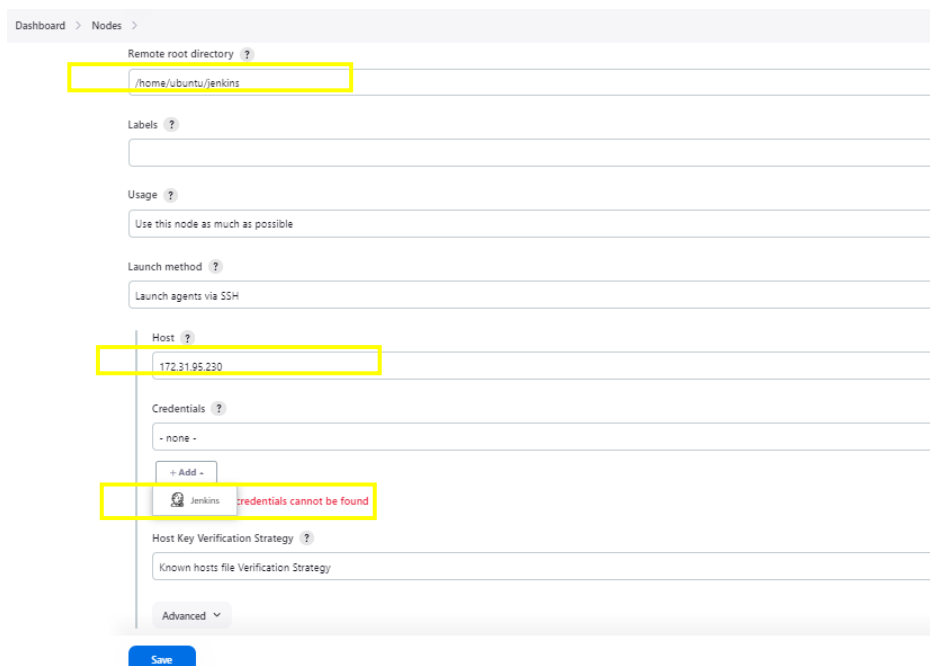
The screenshot shows the AWS Management Console details for an instance named 'Jenkins-Slave'. The instance ID is 'i-0e411f3e540f5272a'. The public IPv4 address is '54.166.161.27'. The private IPv4 address is '172.31.95.230', which is highlighted with a yellow box.

Jenkins-Slave i-0e411f3e540f5272a Running t2.micro 2/2 checks passed View alarms us-east-1d

i-0e411f3e540f5272a (Jenkins-Slave)

Instance ID i-0e411f3e540f5272a (Jenkins-Slave)	Public IPv4 address 54.166.161.27 open address	Private IPv4 addresses 172.31.95.230
--	---	---

Give Remote Location and Launch Method via ssh and Host add credentials



The screenshot shows the Jenkins 'New node' configuration page with the following settings: Remote root directory is '/home/ubuntu/jenkins'; Labels is empty; Usage is 'Use this node as much as possible'; Launch method is 'Launch agents via SSH'; Host is '172.31.95.230'; Credentials is 'Jenkins' (with a red error message 'Credentials cannot be found'); Host Key Verification Strategy is 'Known hosts file Verification Strategy'. A 'Save' button is at the bottom.

Dashboard > Nodes >

Remote root directory ?
/home/ubuntu/jenkins

Labels ?

Usage ?
Use this node as much as possible

Launch method ?
Launch agents via SSH

Host ?
172.31.95.230

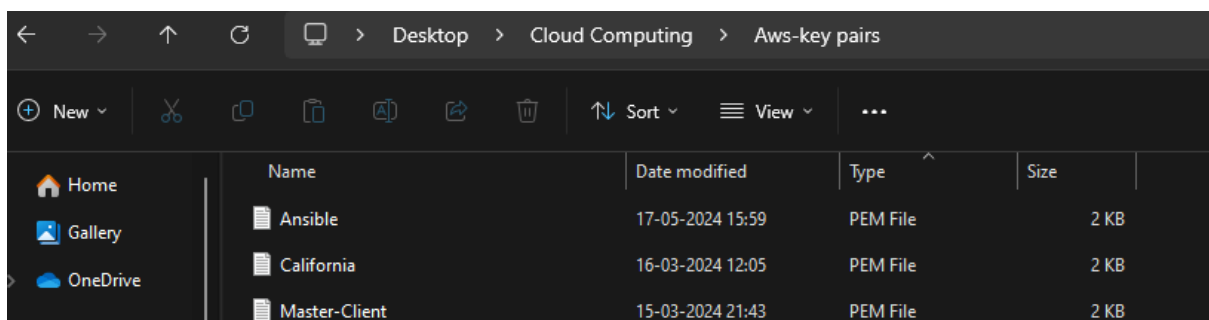
Credentials ?
- none -
+ Add -
Jenkins Credentials cannot be found

Host Key Verification Strategy ?
Known hosts file Verification Strategy

Advanced ▾

Save

Go to Local Path and open attached key pair of your instances and copy it.



The screenshot shows a Windows File Explorer window with the path 'Desktop > Cloud Computing > Aws-key pairs'. It displays a list of files: 'Ansible', 'California', and 'Master-Client', all of which are PEM files, each 2 KB in size.

	Name	Date modified	Type	Size
	Ansible	17-05-2024 15:59	PEM File	2 KB
	California	16-03-2024 12:05	PEM File	2 KB
	Master-Client	15-03-2024 21:43	PEM File	2 KB

Paste key pair and save/add it

Jenkins Credentials Provider: Jenkins

SSH Username with private key

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

ID ?
salman_cred

Description ?

Username
ubuntu

☐ Treat username as secret ?

Private Key
☒ Enter directly

Key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAIRLbt+42ip7pwVkuKPLkHh773u8Y8u+F2mK0zW5/Vh4+pu4
bndocxkx+100/vBROGZ1g1gHETEB2zJngc3B3uXgpy/VLP0dPJA1p3W4h8ouapYmK
-----END RSA PRIVATE KEY-----
```

Select non availability

Dashboard > Nodes >

Host ?
172.31.95.230

Credentials ?
ubuntu

+ Add ▾

Host Key Verification Strategy ?
Non verifying Verification Strategy

Advanced ▾

Availability ?
Keep this agent online as much as possible

Node Properties

☐ Disable deferred wipeout on this node ?

☐ Disk Space Monitoring Thresholds

☐ Environment variables

☐ Tool Locations

Save

Node connected

Jenkins

Search (CTRL+K)

SalmanFarsi

log out

Dashboard > Nodes >

Nodes

Clouds

Build Queue

Build Executor Status

No builds in the queue.

Built-In Node

1 idle

Test

1 idle

Nodes

S

Name

Architecture

Clock Difference

Free Disk Space

Free Swap Space

Free Temp Space

Response Time

Built-In Node

Linux (amd64)In sync4.83 GiB0 B4.83 GiB0ms

Test

Linux (amd64)In sync5.35 GiB0 B5.35 GiB50ms

Data obtained

1 min 4 sec

1 min 4 sec

1 min 3 sec

1 min 3 sec

1 min 3 sec

1 min 3 sec

Icons: S M L

Legend

Create a job

Jenkins

Search (CTRL+K)

SalmanFarsi

log out

Dashboard >

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

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

Create a Job

Add description

Jenkins

Search (CTRL+K)

Dashboard >

Enter an item name

Job1

Required field

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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.

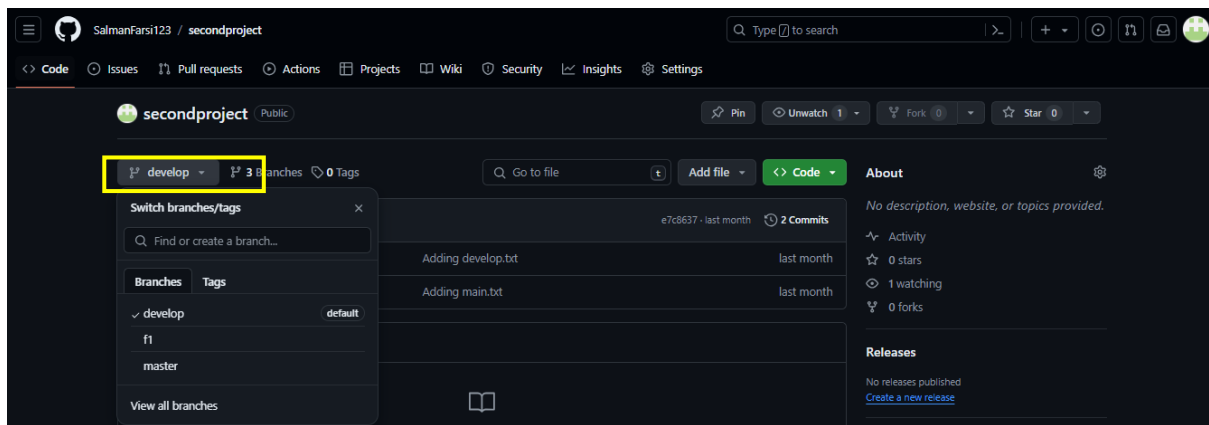
Folder

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.

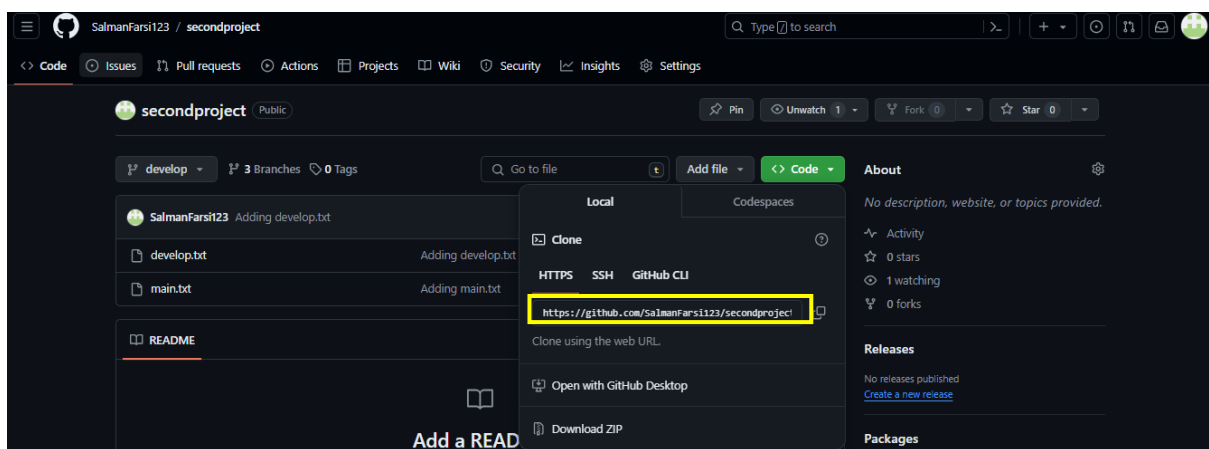
OK

branch Pipeline

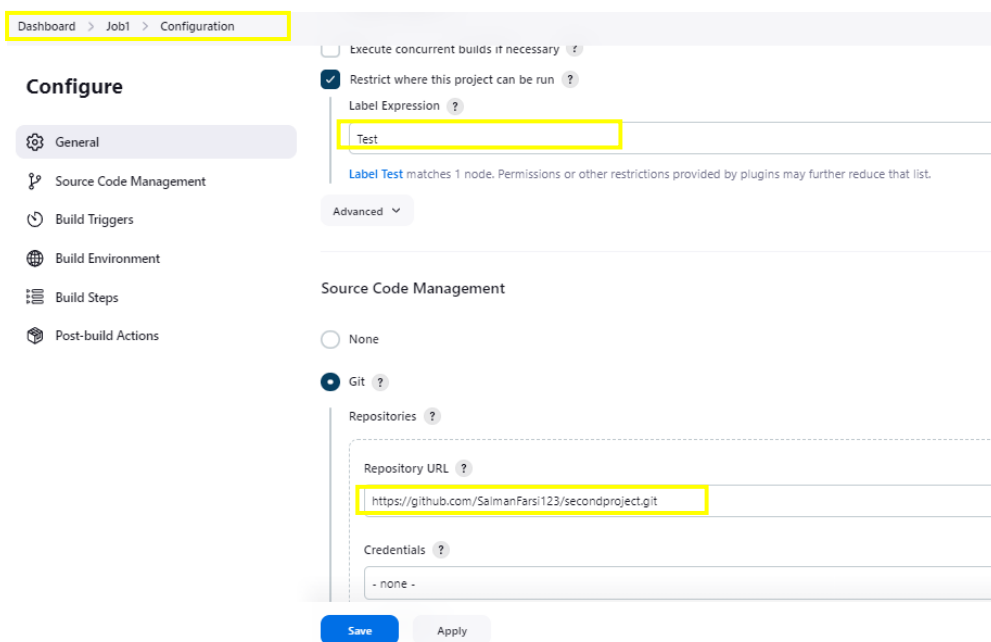
Develop branch



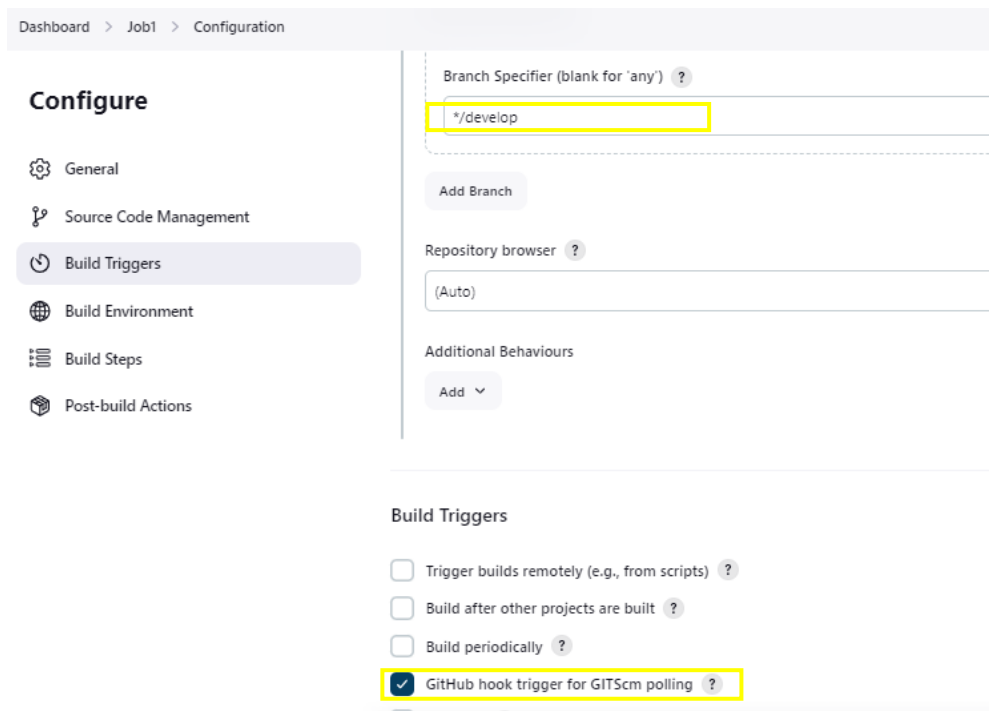
Copy the path



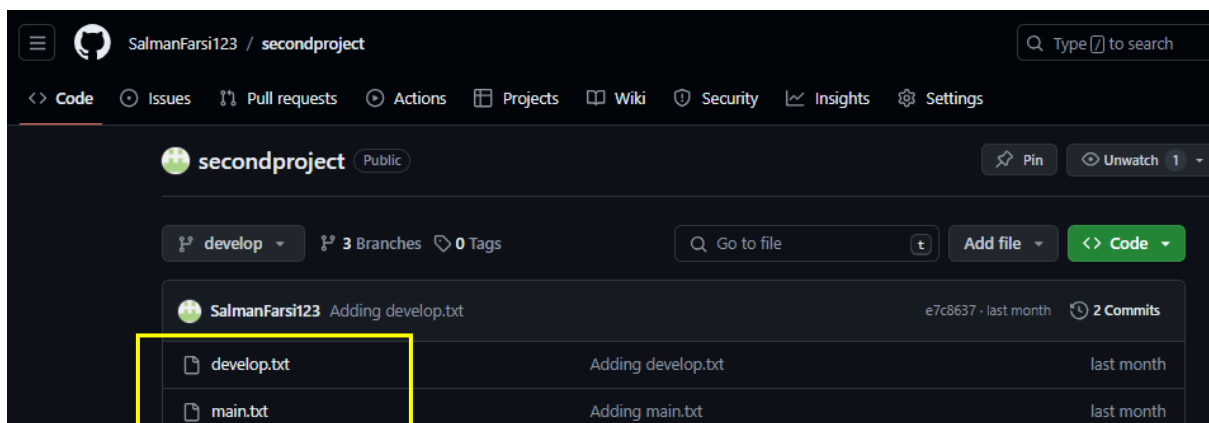
And Come to job1 config and select Test Node and paste the path



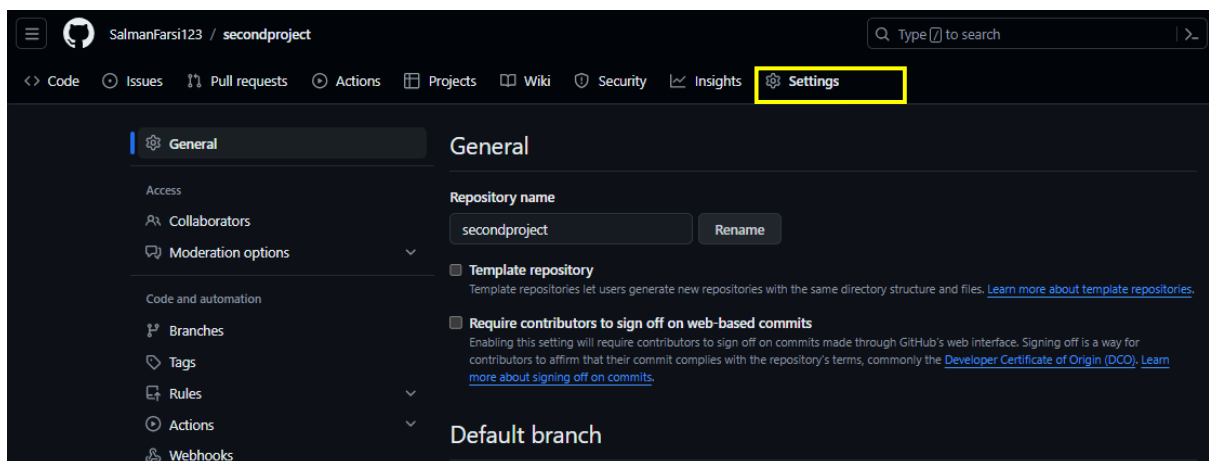
Branch is develop and github triggering



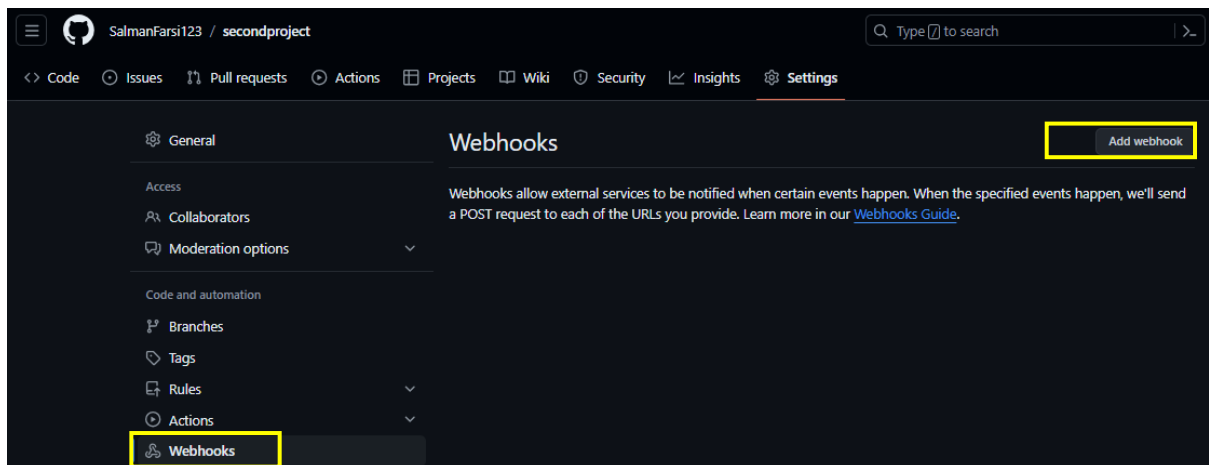
settings



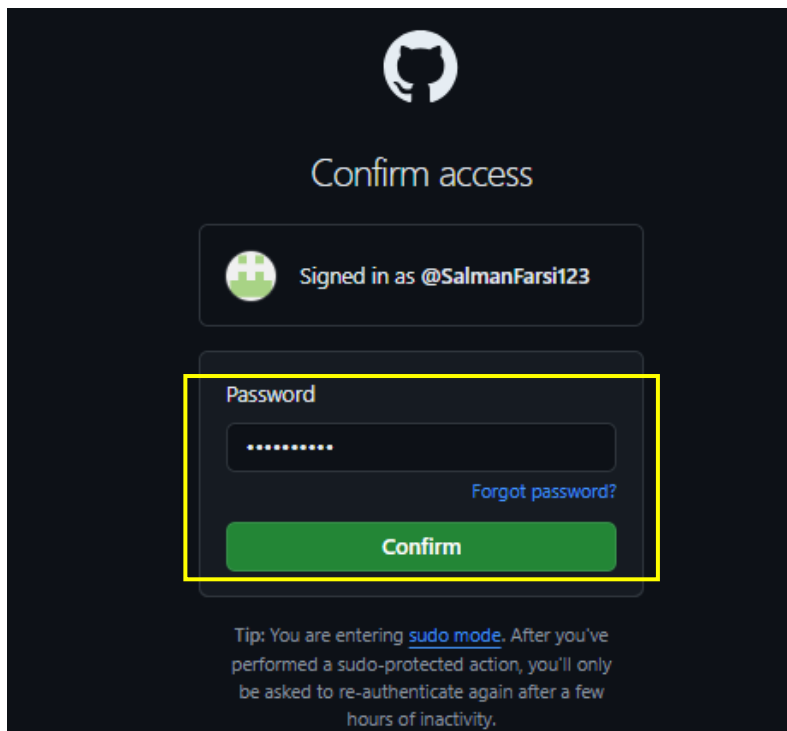
webhooks



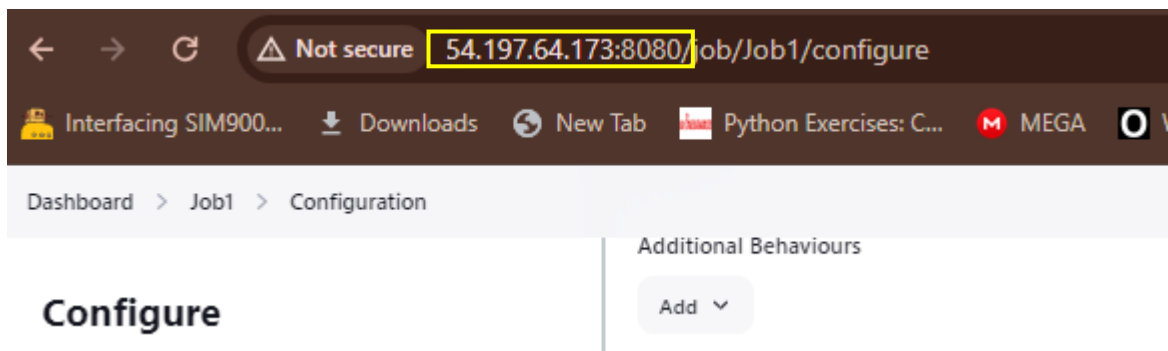
Add Webhooks



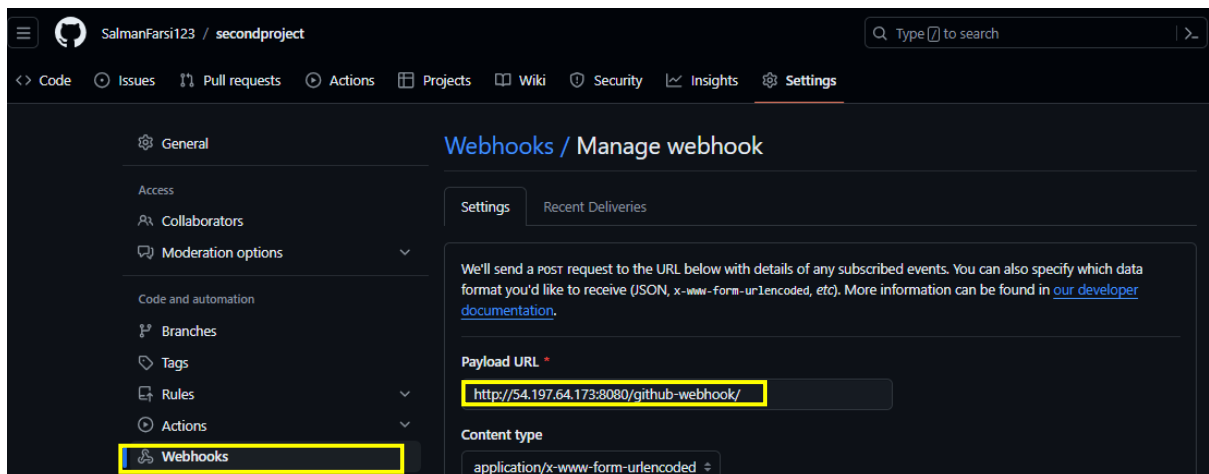
Give Github Credentials



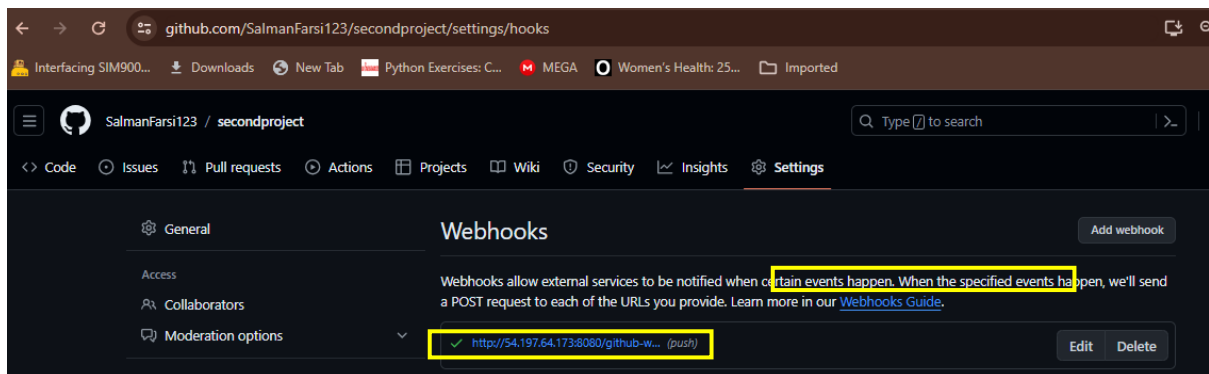
Copy the Jenkins public IP



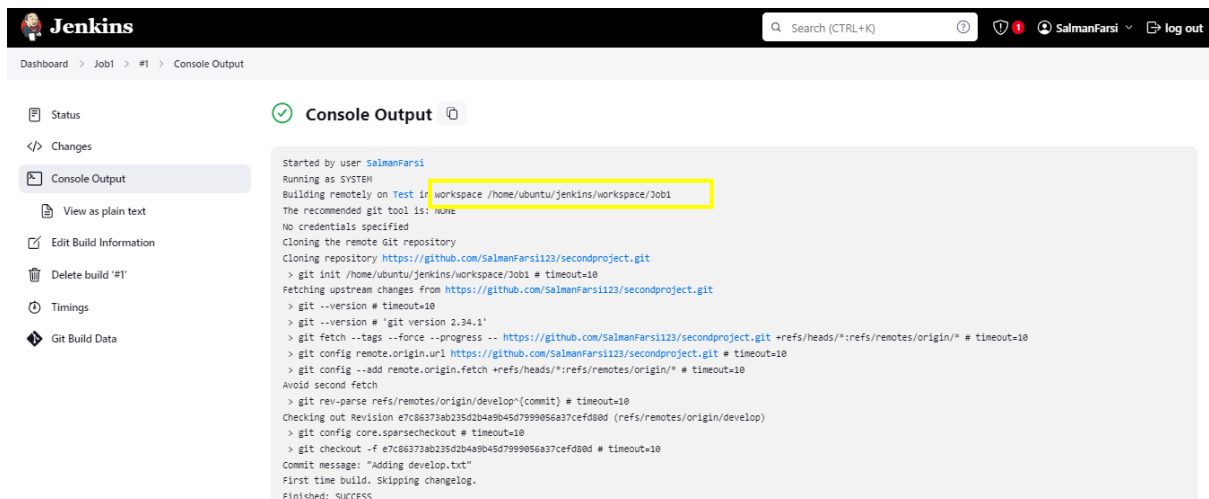
Paste and Give extension github-webhook



Refresh and its added successfully



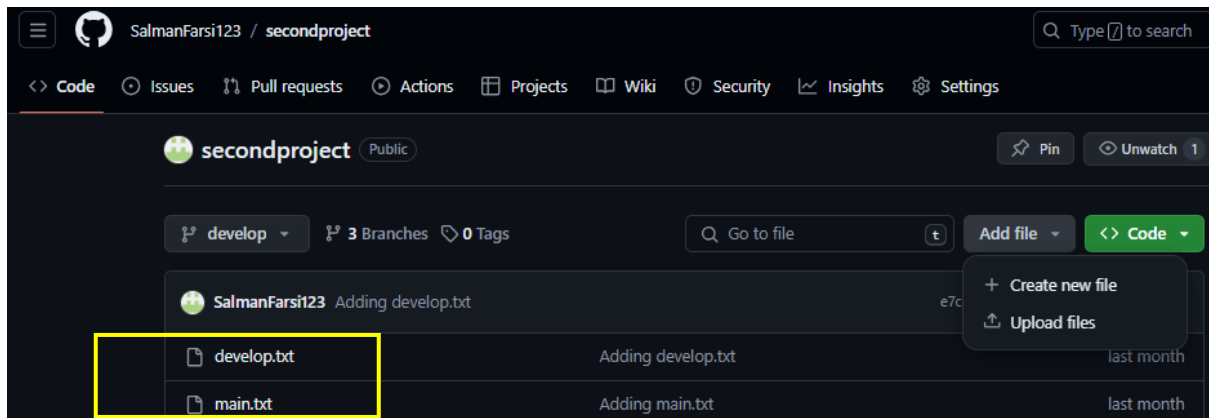
Now go to job1 and build now and the job ran successfully



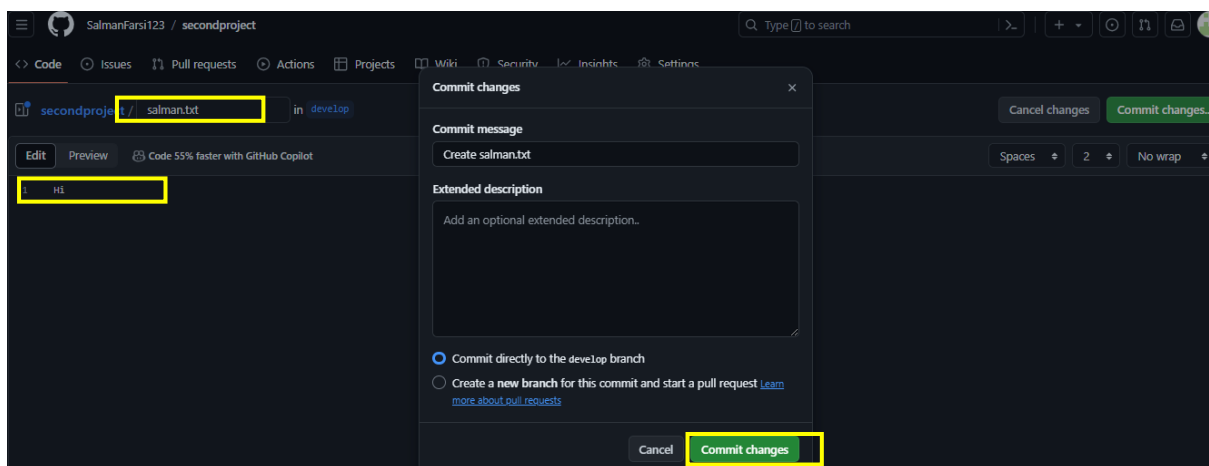
We can see the develop branch data into test node(slave server)

```
ubuntu@Jenkins-Master: ~  
ubuntu@Jenkins-Slave: ~/jen  
ubuntu@Jenkins-Slave:~$ cd /home/ubuntu/jenkins/workspace/Job1  
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$ ls  
develop.txt  main.txt  
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$
```

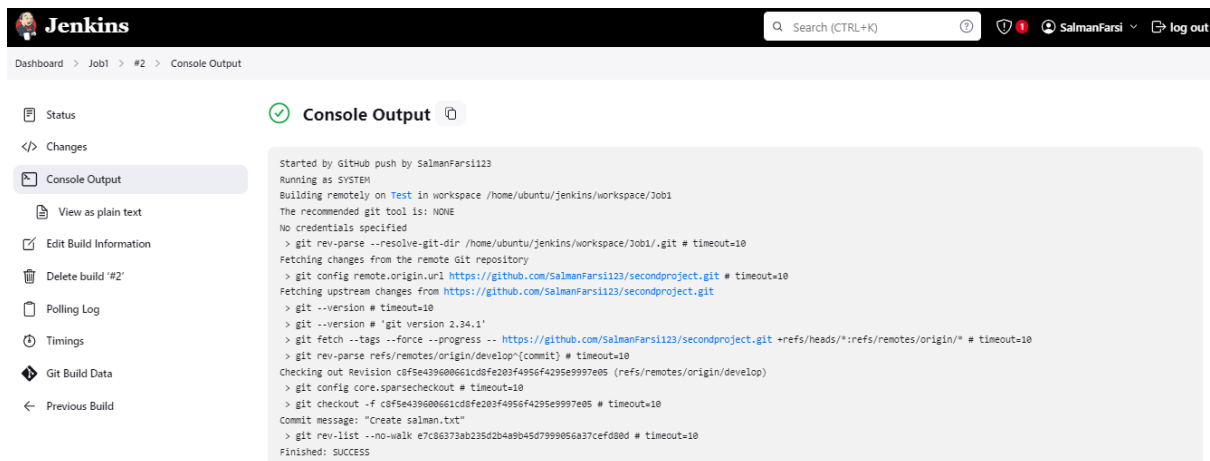
2 files is there in develop branch and its successfully pushed into slave server



Now creating salman.txt in develop branch and commit it.

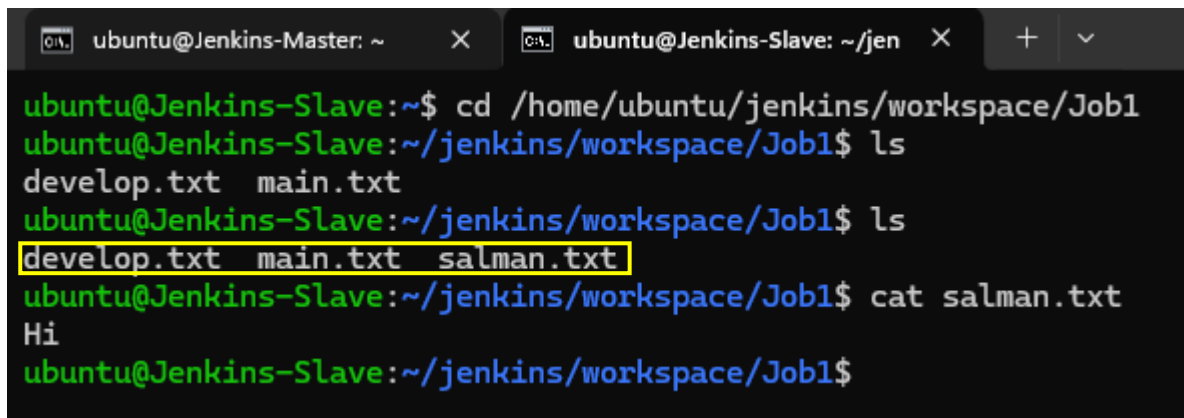


Now the Build Automatically ran and its successfully ran



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and a user profile for 'SalmanFarsi'. The breadcrumb trail is 'Dashboard > Job1 > #2 > Console Output'. On the left sidebar, the 'Console Output' tab is selected. The main area displays the build log, which starts with 'Started by GitHub push by SalmanFarsi123' and shows the execution of various git commands like 'git rev-parse', 'git fetch', and 'git checkout'. The build concludes with 'Finished: SUCCESS'.

I can the salman.txt file



The screenshot shows a terminal window with two tabs: 'ubuntu@Jenkins-Master: ~' and 'ubuntu@Jenkins-Slave: ~/jen'. The active tab is the Jenkins-Slave. The user navigates to the workspace directory and lists files. The output shows 'develop.txt' and 'main.txt'. Then, 'salman.txt' is added to the list. Finally, the content of 'salman.txt' is displayed as 'Hi'.

DevOps Certification Training

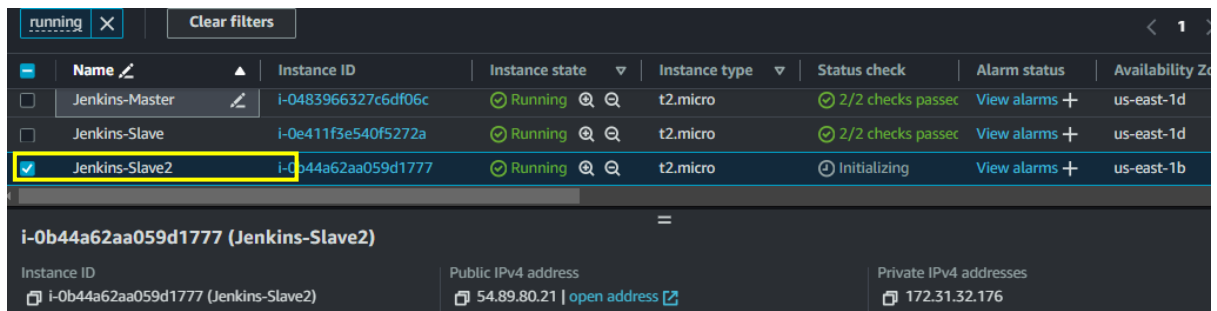


Module-8: Jenkins Assignment - 2

You have been asked to:

- Add 2 nodes to Jenkins master
- Create 2 jobs with the following jobs:
 - Push to test
 - Push to prod
- Once a push is made to test branch copy git files to test server
- Once a push is made to master branch copy git files to prod server

Launch another instances for assignment-2



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Jenkins-Master	i-0483966327c6df06c	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d
<input type="checkbox"/>	Jenkins-Slave	i-0e411f3e540f5272a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d
<input checked="" type="checkbox"/>	Jenkins-Slave2	i-0b44a62aa059d1777	Running	t2.micro	Initializing	View alarms +	us-east-1b

i-0b44a62aa059d1777 (Jenkins-Slave2)		
Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b44a62aa059d1777 (Jenkins-Slave2)	54.89.80.21 open address	172.31.32.176

```
ubuntu@Jenkins-Master: ~  
Microsoft Windows [Version 10.0.22631.3593]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\shaik>cd C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs  
C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-89-80-21.compute-1.amazonaws.com
```

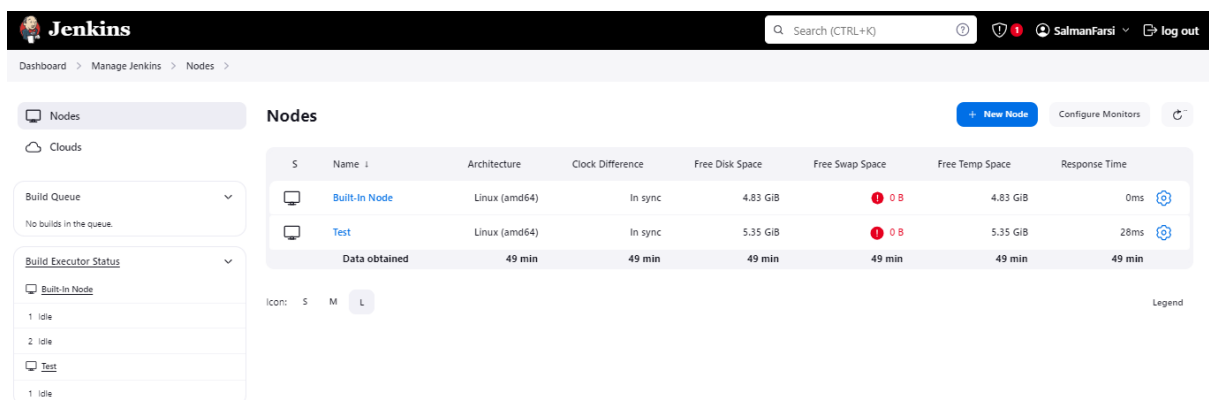
```
ubuntu@ip-172-31-32-176:~$ sudo hostnamectl set-hostname Jenkins-Slave2  
ubuntu@ip-172-31-32-176:~$ exit  
logout  
Connection to ec2-54-89-80-21.compute-1.amazonaws.com closed.  
C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-89-80-21.compute-1.amazonaws.com
```

```
ubuntu@Jenkins-Slave2:~$ sudo apt update -y
```

Install Java for Jenkins-slave

```
ubuntu@Jenkins-Slave2:~$ sudo apt install openjdk-17-jdk -y
```

Perform Task Before how we connected test server



Jenkins Search (CTRL+K) SalmanFarsi log out

Dashboard > Manage Jenkins > Nodes

Nodes + New Node Configure Monitors

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.83 GiB	0 B	4.83 GiB	0ms
	Test	Linux (amd64)	In sync	5.35 GiB	0 B	5.35 GiB	28ms
Data obtained		49 min	49 min	49 min	49 min	49 min	49 min

Icon: S M L Legend

Create node > Name > Copy from Existing Node

Dashboard > Manage Jenkins > Nodes > New node

New node

Node name

Prod

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

Test

Create

Copy the Private Ip of slave2

running x Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	Jenkins-Master	i-0483966327c6df06c	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d
<input type="checkbox"/>	Jenkins-Slave	i-0e411f3e540f5272a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d
<input checked="" type="checkbox"/>	Jenkins-Slave2	i-0b44a62aa059d1777	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b

i-0b44a62aa059d1777 (Jenkins-Slave2)

Details Status and alarms Monitoring Security Networking Storage Tags

▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 address copied
i-0b44a62aa059d1777 (Jenkins-Slave2)	54.89.80.21 open address	172.31.32.176

Give label name Prod, Paste and credentials everything same because same private key pair I used for all instances

Dashboard > Manage Jenkins > Nodes > Prod > Configure

Prod

Usage ?
Use this node as much as possible

Launch method ?
Launch agents via SSH

Host ?
172.31.32.176

Credentials ?
ubuntu
+ Add

Host Key Verification Strategy ?
Non verifying Verification Strategy

Advanced ▾ Edited

Save

connected

Jenkins

Search (CTRL+K)

Dashboard > Manage Jenkins > Nodes >

Nodes

Build Queue
No builds in the queue.

Build Executor Status

Built-In Node
1 Idle

Prod
1 Idle

Test
1 Idle

Nodes

0.71 sec

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.58 GiB	0 B	4.58 GiB	0ms
	Prod	Linux (amd64)	In sync	5.27 GiB	0 B	5.27 GiB	72ms
	Test	Linux (amd64)	In sync	5.34 GiB	0 B	5.34 GiB	32ms
Data obtained		0.71 sec	0.67 sec	0.67 sec	0.64 sec	0.67 sec	0.65 sec

Icon: S M L

Legend

Now Create Job2

Jenkins

Search (CTRL+K)

Dashboard >

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Add description

All +

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Job1	46 min #2	N/A	0.43 sec

Icon: S M L


Job2 and Copy from Job1


Dashboard > All >


Enter an item name


Job2


> Required field


**Freestyle project**
Classic, general purpose job type that checks out from up to one SCM, executes build steps serially, followed by post build steps like archiving artifacts and sending email notifications.

**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.

**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

**Copy from**
Job1

OK

Label Prod

Dashboard > Job2 > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

☒ Restrict where this project can be run ?

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

Advanced ▾

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?
https://github.com/SalmanFarsi123/secondproject.git

Credentials ?
- none -

Save Apply

And Change the Branch master

Dashboard > Job2 > Configuration

Configure

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

Branch Specifier (blank for 'any') ?

***/master**

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) ?
- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ **GitHub hook trigger for GITScm polling ?**
- ☐ Poll SCM ?

Save Apply

And Build Job2

Jenkins

Search (CTRL+K)

SalmanFarsi log out

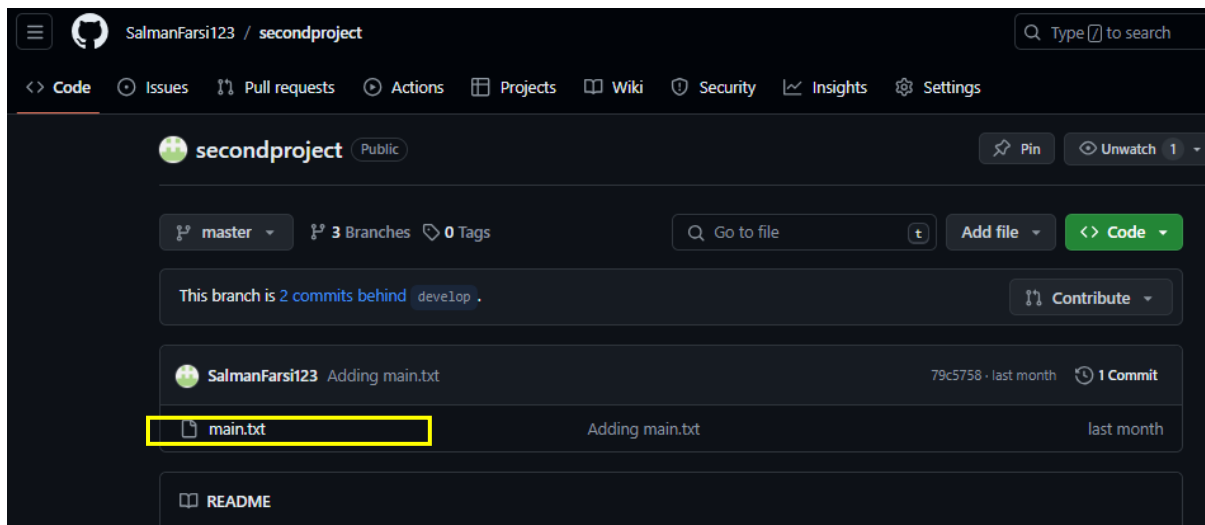
Dashboard > Job2 > #1 > Console Output

- Status
- Changes
- Console Output**
- View as plain text
- Edit Build Information
- Delete build '#1'
- Timings
- Git Build Data

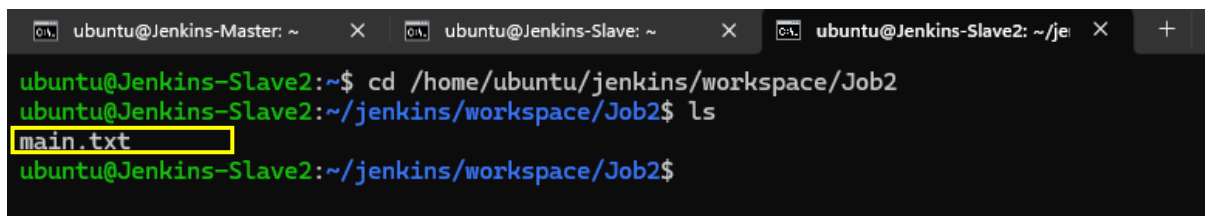
Console Output

```
Started by user SalmanFarsi
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/Job2
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/SalmanFarsi1123/secondproject.git
> git init /home/ubuntu/jenkins/workspace/Job2 # timeout=10
Fetching upstream changes from https://github.com/SalmanFarsi1123/secondproject.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/SalmanFarsi1123/secondproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/SalmanFarsi1123/secondproject.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 79c5758da0316516178ff677ed20725ec95df3d7 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 79c5758da0316516178ff677ed20725ec95df3d7 # timeout=10
Commit message: "Adding main.txt"
First time build. Skipping changelog.
Finished: SUCCESS
```

This is the Data of Master branch only having main.txt



We can see in 2nd slave server



DevOps Certification Training

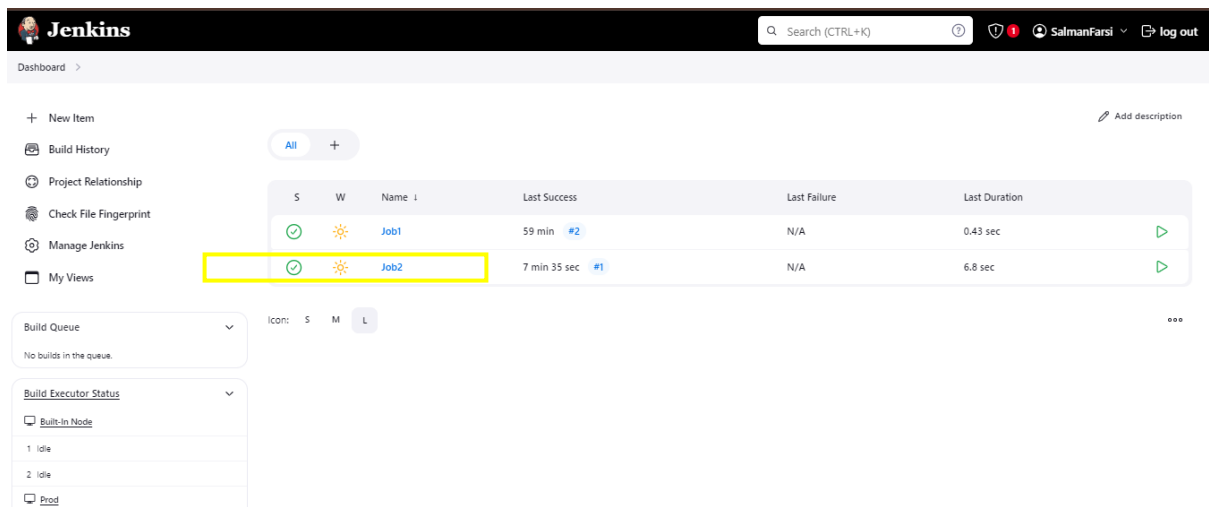


Module-8: Jenkins Assignment - 3

You have been asked to:

- Create a pipeline in jenkins
- Once push is made to "develop" branch in git, trigger job "test". This will copy git files to test node
- If test job is successful, then prod job should be triggered
- Prod jobs should copy files to prod node

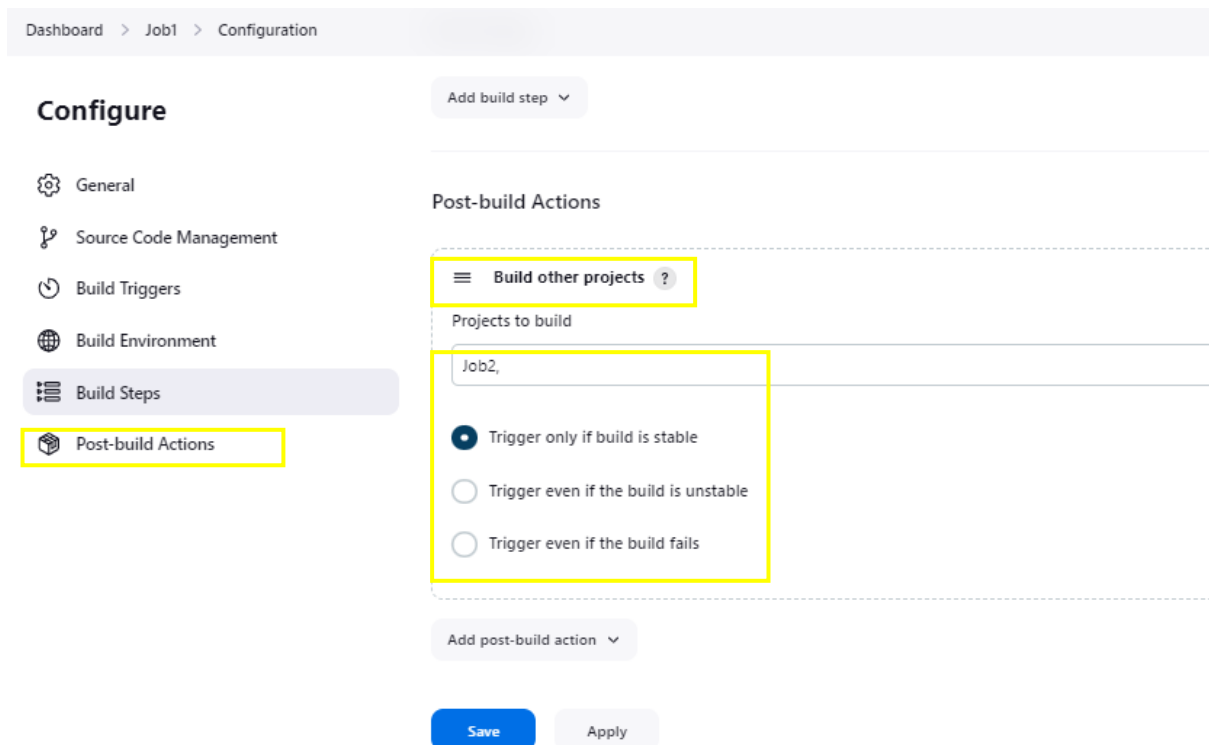
Now Click Job2



The screenshot shows the Jenkins Dashboard. At the top, there's a search bar and user information. The left sidebar contains navigation links: New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main area displays a table of build jobs. The 'Job2' row is highlighted with a yellow box. Below the table, there are sections for 'Build Queue' (showing no builds) and 'Build Executor Status' (showing three executors: 1 idle, 2 idle, and 1 Prod).

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Job1	59 min #2	N/A	0.43 sec
✓	☀	Job2	7 min 35 sec #1	N/A	6.8 sec

Go to Build Steps, here we are integrating job1 and job2



The screenshot shows the Jenkins Job Configuration page for 'Job1'. The left sidebar has a 'Post-build Actions' section highlighted with a yellow box. The main area shows the 'Post-build Actions' configuration. The 'Build other projects' option is selected and highlighted with a yellow box. Below it, the 'Projects to build' field contains 'Job2,'. The 'Trigger only if build is stable' option is selected with a radio button. At the bottom, there are 'Save' and 'Apply' buttons.

Configure

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

Post-build Actions

Build other projects ?

Projects to build

Job2,


☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

Save **Apply**

Click Build Now in job1 and it will automatically trigger to Job2



Jenkins

Dashboard > Job1 >

Status

Changes

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

Job1

Downstream Projects

Job2

Permalinks

- Last build (#2), 1 hr 3 min ago
- Last stable build (#2), 1 hr 3 min ago
- Last successful build (#2), 1 hr 3 min ago
- Last completed build (#2), 1 hr 3 min ago

Build History

trend

Filter...


#3

Jun 19, 2024, 12:28 PM

#2

Jun 19, 2024, 11:25 AM

#1



Jenkins

Search (CTRL+K)

SalmanFarsi

log out

Dashboard > Job1 > #3 > Console Output

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#3'

Timings

Git Build Data

Previous Build

Console Output

```
Started by user SalmanFarsi
Running as SYSTEM
Building remotely on Test in workspace /home/ubuntu/jenkins/workspace/Job1
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Job1/.git # timeout=10
Fetching changes from the remote git repository
> git config remote.origin.url https://github.com/SalmanFarsi1123/secondproject.git # timeout=10
Fetching upstream changes from https://github.com/SalmanFarsi1123/secondproject.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/SalmanFarsi1123/secondproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/develop^{commit} # timeout=10
Checking out Revision c8f5e439600661cd8fe203f4956f4295e9997e05 (refs/remotes/origin/develop)
> git config core.sparsecheckout # timeout=10
> git checkout -f c8f5e439600661cd8fe203f4956f4295e9997e05 # timeout=10
Commit message: "Create salman.txt"
> git rev-list --no-walk c8f5e439600661cd8fe203f4956f4295e9997e05 # timeout=10
Triggering a new build of Job2
Finished: SUCCESS
```


We can see its auto triggered and ran successfully

Dashboard > Job2 >

Status

Changes

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

Job2

Upstream Projects

Job1

Permalinks

Last build (#2), 19 sec ago

Last stable build (#2), 19 sec ago

Last successful build (#2), 19 sec ago

Last completed build (#2), 19 sec ago

Build History

trend

Filter...

#2

Jun 19, 2024, 12:28 PM

#1

Jun 19, 2024, 12:17 PM

Atom feed for all

Atom feed for failures

Jenkins

Search (CTRL+K)

SalmanFarsi

log out

Dashboard > Job2 > #2 > Console Output

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#2'

Timings

Git Build Data

Previous Build

Console Output

Started by upstream project "Job1" build number 3 originally caused by:
Started by user SalmanFarsi
Running as SYSTEM
Building remotely on Prod in workspace /home/ubuntu/jenkins/workspace/Job2
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/Job2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/SalmanFarsi123/secondproject.git # timeout=10
Fetching upstream changes from https://github.com/SalmanFarsi123/secondproject.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/SalmanFarsi123/secondproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 79c5758da0316516178ff677ed20725ec95df3d7 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 79c5758da0316516178ff677ed20725ec95df3d7 # timeout=10
Commit message: "Adding main.txt"
> git rev-list --no-walk 79c5758da0316516178ff677ed20725ec95df3d7 # timeout=10
Finished: SUCCESS