

JENKINS-DEMO

1. Launch Instances and Named as Jenkins-Master

The screenshot shows the AWS CloudWatch Instances console. At the top, there's a search bar with placeholder text 'Find Instance by attribute or tag (case-sensitive)' and a dropdown menu set to 'All states'. Below the search bar is a button labeled 'running' with a clear filters icon. To the right are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. A yellow 'Launch instances' button is highlighted. The main table lists one instance: 'Jenkins-Master' (Instance ID: i-02ae4cf2c64df9658), which is 'Running' (Status check: Initializing) and has an 't2.medium' instance type. It is located in the 'us-east-1e' availability zone. Below the table, the instance details are shown: 'i-02ae4cf2c64df9658 (Jenkins-Master)'. The 'Details' tab is selected, followed by 'Status and alarms', 'Monitoring', 'Security', 'Networking', 'Storage', and 'Tags'. Under 'Instance summary', it shows the Instance ID (i-02ae4cf2c64df9658), Public IPv4 address (54.165.12.194), and Private IPv4 addresses (172.31.61.40).

2. Copy SSH

This screenshot shows the 'Details' tab for the 'Jenkins-Master' instance. It includes a section titled '1. Open an SSH client.' with numbered steps: 1. Open an SSH client., 2. Locate your private key file. The key used to launch this instance is Master-Client.pem, 3. Run this command, if necessary, to ensure your key is not publicly viewable. chmod 400 "Master-Client.pem", and 4. Connect to your instance using its Public DNS: ec2-54-165-12-194.compute-1.amazonaws.com. Below this, there's an 'Example:' section with the command ssh -i "Master-Client.pem" ubuntu@ec2-54-165-12-194.compute-1.amazonaws.com.

3. Go to Private Key pair path and paste the ssh and connect

```
C:\WINDOWS\system32\cmd. 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@ec2-54-165-12-194.compute-1.amazonaws.com
```

4. change hostname Jenkins-Master and connect again.

```
C:\WINDOWS\system32\cmd. x + v
ubuntu@ip-172-31-61-40:~$ sudo hostnamectl set-hostname Jenkins-Master
ubuntu@ip-172-31-61-40:~$ exit
logout
Connection to ec2-54-165-12-194.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-54-165-12-194.compute-1.amazonaws.com
```

5. Install Jenkins

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

6. Paste Jenkins installation Stuff and Save it.

```
GNU nano 7.2                                         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

sudo systemctl daemon-reload
sudo systemctl enable jenkins
sudo systemctl start jenkins
sudo systemctl status jenkins
```

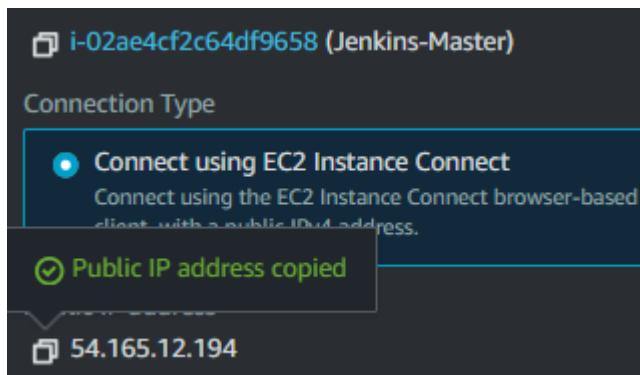
7. Give Permission and execute it.

```
ubuntu@Jenkins-Master:~$ nano jenkins_install.sh
ubuntu@Jenkins-Master:~$ chmod +x jenkins_install.sh
ubuntu@Jenkins-Master:~$ bash jenkins_install.sh
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 Packages [1401 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main Translation-en [513 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [88.0 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [24.5 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [6876 B]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [34.9 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [13.4 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
```

8. it will install it and control + c to come back

```
Jun 03 07:00:31 Jenkins-Master jenkins[3627]: cad15dd7f377468ea12882af594888f7
Jun 03 07:00:31 Jenkins-Master jenkins[3627]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPw
Jun 03 07:00:31 Jenkins-Master jenkins[3627]: ****
Jun 03 07:00:31 Jenkins-Master jenkins[3627]: ****
Jun 03 07:00:31 Jenkins-Master jenkins[3627]: ****
Jun 03 07:00:45 Jenkins-Master jenkins[3627]: 2024-06-03 07:00:45.284+0000 [id=49]           INFO      h.m.Down
Jun 03 07:00:45 Jenkins-Master jenkins[3627]: 2024-06-03 07:00:45.287+0000 [id=49]           INFO      hudson.util
Jun 03 07:00:45 Jenkins-Master jenkins[3627]: 2024-06-03 07:00:45.287+0000 [id=31]           INFO      jenkins.I
Jun 03 07:00:45 Jenkins-Master jenkins[3627]: 2024-06-03 07:00:45.304+0000 [id=24]           INFO      hudson.l
Jun 03 07:00:45 Jenkins-Master systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
lines 1-20/20 (END)
```

9. Copy Public-IP and paste in the browser



10. Asking Administrator Password go to path and copy the password

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

11. Copy it.

```
ubuntu@Jenkins-Master: ~      x + | v
ubuntu@Jenkins-Master:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
cad15dd7f377468ea12882af594888f7
ubuntu@Jenkins-Master:~$
```

12. Paste it.

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

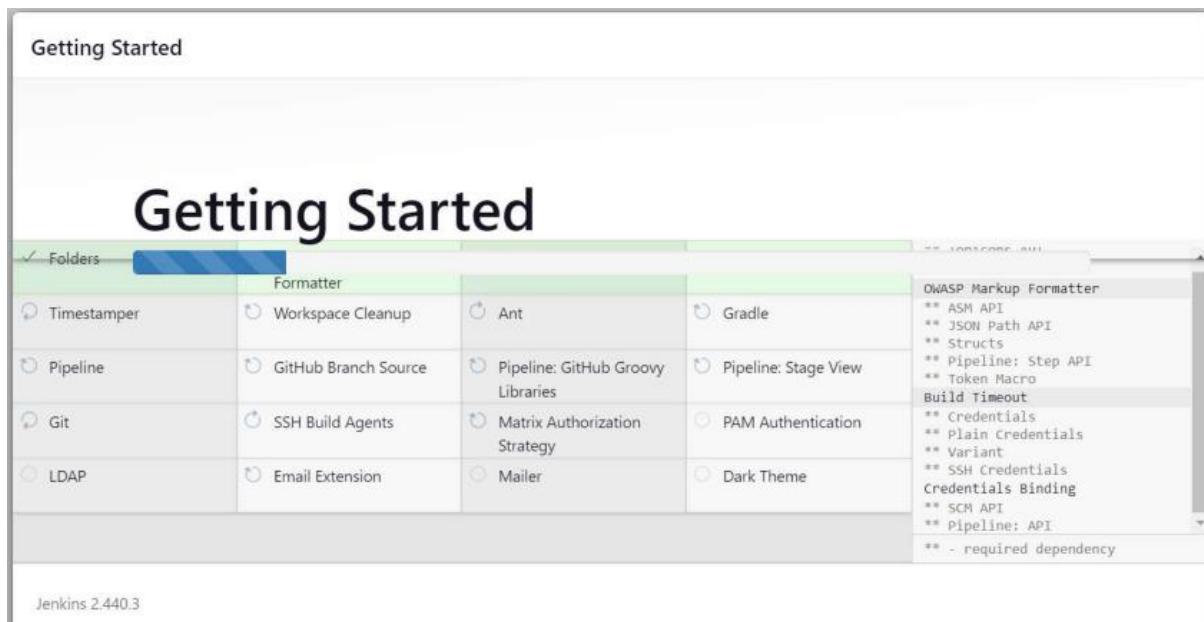
.....

Continue

13. Install Suggested plugins

The screenshot shows the Jenkins 'Customize Jenkins' setup screen. At the top, there's a 'Getting Started' header with a close button (X). Below it is a large title 'Customize Jenkins'. A sub-instruction says 'Plugins extend Jenkins with additional features to support many different needs.' There are two main options: 'Install suggested plugins' (selected) and 'Select plugins to install'. The 'Install suggested plugins' section includes a sub-instruction: 'Install plugins the Jenkins community finds most useful.' The 'Select plugins to install' section includes a sub-instruction: 'Select and install plugins most suitable for your needs.' At the bottom left, it says 'Jenkins 2.440.3'.

14. It will Install



15. Create your username and password

The screenshot shows the 'Create First Admin User' form. It consists of several input fields:

- Username: 'salmanfarsi'
- Password: '.....'
- Confirm password: '.....'
- Full name: 'SalmanFarsi'
- E-mail address: 'shaiksalmanfarsi80@gmail.com'

At the bottom of the form, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'.

16. save and Continue

Getting Started

Instance Configuration

Jenkins URL:

http://54.165.12.194:8080/

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.

17. Start Using Jenkins

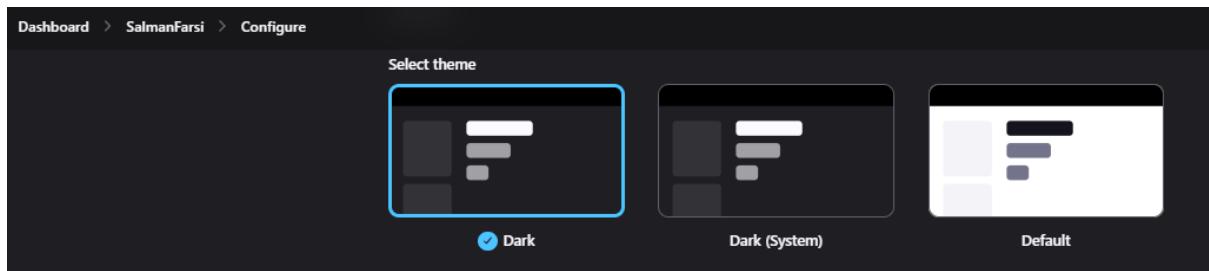
Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

18. I Selected Dark Window



19. Dashboard

The screenshot shows the Jenkins dashboard with a dark theme. On the left, there's a sidebar with links like 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below the sidebar are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 idle). The main area has a 'Welcome to Jenkins!' message: '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.' It includes a 'Create a job' button and links for 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

20. Click on Manage Jenkins > Plugins

The screenshot shows the 'Manage Jenkins' page. In the sidebar, 'Manage Jenkins' is highlighted. The main area has sections for 'System Configuration' (System, Tools, Nodes, Clouds), 'Appearance', and 'Plugins'. The 'Plugins' section is highlighted with a yellow box and contains a sub-section for 'Available plugins'. A search bar at the top right is also highlighted.

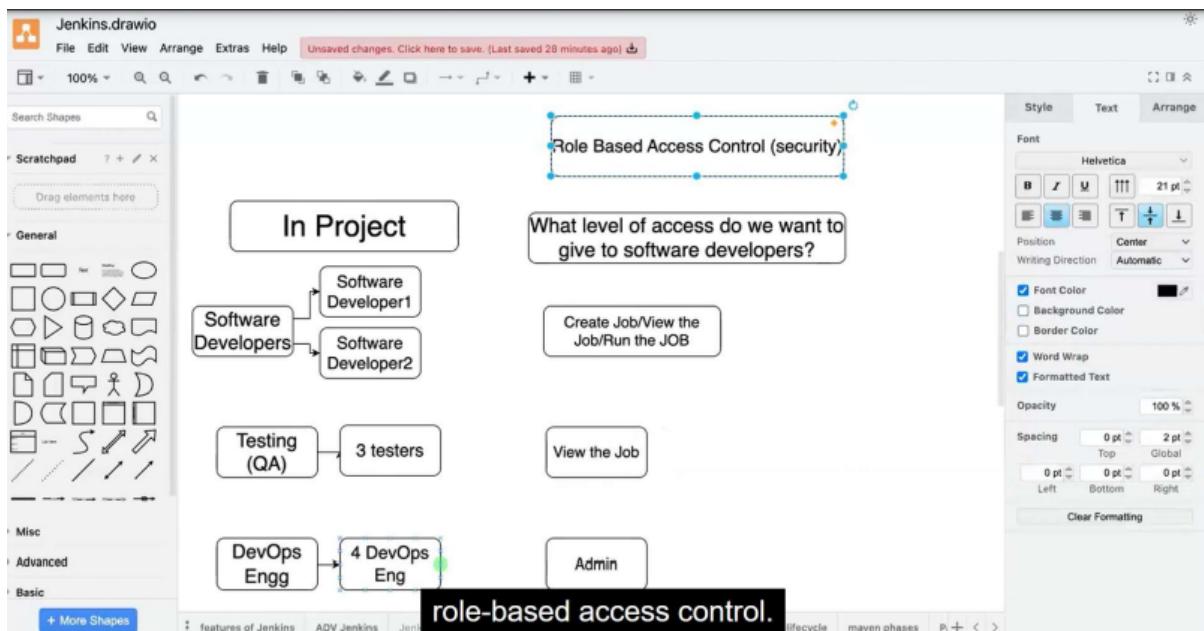
21. Available Plugins > Role based Authorization strategy(Install It)

The screenshot shows the 'Available plugins' page. A search bar at the top is highlighted with the word 'role'. A specific plugin, 'Role-based Authorization Strategy', is highlighted with a yellow box. This plugin is described as enabling user authorization using a Role-Based strategy. It was released 29 days ago. At the bottom right of the plugin card is a large blue 'Install' button, which is also highlighted with a yellow box.

22. Installed

The screenshot shows the Jenkins 'Plugins' page under 'Manage Jenkins'. On the left, there's a sidebar with links: 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress' (which is selected). The main area lists installed plugins with their status: Metrics (Success), Pipeline Graph View (Success), Git (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), Role-based Authorization Strategy (Success), and Loading plugin extensions (Success). At the bottom, there are two buttons: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

23. Role Based Access Control(Assigning the Role to the Employ)



24. Come Again Manage Jenkins > Security

The screenshot shows the Jenkins 'Manage Jenkins' interface under 'System Configuration'. On the left, there's a sidebar with links like 'Build History', 'Manage Jenkins' (which is selected and highlighted in grey), 'My Views', 'Build Queue', and 'Build Executor Status'. The main area has several sections: 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers), 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), 'Security' (Secure Jenkins; define who is allowed to access/use the system), 'Credentials' (Configure credentials), 'Users' (Create/delete/modify users that can log in to this Jenkins), 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), 'Appearance' (Configure the look and feel of Jenkins), and 'Credential Providers' (Configure the credential providers and types). The 'Security' section is specifically highlighted with a yellow box.

25. Selecting Role-Based Strategy Plugin and providing Authorization and save it.

The screenshot shows the Jenkins 'Manage Jenkins' interface under 'Security'. It includes sections for 'Authentication' (checkbox for 'Disable "Keep me signed in"'), 'Security Realm' (set to 'Jenkins' own user database'), and 'Authorization' (checkbox for 'Allow users to sign up'). Under 'Authorization', the 'Role-Based Strategy' option is highlighted with a yellow box. At the bottom are 'Save' and 'Apply' buttons.

26. Click Users, I think

The screenshot shows the Jenkins 'Manage Jenkins' interface under 'System Configuration'. The 'Security' section from the previous screenshot is still present. In the main area, the 'Users' section is highlighted with a yellow box. Other sections include 'Security' (Secure Jenkins; define who is allowed to access/use the system), 'Credentials' (Configure credentials), 'Manage and Assign Roles' (Handle permissions by creating roles and assigning them to users/groups), and 'Users' (Create/delete/modify users that can log in to this Jenkins).

27. Click Create User

The screenshot shows the Jenkins interface for managing users. At the top, there's a navigation bar with links for Dashboard, Manage Jenkins, and Jenkins' own user database. Below this, the 'Users' section is displayed with a table. The table has two columns: 'User ID' and 'Name'. There is one entry: 'salmanfarsi' under 'User ID' and 'SalmanFarsi' under 'Name'. To the right of the table is a blue button labeled '+ Create User', which is highlighted with a yellow box.

28. I am Creating Developer1 Role > Name > Sultan and Create user

This screenshot shows the 'Create User' form. It has several input fields: 'Username' (containing 'Developer1'), 'Password' (containing '*****'), 'Confirm password' (containing '*****'), 'Full name' (containing 'Sultan'), 'E-mail address' (containing 'example@gmail.com'), and a 'Create User' button at the bottom. A yellow box highlights the 'Create User' button.

29. I am Creating Tester Role > Name > Manoj and Create user

This screenshot shows the 'Create User' form again. The 'Username' field now contains 'Tester'. The other fields ('Password', 'Confirm password', 'Full name', and 'E-mail address') are still empty. The 'Create User' button at the bottom is highlighted with a yellow box.

30. I am Creating DevOps Role > Name > jhonny and Create user

The screenshot shows the Jenkins 'Create User' interface. The 'Username' field is filled with 'DevOps' and has a yellow border around it. Below it are fields for 'Password' (with dots), 'Confirm password' (with dots), 'Full name' (filled with 'jhonny'), and 'E-mail address' (filled with 'example2@gmail.com'). At the bottom is a blue 'Create User' button.

31. Come to back and We can see the all users which we are created

The screenshot shows the Jenkins 'Users' page. It lists four users: 'Developer1', 'DevOps', 'salmanfarsi', and 'Tester'. Each user entry includes a profile icon, the user's name, and two small icons for editing and deleting. At the top right of the table is a blue '+ Create User' button.

User ID	Name	Action
Developer1	Sultan	
DevOps	jhonny	
salmanfarsi	SalmanFarsi	
Tester	manoj	

32. Need to Install Active Directory Plugins

The screenshot shows the Jenkins 'Plugins' page. The 'Available plugins' tab is selected. A search bar at the top contains the text 'active dire'. In the list, the 'Active Directory' plugin by 'Authentication and User Management' is highlighted with a yellow box. This plugin is described as enabling authentication through Active Directory. Other visible plugins include 'Microsoft Entra ID (previously Azure AD)' and 'Windows Negotiate SSO'.

33. Installed

The screenshot shows the Jenkins 'Plugins' page under 'Manage Jenkins'. On the left, there are navigation links: 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress' (which is highlighted). The main area lists various Jenkins plugins with their status: Git (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), Role-based Authorization Strategy (Success), Loading plugin extensions (Success), Active Directory (Success), and Loading plugin extensions (Success). Below the list are two buttons: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

34. This time Go to Manage and Assign Roles

The screenshot shows the Jenkins 'Manage Jenkins' dashboard. On the left, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 idle, 2 idle). The main area has several cards: 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), 'Appearance' (Configure the look and feel of Jenkins), 'Security' (Secure Jenkins; define who is allowed to access/use the system), 'Credentials' (Configure credentials), 'Credential Providers' (Configure the credential providers and types), and 'Manage and Assign Roles' (Handle permissions by creating roles and assigning them to users/groups). The 'Manage and Assign Roles' link is highlighted with a yellow box.

35. Manage Roles and Save it.

The screenshot shows the Jenkins 'Manage Roles' page under 'Manage Jenkins'. On the left, there are links for 'Assign Roles', 'Permission Templates', and 'Role Strategy Macros'. The main area has two sections: 'Global roles' and 'Item roles'. The 'Global roles' section displays a grid of permissions for four roles: 'admin', 'Developer', 'Tester', and 'Devops'. The grid columns include: Overall, Credentials (ManageDomains, Create, Delete, Read), Agent (Connect, Create, Delete, Read, Update), Job (Create, Delete, Discover, Configure, Cancel, Run, View, SCM, Metrics), and Workspace (Read, Delete, Create, Configure, Update). The 'Item roles' section shows a list of roles ('Devops') and buttons for 'Save' and 'Apply'.

36. In Assign Roles > Add User

The screenshot shows the Jenkins 'Assign Roles' page under 'Manage and Assign Roles'. On the left, there's a sidebar with 'Manage Roles', 'Assign Roles' (which is highlighted with a yellow box), 'Permission Templates', and 'Role Strategy Macros'. The main area is titled 'Assign Roles' and contains a 'Global roles' section. This section has a grid where rows represent 'User/Group' (Anonymous, Authenticated Users, SalmanFarsi) and columns represent roles (Developer, Devops, Tester, admin). For the 'SalmanFarsi' row, the 'Developer' column has a checked checkbox. Below the grid are 'Add User' and 'Add Group' buttons, with 'Add User' also highlighted with a yellow box. Further down is an 'Item roles' section with a 'User/Group' dropdown and 'Save' and 'Apply' buttons, with 'Save' highlighted with a yellow box.

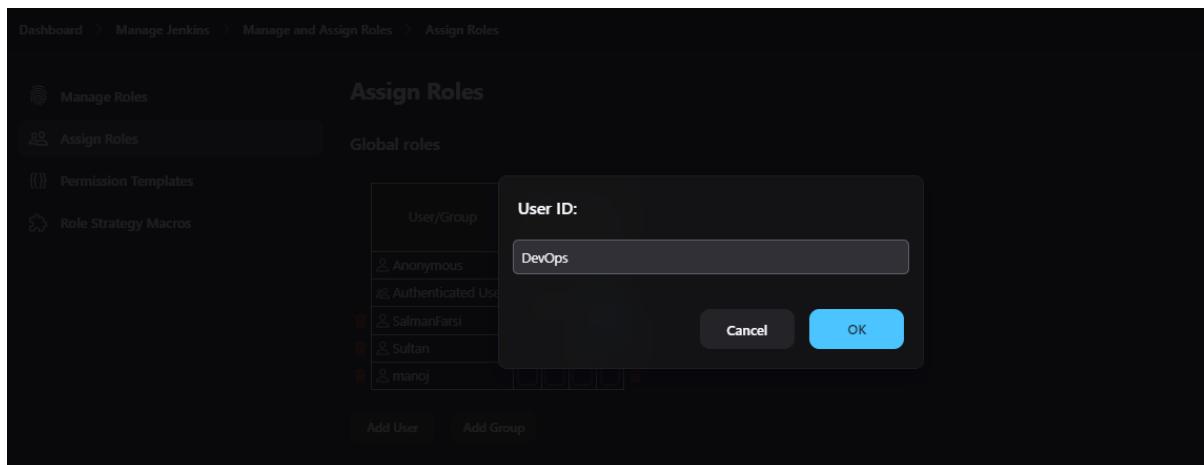
37. Developer1 > click ok

This screenshot shows the same Jenkins 'Assign Roles' page as above, but with a modal dialog open over it. The dialog is titled 'User ID:' and contains a text input field with 'Developer1'. At the bottom right of the dialog are 'Cancel' and 'OK' buttons, with 'OK' highlighted with a yellow box.

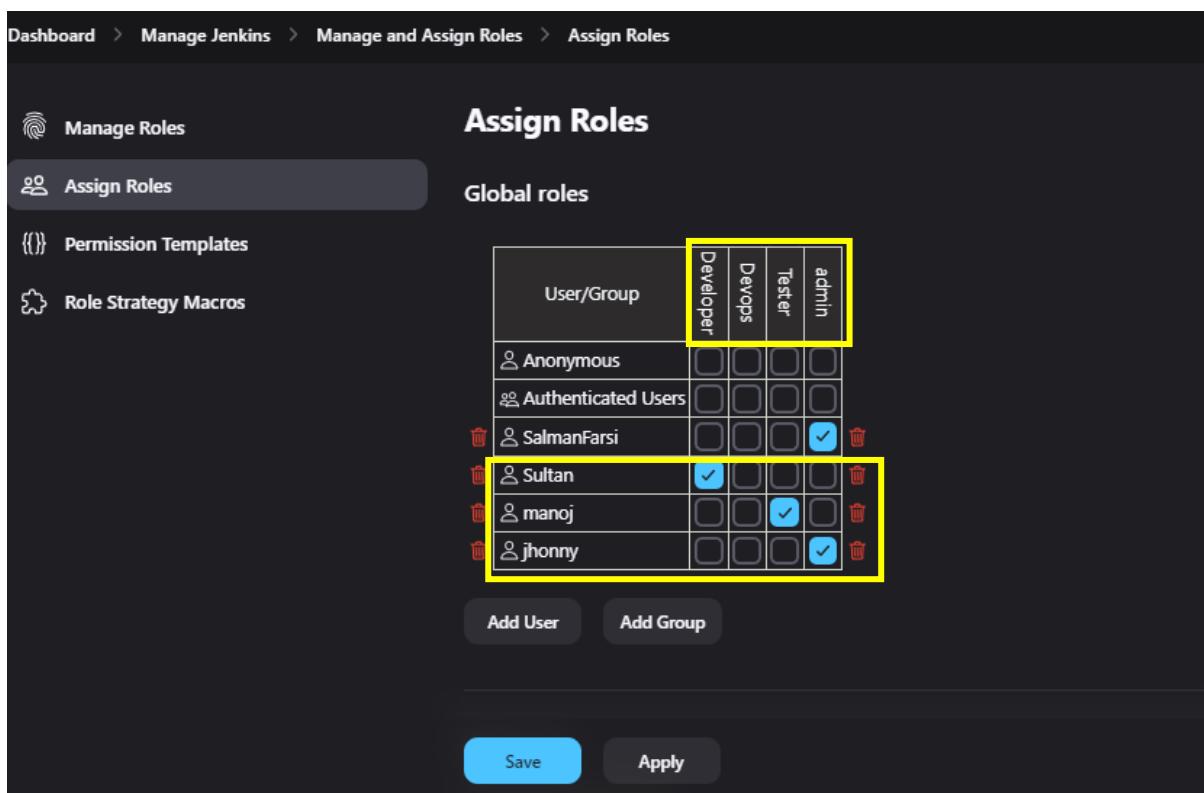
38. Like Add another user Tester click ok

This screenshot shows the Jenkins 'Assign Roles' page again with a modal dialog open. This time, the dialog has 'Tester' entered in the 'User ID:' field. The 'OK' button at the bottom right is highlighted with a yellow box.

39. Add **DevOps** user and ok



40. After that it will fetch the names and Assign developer,tester,devops according to name and sign out



41. Login With Developer1



Sign in to Jenkins

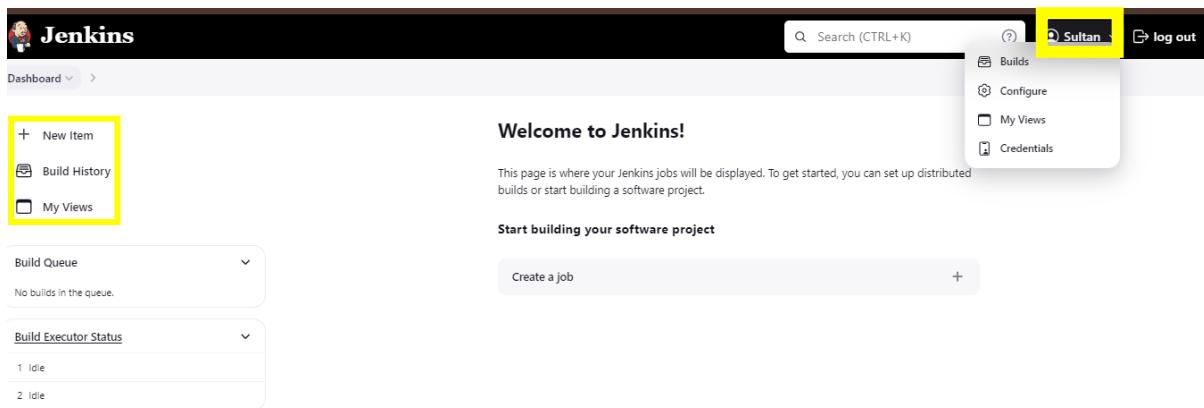
Username

Password

Keep me signed in

Sign in

42. See the limited features is there to the developer1 and he can read and create the job but cant run



The screenshot shows the Jenkins dashboard for a user named 'Developer1'. The top navigation bar includes links for 'Builds', 'Configure', 'My Views', and 'Credentials'. The main content area displays a 'Welcome to Jenkins!' message and a 'Start building your software project' button. On the left, there's a sidebar with options like '+ New Item', 'Build History', and 'My Views'. Below the sidebar, sections for 'Build Queue' (empty) and 'Build Executor Status' (showing 1 idle and 2 idle executors) are present.

43. Now Logging in as a Tester



Sign in to Jenkins

Username

Password

Keep me signed in

Sign in

44. There is not create to him

The screenshot shows the Jenkins dashboard for a user named 'manoj'. At the top right, there is a yellow box highlighting the user icon and the 'log out' link. Below the header, there is a navigation bar with two items: 'Build History' and 'My Views', both of which are highlighted with a yellow box. The main content area contains sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 1 Idle and 2 Idle executors).

45. Now Login as DevOps user

The screenshot shows the Jenkins sign-in page. On the left, there is a cartoon illustration of a man in a tuxedo holding a mug. The main form has fields for 'Username' (containing 'DevOps', highlighted with a yellow box) and 'Password' (containing '*****'). There is also a 'Keep me signed in' checkbox and a blue 'Sign in' button.

46. Having All Access. And like we provide accordingly to all these 3 roles

The screenshot shows the Jenkins dashboard for a user named 'jhonny'. At the top right, there is a yellow box highlighting the user icon and the 'log out' link. Below the header, there is a navigation bar with three items: '+ New Item', 'Build History', and 'Manage Jenkins', all of which are highlighted with a yellow box. The main content area contains sections for 'Build Queue' (showing 'No builds in the queue.'), 'Build Executor Status' (showing 1 Idle and 2 Idle executors), and a 'Create a job' button. On the right side, there is a 'Welcome to Jenkins!' message, a 'Start building your software project' section, and a 'Set up a distributed build' section with options for 'Set up an agent' and 'Configure a cloud'.

47. Now I logged in as a **Normal user** and create a job

The screenshot shows the Jenkins dashboard. At the top left is the Jenkins logo and the word "Dashboard". On the right, there's a search bar with "Search (CTRL+K)" and a user icon labeled "SalmanFarsi". Below the dashboard, there's a sidebar with links like "+ New Item", "Build History", "Manage Jenkins", and "My Views". The main area has a heading "Welcome to Jenkins!" and a sub-heading "Start building your software project". It includes sections for "Create a job", "Set up a distributed build", "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". A "Build Queue" section shows "No builds in the queue." and a "Build Executor Status" section showing 1 Idle and 2 Idle.

48. Now Creating a **Job** and Selected **Freestyle Project**

The screenshot shows the "Enter an item name" dialog. The input field contains "Job1", which is highlighted with a yellow box. Below the input field, there are three project types listed: "Freestyle project" (selected), "Pipeline", and "Multi-configuration project". Each type has a small icon and a brief description. At the bottom of the dialog is a blue "OK" button.

49. Go to Inside the Job, In the General Description, I Wrote something

The screenshot shows the "Configuration" dialog for a job named "Job1". The left sidebar has tabs for "General", "Source Code Management", "Build Triggers", "Build Environment", "Build Steps", and "Post-build Actions". The "General" tab is selected and highlighted with a yellow box. The main area shows the "General" configuration. Under the "Description" section, the text "The Purpose of this Job Exploring Jenkins" is written and highlighted with a yellow box. There are also checkboxes for "Discard old builds", "GitHub project", "This project is parameterized", and "Throttle builds". At the bottom are "Save" and "Apply" buttons.

50. Go to Build Environment > Execute Shell

The screenshot shows the Jenkins job configuration interface for 'Job1'. The 'Configure' section is open. On the left, there are several tabs: General, Source Code Management, Build Triggers, Build Environment (which is highlighted with a yellow box), Build Steps, and Post-build Actions. Under the 'Build Environment' tab, there is a 'Build Steps' section. A dropdown menu titled 'Add build step ^' is open, showing options like 'Execute Windows batch command', 'Execute shell' (which is highlighted with a yellow box), 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'. There are also checkboxes for 'Add timestamps to the Console Output', 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'.

51. echo "Hello World"

This screenshot shows the 'Execute shell' step configuration. The 'Command' field contains the text 'echo "Hello World"'. The 'Save' and 'Apply' buttons are visible at the bottom of the step configuration panel.

52. Go to Dashboard > Job1 > Build Now

The screenshot shows the Jenkins dashboard with 'Job1' selected. The left sidebar has buttons for Status, Changes, Workspace, Build Now (which is highlighted with a yellow box), Configure, Delete Project, and Rename. The main area is titled 'Job1' with the subtitle 'The Purpose of this Job Exploring Jenkins'. It includes sections for Permalinks, Build History (with a recent build highlighted with a yellow box), and links for Atom feed for all and Atom feed for failures.

53. Click on Console Output to see the logs and its created the Job1 directory and u can see the location at the terminal, just copy it and cd paste the path

The screenshot shows the Jenkins interface for a job named 'Job1'. The 'Console Output' link in the left sidebar is highlighted with a yellow box. The main content area displays the build logs for build #1, which include the command 'echo Hello World' and the output 'Hello World'. The status bar at the top right shows the user 'SalmanFarsi'.

54. Again go to job1 > Configure

The screenshot shows the 'Configure' page for the 'Job1' project. The 'Build Steps' section is highlighted with a yellow box. It contains a single 'Execute shell' step with the command 'xyz123' entered into the text area. Other sections like 'General', 'Source Code Management', and 'Post-build Actions' are also visible.

55. In the Build steps > Execute Shell change xyz123, it will show an error because there is no command with xyz123

The screenshot shows the 'Configure' page for the 'Job1' project. The 'Build Steps' section is highlighted with a yellow box. The 'Execute shell' step has the command 'xyz123' entered. A red error message 'xyz123: command not found' is displayed below the command input field. The 'Post-build Actions' section is also visible at the bottom.

56. Click Build now and its shown an error

The screenshot shows the Jenkins Job1 dashboard. On the left, there's a sidebar with options like Status, Changes, Workspace, Build Now (which is highlighted with a yellow box), Configure, Delete Project, and Rename. The main area has a green checkmark icon and the text "JOB 1". Below it, it says "The Purpose of this Job Exploring Jenkins". There's a "Permalinks" section with a link. On the right, there are "Edit description" and "Disable Project" buttons. At the bottom, there's a "Build History" section with three entries: #3 (Jun 3, 2024, 7:50 AM) which is failing (red circle), #2 (Jun 3, 2024, 7:50 AM) which is stable (green circle), and #1 (Jun 3, 2024, 7:48 AM) which is successful (green circle).

57. Go Console output and Check the Error.

The screenshot shows the Jenkins Job1 build #3 details page. The sidebar on the left includes Status (highlighted with a yellow box), Changes, Console Output (highlighted with a yellow box), Edit Build Information, Delete build "#3", Timings, and Previous Build. The main content shows build #3 started by user SalmanFarsi at Jun 3, 2024, 7:50:52 AM. It indicates "No changes." and "Started by user SalmanFarsi". The "Timings" section shows: 2 ms waiting, 14 ms build duration, and 16 ms total from scheduled to completion. A warning message at the bottom states: "⚠ The new name is the same as the current name." and has a "Rename" button.

58. U can Rename

The screenshot shows the Jenkins Job1 rename project dialog. The sidebar on the left includes Status, Changes, Workspace, Build Now, Configure, Delete Project, and a "Rename" button (highlighted with a yellow box). The main area has a title "Rename Project Job1" and a "New Name" input field containing "Job1" (highlighted with a yellow box). A warning message below the input field says: "⚠ The new name is the same as the current name." and has a "Rename" button.

59. And U can check in the Workspace what are the directories and files are available, so there is no file available currently

The screenshot shows the Jenkins interface for the 'Job1' project. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'SalmanFarsi'. Below the header, the breadcrumb navigation shows 'Dashboard > Job1 > Workspace of Job1 on Built-In Node'. The main content area is titled 'Workspace of Job1 on Built-In Node'. On the left, a sidebar lists options: Status, Changes, Workspace (which is highlighted with a yellow box), Wipe Out Current Workspace, Build Now, Configure (highlighted with a yellow box), Delete Project, and Rename. The central workspace area shows a directory structure: 'Job1 /' followed by a box containing 'No files in directory'. At the bottom, there is a 'Build History' section with a 'trend' dropdown set to 'trend' and a 'Filter...' input field.

60. Go to Configure.

The screenshot shows the Jenkins configuration page for the 'Job1' project. The top navigation bar shows 'Dashboard > Job1 >'. The main title is 'Job1' with a checkmark icon. Below the title, the purpose of the job is listed as 'The Purpose of this Job Exploring Jenkins'. On the right, there are buttons for 'Edit description' and 'Disable Project'. The left sidebar contains options: Status, Changes, Workspace, Build Now, Configure (highlighted with a yellow box), Delete Project, and Rename. The central content area features a 'Permalinks' section with a bulleted list of build links. Below this is a 'Build History' section with a 'trend' dropdown set to 'trend' and a 'Filter...' input field. A specific build entry for '#1' is shown, dated 'Jun 3, 2024, 7:48 AM', with links for 'Atom feed for all' and 'Atom feed for failures'.

61. Go to Build Steps > Execute shell and Create one touch file1 and save it.

The screenshot shows the Jenkins job configuration interface. The left sidebar has a 'Build Steps' section highlighted with a yellow box. In the main area, there is a 'Execute shell' step with the command 'touch file1' entered. Below the command input is an 'Advanced' dropdown. At the bottom are 'Save' and 'Apply' buttons.

62. Click on Build Now.

The screenshot shows the Jenkins job page for 'Job1'. The 'Build Now' button is highlighted with a yellow box. On the right, there is a 'Permalinks' section with a list of recent builds. On the left, there is a 'Build History' sidebar showing two builds: '#4' (selected) and '#1'. The build history table includes columns for build number, date, and time.

63. Now see its created one touch file named as file1

The screenshot shows the Jenkins console output for build '#4'. The 'Console Output' tab is selected. The output text is highlighted with a yellow box and shows the command 'touch file1' being executed and completed successfully.

```
Started by user SalmanFarsi
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Job1
[Job1] $ /bin/sh -xe /tmp/jenkins3183597994151051243.sh
+ touch file1
Finished: SUCCESS
```

64. Now we can See the file1 in workspace

The screenshot shows the Jenkins interface for the 'Job1' workspace. The top navigation bar includes 'Dashboard', 'Job1', and 'Workspace of Job1 on Built-In Node'. The user 'SalmanFarsi' is logged in. The main area is titled 'Workspace of Job1 on Built-In Node'. On the left, there's a sidebar with options like 'Status', 'Changes', 'Workspace' (which is selected and highlighted in yellow), 'Wipe Out Current Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The 'Workspace' section shows a single item: 'file1' (Jun 3, 2024, 8:43:44 AM, 0 B). A download link '(all files in zip)' is also present.

65. Location in the terminal

```
root@Jenkins-Master:/var/lib/ + ~
ubuntu@Jenkins-Master:~$ sudo su -
root@Jenkins-Master:~# cd /var/lib/jenkins
root@Jenkins-Master:/var/lib/jenkins# ls
```

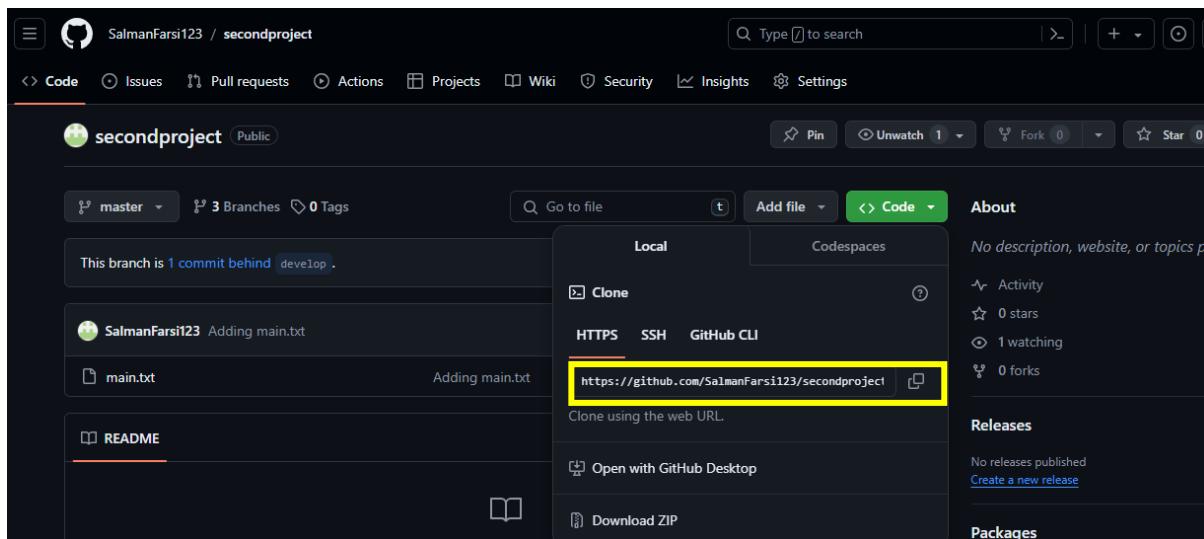
66. we can see the workspace.

```
root@Jenkins-Master:/var/lib/ + ~
ubuntu@Jenkins-Master:~$ sudo su -
root@Jenkins-Master:~# cd /var/lib/jenkins
root@Jenkins-Master:/var/lib/jenkins# ls
config.xml
hudson.model.UpdateCenter.xml
hudson.plugins.git.GitTool.xml
identity.key.enc
jenkins.install.InstallUtil.lastExecVersion
jenkins.install.UpgradeWizard.state
jenkins.model.JenkinsLocationConfiguration.xml
jenkins.plugins.git.GitHooksConfiguration.xml
jenkins.security.QueueItemAuthenticatorConfiguration.xml
jenkins.security.UpdateSiteWarningsConfiguration.xml
jenkins.security.apitoken.ApiTokenPropertyConfiguration.xml
jenkins.telemetry.Correlator.xml
jobs
logs
nodeMonitors.xml
org.jenkinsci.plugins.gitclient.GitHostKeyVerificationConfiguration.xml
plugins
queue.xml
queue.xml.bak
secret.key
secret.key.not-so-secret
secrets
updates
userContent
users
workspace
root@Jenkins-Master:/var/lib/jenkins#
```

67. We can see the Job1 directory in the Workspace directory

```
root@Jenkins-Master:/var/lib/jenkins# cd workspace
root@Jenkins-Master:/var/lib/jenkins/workspace# ls
Job1
root@Jenkins-Master:/var/lib/jenkins/workspace# cd Job1
root@Jenkins-Master:/var/lib/jenkins/workspace/Job1# ls
file1
root@Jenkins-Master:/var/lib/jenkins/workspace/Job1#
```

68. Now Another Task, Copy your [Github any Project repository](#)



69. Come to Jenkins and go [Job1 > Configuration > Source Code Management](#) in that [Git and Paste](#) the Git Hub Repository Link

A screenshot of the Jenkins Job1 configuration screen. Under the 'Source Code Management' section, the 'Git' option is selected. The 'Repository URL' field contains the value 'https://github.com/SalmanFarsi123/secondproject.git', which is also highlighted with a yellow box. The 'Credentials' dropdown is set to '- none -'. At the bottom, there are 'Save' and 'Apply' buttons.

70. I am Fetching the data from master branch

The screenshot shows the Jenkins job configuration interface. On the left, a sidebar lists options: General, Source Code Management (selected), Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main panel is titled 'Configure' and contains 'Branches to build'. A text input field contains the value '/master', which is highlighted with a yellow box. Below this is a 'Repository browser' dropdown set to '(Auto)'. At the bottom are 'Save' and 'Apply' buttons.

71. Build Now.

The screenshot shows the Jenkins dashboard for 'Job1'. The left sidebar includes 'Status', 'Changes', 'Workspace', 'Build Now' (highlighted with a yellow box), 'Configure', 'Delete Project', and 'Rename'. The main area displays the purpose of the job: 'The Purpose of this Job Exploring Jenkins'. It features a 'Permalinks' section with a bulleted list of recent builds and a 'Build History' table with one entry for build #5. The 'Build History' table has columns for 'Build Number', 'Last Result', 'Timestamp', and 'Duration'. The entry for build #5 shows 'trend' and '8:55 AM'.

72. Click Console Output

The screenshot shows the Jenkins build details for '#5 (Jun 3, 2024, 8:55:06 AM)'. The left sidebar includes 'Status', 'Changes', 'Console Output' (highlighted with a yellow box), 'Edit Build Information', 'Delete build #5', 'Timings', 'Git Build Data', and 'Previous Build'. The main area shows 'No changes.' and 'Started by user SalmanFarsi'. It includes a 'Timings' section with a pie chart and a table of run statistics. At the bottom, it shows the git revision and repository information.

73. Added main.txt in workspace.

Dashboard > Job1 > #5 > Console Output

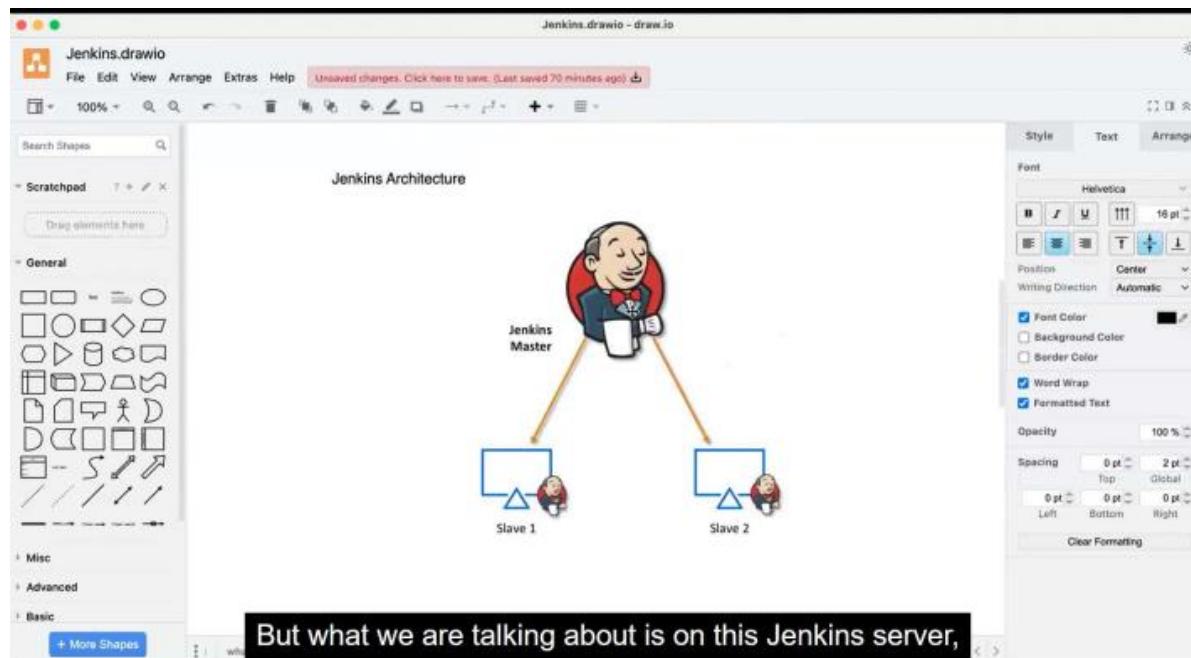
Console Output

Started by user **SalmanFarsi**
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Job1
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
 from https://github.com/SalmanFarsi123/secondproject.git
> git init /var/lib/jenkins/workspace/Job1 # timeout=10
 fetching upstream changes via https://github.com/SalmanFarsi123/secondproject.git
> git config remote.origin.url https://github.com/SalmanFarsi123/secondproject.git
> 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 79c5758da0a316516178ff677ed20725ec95df3d7 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 79c5758da0a316516178ff677ed20725ec95df3d7 # timeout=10
Commit message: "Adding main.txt"
First time build. Skipping changelog.
[Job1] \$ /bin/sh -xe /tmp/jenkins10410775220257290545.sh
+ touch file1
Finished: SUCCESS

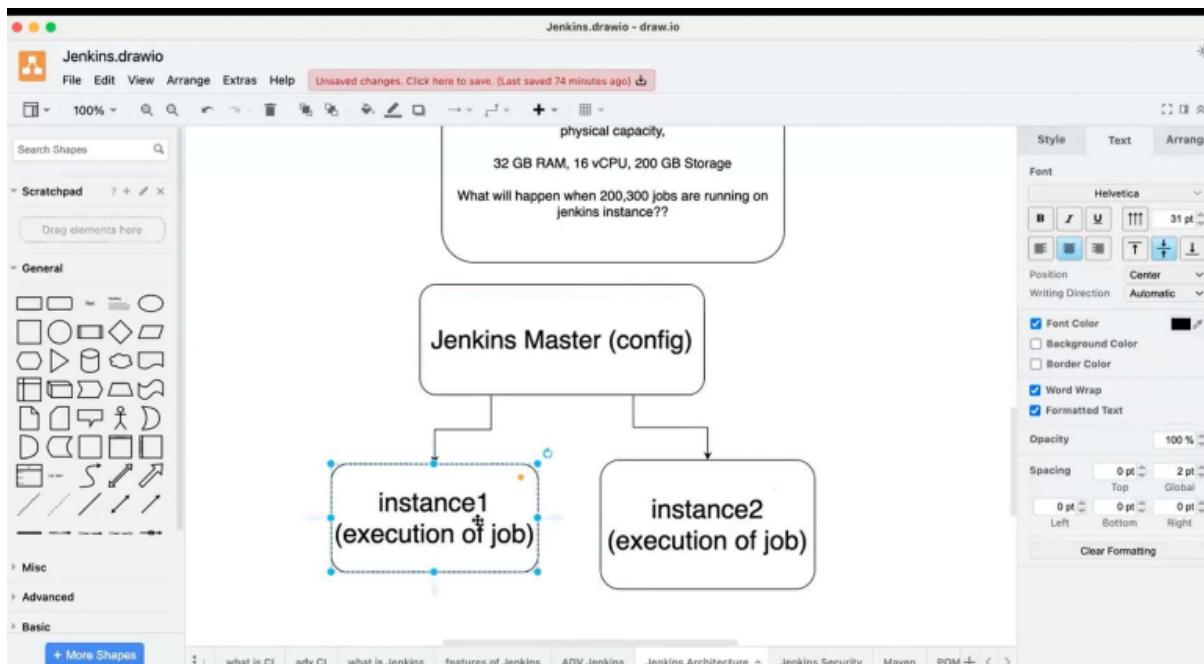
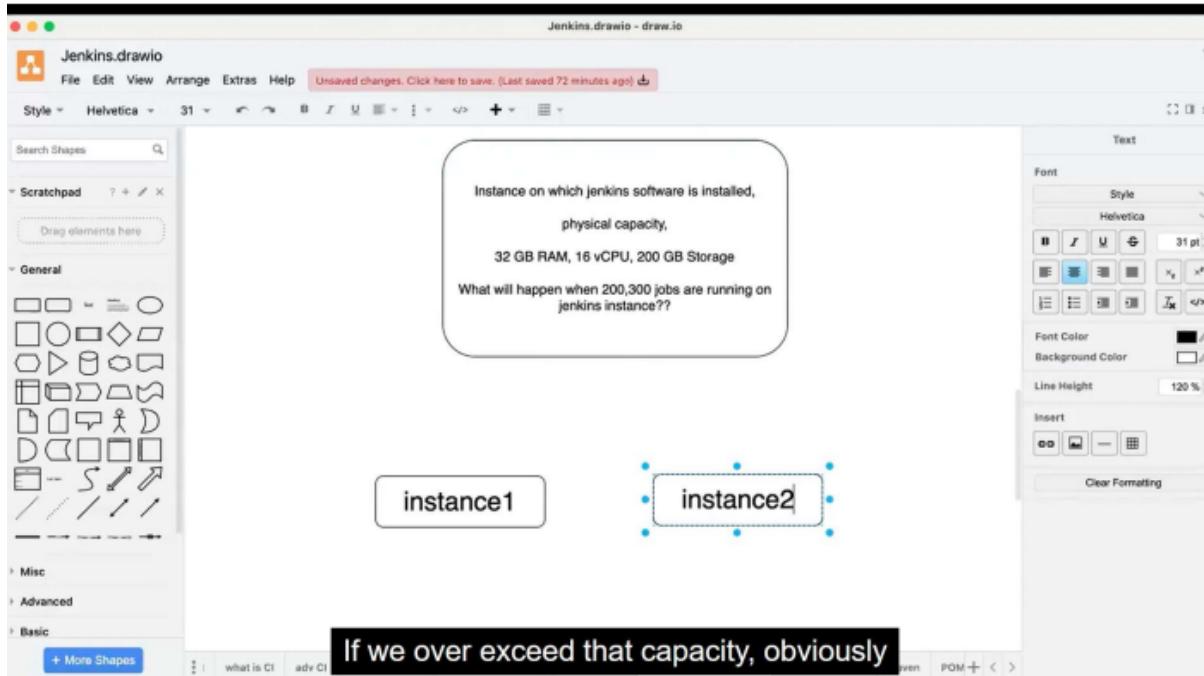
74. we can see main.txt

```
ubuntu@Jenkins-Master:/var      X  + | ^  
ubuntu@Jenkins-Master:/var/lib/jenkins$ cd workspace  
ubuntu@Jenkins-Master:/var/lib/jenkins/workspace$ ls  
Job1  
ubuntu@Jenkins-Master:/var/lib/jenkins/workspace$ cd Job1  
ubuntu@Jenkins-Master:/var/lib/jenkins/workspace/Job1$ ls  
file1  main.txt  
ubuntu@Jenkins-Master:/var/lib/jenkins/workspace/Job1$ cat main.txt  
ubuntu@Jenkins-Master:/var/lib/jenkins/workspace/Job1$
```

75. So this Master and save Node Concept we can integrate the stuff in slaves.



76. So we are working in single instances and storing and integrating in single instance then it will be crash so we are integrate and performing in slave nodes and we no need to install Jenkins in slave node.



77. Launching another Instances with the name of Jenkins-Slave.

Instances (10) Info								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Actions
<input type="checkbox"/>	Kubernetes-Master	i-0102e4cf2c64df9658	Stopped	t2.medium	-	View alarms	us-east-1e	Launch instances
<input type="checkbox"/>	Kubernetes-Slave	i-0e0d3dc5efcf9445	Stopped	t2.medium	-	View alarms	us-east-1e	Launch instances
<input type="checkbox"/>	Jenkins-Master	i-02ae4cf2c64df9658	Running	t2.medium	Initializing	View alarms	us-east-1e	Launch instances
<input type="checkbox"/>	Jenkins-Slave	i-01a0d9d57466e4ca6	Running	t2.medium	Initializing	View alarms	us-east-1e	Launch instances

78. Change hostname.

```
ubuntu@ip-172-31-50-152:~$ sudo hostnamectl set-hostname Jenkins-Slave
ubuntu@ip-172-31-50-152:~$ exit
logout
Connection to ec2-18-209-178-25.compute-1.amazonaws.com closed.

C:\Users\shaik\Desktop\Cloud Computing\Aws-key pairs>ssh -i "Master-Client.pem" ubuntu@ec2-18-209-178-25.compute-1.amazonaws.com
```

79. update in slave node

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

80. We need to Install java at slave node

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

81. Installed successfully.

```
ubuntu@Jenkins-Slave:~$ sudo apt install openjdk-17-jre -y
Reading package lists... done
Building dependency tree... done
Reading state information... done
The following additional packages will be installed:
adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-common at-spi2-core ca-certificates-java
dconf-gsettings-backend dconf-service fontconfig fonts-dejavu-extra gsettings-desktop-schemas gtk-update-icon-cache
hicolor-icon-theme humanity-icon-theme java-common libasound2-data libasound2t64 libatk-bridge2.0-0t64
libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3 libavahi-common-data
libavahi-common3 libcairo-gobject2 libcairo2 libcups2t64 libdatriel libdconf1 libdrm-amdgpu libdrm-intel
libdrm-nouveau2 libdrm-radeon1 libgail-common libgail18t64 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgif7 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0
libgraphite2-3 libgtk2.0-0t64 libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice6 liblcms2-2 liblvm17t64
libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpiciaccess0 libpcslite1 libpixman-1-0 librsvg2-2
librsvg2-common libsm6 libthai-data libvulkan1 libwayland-client0 libxi1-xcb1 libxaw7 libxcb-dri2-0
libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0 libxcb-shape0 libxcb-shm0 libxcb-sync1
libxcb-xfixes0 libxcompositel libxcursor1 libxdamage1 libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxml2
libxrandr2 libxrender1 libxshmfence1 libxt6t64 libxtst6 libxv1 libxf86dg1 libxf86vm1 mesa-vulkan-drivers
openjdk-17-jre-headless session-migration ubuntu-mono x11-common x11-utils
Suggested packages:
```

82. Click Security.

The screenshot shows the Jenkins 'Manage Jenkins' page with a yellow box highlighting the 'Security' link in the top navigation bar. The 'Security' section is also highlighted with a yellow box. It contains links for 'Security', 'Manage and Assign Roles', 'Credentials', 'Credential Providers', and 'Users'. Below this is a 'Status Information' section with links for 'System Information', 'System Log', and 'Load Statistics'.

83. Agents > Fixed > 5666 port no means connecting b/w client-server relationship like here master and slave relation

The screenshot shows the Jenkins 'Manage Jenkins' page with a yellow box highlighting the 'Security' link in the top navigation bar. The 'Agents' section is highlighted with a yellow box. It shows the 'TCP port for inbound agents' configuration, where 'Fixed' is selected and the value '5666' is entered. There are also options for 'Random' and 'Disable'. At the bottom are 'Save' and 'Apply' buttons.

84. Come back and click nodes

The screenshot shows the Jenkins 'Manage Jenkins' page with a yellow box highlighting the 'Nodes' link in the left sidebar under 'Build Executor Status'. The 'Nodes' section is highlighted with a yellow box. It contains links for 'System', 'Tools', 'Clouds', 'Plugins', and 'Appearance'. Below this is a 'Security' section with links for 'Security', 'Manage and Assign Roles', 'Credentials', 'Credential Providers', and 'Users'.

85. Dashboard > Manage Jenkins > Nodes

The screenshot shows the Jenkins 'Nodes' page. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main area is titled 'Nodes' and contains a table with one row. The table columns are: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The single node listed is 'Built-In Node' (Linux (amd64)), which is 'In sync' with 3.89 GiB free disk space, 0 B free swap space, 3.89 GiB free temp space, and 0ms response time. A yellow box highlights the '+ New Node' button at the top right of the table header.

86. Click New Node and Named as slave1

The screenshot shows the 'New node' configuration page. The 'Node name' field contains 'slave1', which is highlighted with a yellow box. The 'Type' section has a radio button selected for 'Permanent Agent'. Below it, a description states: '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.' At the bottom, a blue 'Create' button is visible.

87. Number of executors given 10 and remote root directory path I shared which going to perform the task in that location and labels name as slave1

The screenshot shows the 'New node' configuration page with the following fields filled in:

- Number of executors:** 10 (highlighted with a yellow box)
- Remote root directory:** /home/ubuntu/jenkins (highlighted with a yellow box)
- Labels:** slave1 (highlighted with a yellow box)
- Usage:** Use this node as much as possible
- Launch method:** Launch agent by connecting it to the controller

A blue 'Save' button is located at the bottom left.

88. Nothing do here just save it.

Launch method ?

Launch agent by connecting it to the controller

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

89. Now its added slave node but its not connected to master node so Click on Slave1 to connect

The screenshot shows the Jenkins 'Nodes' page. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (Built-In Node, 1 Idle, 2 Idle, slave1 (offline)). The main area is titled 'Nodes' and contains a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. There are two rows: 'Built-In Node' (Sync, 3.89 GiB free disk, 0 B swap, 3.89 GiB temp, 0ms response) and 'slave1' (Data obtained, N/A for all metrics). The 'slave1' row has a yellow box around it, indicating it is selected or highlighted.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.89 GiB	0 B	3.89 GiB	0ms
	slave1		N/A	N/A	N/A	N/A	N/A

90. Copy the commands

The screenshot shows the Jenkins 'Agent slave1' page. On the left, there is a sidebar with links: Status, Delete Agent, Configure, Build History, Load Statistics, and Log. The main area is titled 'Agent slave1' and contains a section 'Run from agent command line: (Unix)' with a text input field. The input field contains a command with a yellow box around it, indicating it is the copied command.

```
curl -s0 http://54.165.12.194:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://54.165.12.194:8080/ -secret 392444ca480dd3a7ea6178ee18bfb752766138f43477789037841c58c57b327
-name slave1 -workDir "/home/ubuntu/jenkins"
```

Dashboard > Manage Jenkins > Nodes > slave1

Agent slave1

Status Delete Agent Configure Build History Load Statistics Log

Run from agent command line: (Unix)

```
curl -s0 http://54.165.12.194:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://54.165.12.194:8080/ -secret 392444ca480dd3a7ea6178ee18bfbb752766138f43477789037841c58c57b327
-name slave1 -workDir "/home/ubuntu/jenkins"
```

Add description

91. Observe here Public-Ip of Jenkins Master, it was old Public Ip address of Jenkins Master but after Instance Stopped and Login again and the IP address was changed so U change the IP address accordingly

```
ubuntu@Jenkins-Slave:~$ curl -s0 http://54.165.12.194:8080/jnlpJars/agent.jar
```

92. See the Jenkins Master Current Public-IP Address

Not secure 54.210.89.89:8080/manage/computer/slave1

Interfacing SIM900... Downloads New Tab Python Exercises C... MEGA youtube - Polarity Y... Women's Health: 25... Imported

Dashboard > Manage Jenkins > Nodes > slave1

Agent slave1

Status Delete Agent Configure Build History Load Statistics Log

Run from agent command line: (Unix)

```
curl -s0 http://54.165.12.194:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://54.165.12.194:8080/ -secret 392444ca480dd3a7ea6178ee18bfbb752766138f43477789037841c58c57b327
-name slave1 -workDir "/home/ubuntu/jenkins"
```

Add description

93. So here I changed the IP address and enter it.

```
ubuntu@Jenkins-Slave:~$ curl -s0 http://54.210.89.89:8080/jnlpJars/agent.jar
ubuntu@Jenkins-Slave:~$ ls
agent.jar
ubuntu@Jenkins-Slave:~$
```

94. Now paste the second command enter it.

```
ubuntu@Jenkins-Slave:~$ java -jar agent.jar -url http://54.210.89.89:8080/ -secret 392444ca480dd3a7ea6178ee18bfbb752766138f43477789037841c58c57b327
-name slave1 -workDir "/home/ubuntu/jenkins" &
[1] 3300
```

95. And it will be connected.

```
Jun 03, 2024 11:01:04 AM hudson.remoting.Launcher$CuiListener status
INFO: Remote identity confirmed: bd:de:59:4f:e7:89:21:4e:94:7c:4a:85:8d:55:81:85
Jun 03, 2024 11:01:04 AM hudson.remoting.Launcher$CuiListener status
INFO: Connected
ubuntu@Jenkins-Slave:~$
```

96. Now Its Connected.

The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with sections like 'Build Queue' (No builds in the queue), 'Build Executor Status' (Built-In Node: 1 Idle, 2 Idle; slave1: 1 Idle, 2 Idle), and 'Clouds'. The main area is titled 'Nodes' and contains a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. It lists two nodes: 'Built-In Node' (Linux (amd64), In sync, 3.79 GiB free disk, 0B free swap, 3.79 GiB free temp, 0ms response) and 'slave1' (Linux (amd64), In sync, 4.44 GiB free disk, 0B free swap, 4.44 GiB free temp, 74ms response). The 'slave1' row is highlighted with a yellow background. A legend at the bottom shows icons for S, M, and L.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.79 GiB	0 B	3.79 GiB	0ms
	slave1	Linux (amd64)	In sync	4.44 GiB	0 B	4.44 GiB	74ms

97. This is Inside the Slave1

The screenshot shows the Jenkins 'Agent slave1' page. The left sidebar includes options like 'Status' (Delete Agent, Configure, Build History, Load Statistics, Script Console, Log, System Information, Disconnect), 'Monitoring Data' (Add description), and 'Projects tied to slave1' (None). The main content area displays the title 'Agent slave1' and a message 'Agent is connected.' with a timestamp '2023-09-12 10:30:00'.

98. Come to Dashboard.

The screenshot shows the Jenkins 'Dashboard' page. The left sidebar has links for 'New Item', 'Build History', 'Manage Jenkins', 'My Views', 'Build Queue' (No builds in the queue), and 'Build Executor Status' (Built-In Node: 1 Idle, 2 Idle; slave1). The main area features a table for 'Last Success', 'Last Failure', and 'Last Duration' with one entry: 'Job1' (Last Success: 2 hr 12 min, Last Failure: N/A, Last Duration: 3.4 sec). A legend at the bottom shows icons for S, W, and L.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Job1	2 hr 12 min #5	N/A	3.4 sec

99. Go to Job1 > Configuration > General > Restrict where this project can be run > Label Name we given slave1, so now it will going to run in slave node.

The screenshot shows the Jenkins configuration page for 'Job1'. In the 'General' section, there is a checkbox labeled 'Restrict where this project can be run' which is checked. Below this, a 'Label Expression' field contains the value 'slave1'. A yellow box highlights this entire section. At the bottom of the page are 'Save' and 'Apply' buttons.

100. SCM > Git > pasting my github project repository

The screenshot shows the Jenkins configuration page for 'Job1'. In the 'Source Code Management' section, the 'Git' option is selected. Under 'Repositories', there is a 'Repository URL' field containing 'https://github.com/SalmanFarsi123/secondproject.git'. A yellow box highlights this field. At the bottom of the page are 'Save' and 'Apply' buttons.

101. Fetching the data from Master branch

The screenshot shows the Jenkins configuration page for 'Job1'. In the 'Build Triggers' section, there is a 'Branch Specifier' field containing '/master'. A yellow box highlights this field. At the bottom of the page are 'Save' and 'Apply' buttons.

102. Build Steps > Typed echo “clone successfully”

The screenshot shows the Jenkins Job1 configuration page. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected and highlighted with a yellow box), and 'Post-build Actions'. The main area is titled 'Build Steps' and contains a single step named 'Execute shell'. The command entered is 'echo "clone Successfully"'. There is an 'Advanced' dropdown and buttons for 'Save' and 'Apply'.

103. Build Now and go inside the job1 and go inside the console output

The screenshot shows the Jenkins Job1 build #6 status page. The top navigation bar shows 'Dashboard > Job1 > #6'. The main content area shows the build status as 'Success' (#6 (Jun 3, 2024, 11:13:42 AM)). It includes sections for 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Delete build #6', 'Timings', 'Git Build Data', and 'Previous Build'. The 'Console Output' section is expanded, showing the build log. Key logs include:

```
</> No changes.  
⌚ Started by user SalmanFarsi  
⌚ This run spent:

- 59 ms waiting;
- 6.5 sec build duration;
- 6.6 sec total from scheduled to completion.

git Revision: 79c5758da0316516178ff677ed20725ec95df3d7  
Repository: https://github.com/SalmanFarsi123/secondproject.git  
refs/remotes/origin/master
```

104. Its ran successfully.

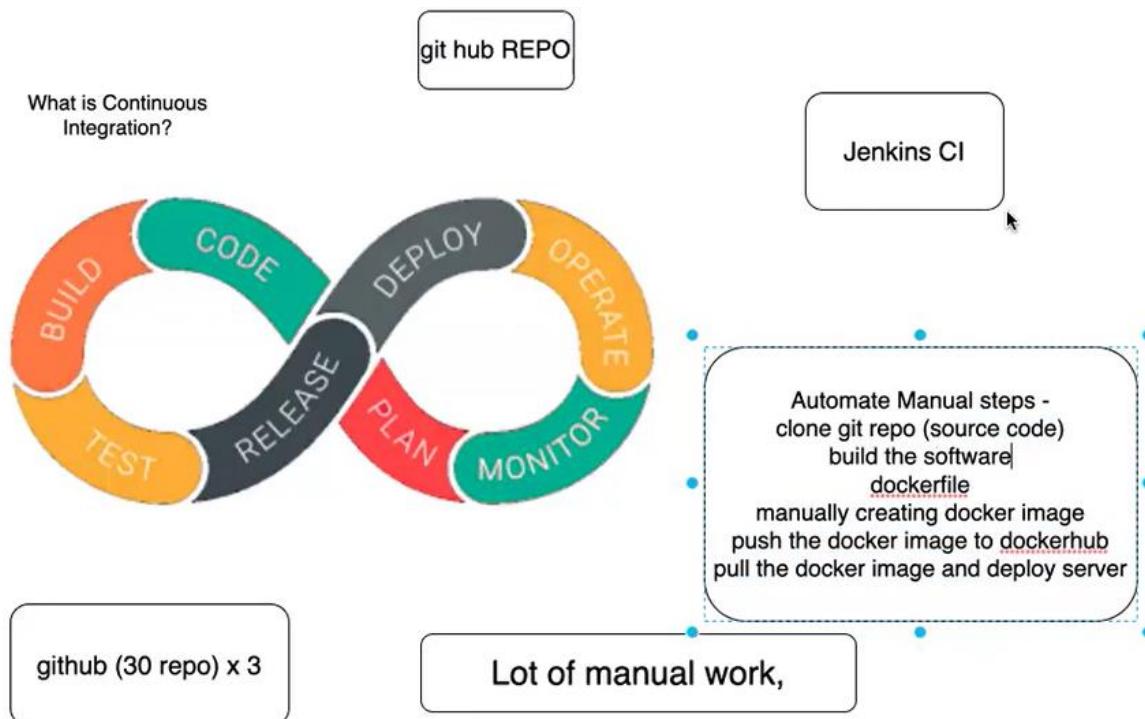
The screenshot shows the Jenkins Job1 build #6 console output. The top navigation bar shows 'Dashboard > Job1 > #6 > Console Output'. The main content area shows the build log. Key logs include:

```
Started by user SalmanFarsi  
Running as SYSTEM  
Building remotely on slave1 in workspace /home/ubuntu/jenkins/workspace/Job1  
The recommended git tool is: NONE  
No credentials specified  
Cloning the remote Git repository  
Cloning repository https://github.com/SalmanFarsi123/secondproject.git  
> git init /home/ubuntu/jenkins/workspace/Job1 # timeout=10  
> git --version # timeout=10  
> git --version # "git version 2.43.0"  
> git fetch --tags --force --progress -- https://github.com/SalmanFarsi123/secondproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10  
> git config remote.origin.url https://github.com/SalmanFarsi123/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"  
> git rev-list --no-walk 79c5758da0316516178ff677ed20725ec95df3d7 # timeout=10  
[Job1] $ /bin/sh -xe /tmp/jenkins11865597202986845843.sh  
+ echo clone Successfully  
clone Successfully  
Finished: SUCCESS
```

105. Now I came to slave node, I gave remote directory repo is `/home/ubuntu/Jenkins` and in that Jenkins folder it will creates the workspace and in that Job1 folder created and inside the job1 its available main.txt which is fetched from master branch into the github.

```
ubuntu@Jenkins-Master: ~      X  ubuntu@Jenkins-Slave: ~/jen  X  +  v
ubuntu@Jenkins-Slave:~$ ls
agent.jar  jenkins
ubuntu@Jenkins-Slave:~$ cd /home/ubuntu/jenkins/workspace/Job1
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$ ls
main.txt
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$
```

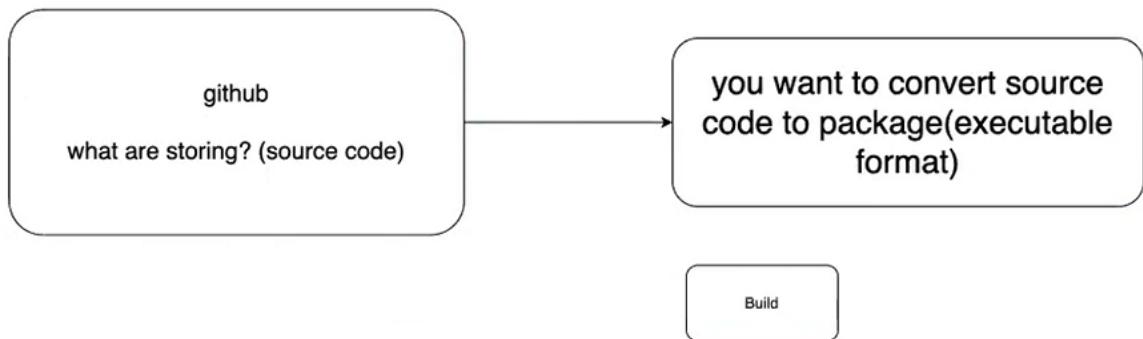
106. Architecture of DevOps, Instead of Doing Manual Work, we can do Automation with DevOps.



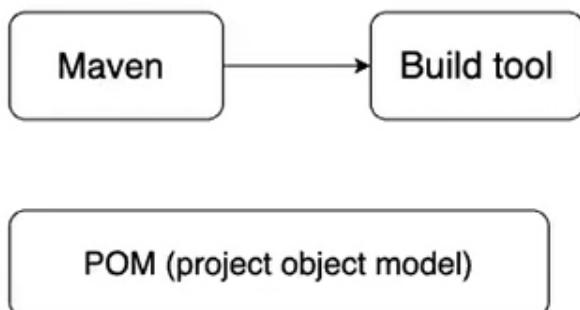
107. Agenda.

Agenda

1. What is Build?
2. Why maven & What does maven do??
3. What is maven plugin & Install Maven plugin?
4. What is POM file?
5. Create Job to build a software code with Maven
6. Build triggers (Job exploration)
6. Upstream and DownstreamJob



108. Maven is Build Tool In the Github > we get Source code which is not executable, so we use maven tool to convert into executable format.



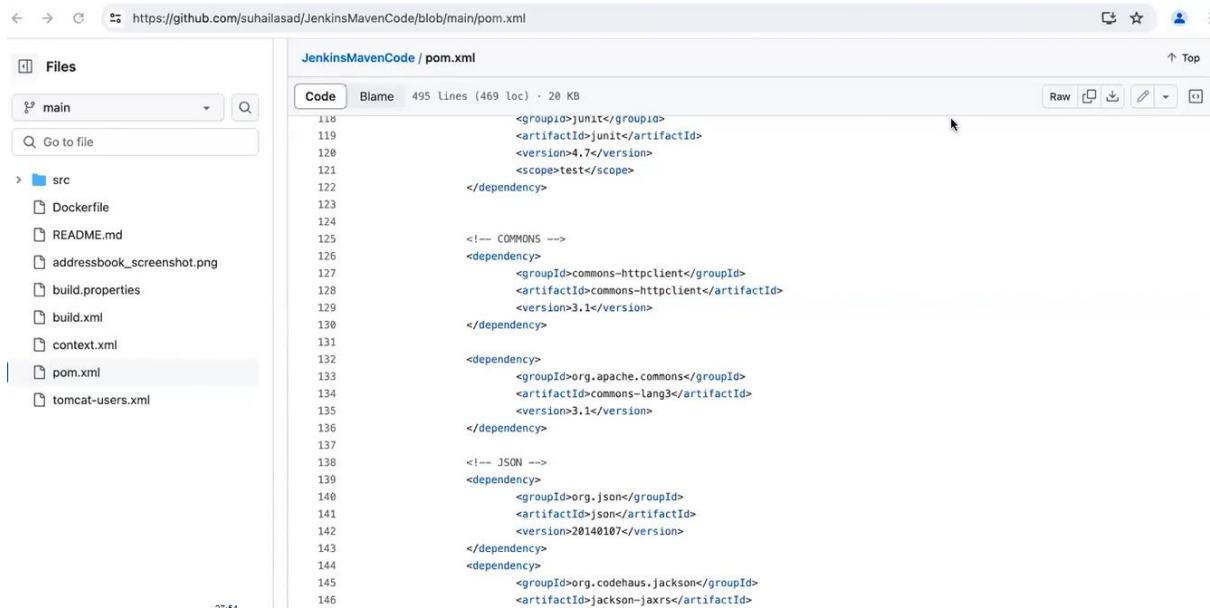
109. This is Suhailasad Maven github Repository.

The screenshot shows a GitHub repository page. At the top, there are navigation links: back, forward, search, and a URL bar showing <https://github.com/suhailasad/JenkinsMavencode>. Below the URL bar are tabs for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The 'Code' tab is selected. On the left, there's a sidebar with a dropdown for branches ('main'), a link to 4 Branches, and a link to 6 Tags. A search bar says 'Go to file'. On the right, there's a green button labeled '<> Code'. The main area displays a list of files and their commit history:

File	Commit Message	Date
src	initial commit	8 years ago
Dockerfile	Dockerfile updated	9 months ago
README.md	README.md	2 months ago
addressbook_screenshot.png	initial commit	8 years ago
build.properties	initial commit	8 years ago
build.xml	initial commit	8 years ago
context.xml	Dockerfile updated	9 months ago
pom.xml	initial commit	8 years ago
tomcat-users.xml	Dockerfile updated	9 months ago

The file "pom.xml" is highlighted with a yellow box.

110. This is Inside the pom.xml file, POM is an acronym for Project Object Model. The pom.xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc. Maven reads the pom.



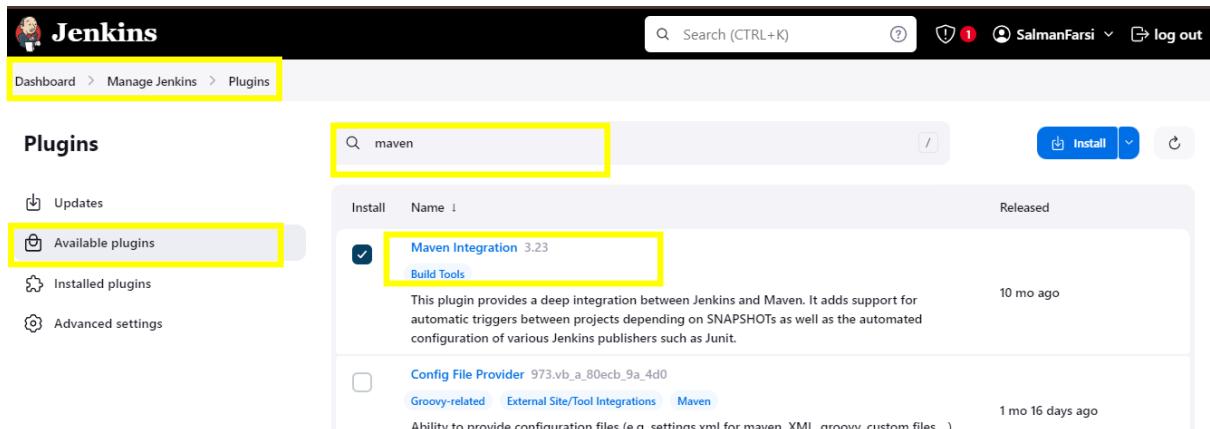
The screenshot shows a GitHub page for the repository 'JenkinsMavenCode'. The URL is https://github.com/suhailasad/JenkinsMavenCode/blob/main/pom.xml. The page displays the contents of the pom.xml file. The code is as follows:

```
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.7</version>
<scope>test</scope>
</dependency>
<!-- COMMONS -->
<dependency>
    <groupId>commons-httpclient</groupId>
    <artifactId>commons-httpclient</artifactId>
    <version>3.1</version>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.1</version>
</dependency>
<!-- JSON -->
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20140107</version>
</dependency>
<dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-jaxrs</artifactId>

```

MAVEN:

111. Installing the Maven Plugin



The screenshot shows the Jenkins Plugins page. The URL is https://jenkins.example.com/plugins. The search bar at the top has 'maven' typed into it. The 'Available plugins' tab is selected. A yellow box highlights the 'Available plugins' tab. A yellow box also highlights the 'Maven Integration' plugin entry in the list. The 'Maven Integration' plugin is version 3.23, released 10 months ago. It provides a deep integration between Jenkins and Maven. The 'Install' button is visible next to the plugin entry.

112. Installed Successfully.

The screenshot shows the Jenkins 'Plugins' page under 'Manage Jenkins'. On the left, there's a sidebar with links like 'Updates', 'Available plugins', 'Installed plugins', 'Advanced settings', and 'Download progress'. The 'Download progress' link is highlighted with a grey box. The main area is titled 'Download progress' and shows a 'Preparation' section with three items: 'Checking internet connectivity', 'Checking update center connectivity', and 'Success'. Below this is a table with four rows: 'Javadoc' (Success), 'JSch dependency' (Success), 'Maven Integration' (Success), and 'Loading plugin extensions' (Success). At the bottom, there are two buttons: a blue 'Go back to the top page' button and a grey 'Restart Jenkins when installation is complete and no jobs are running' button.

113. Click On Tools

The screenshot shows the 'Manage Jenkins' page. The left sidebar has links for 'New Item', 'Build History', 'Manage Jenkins' (which is highlighted with a yellow box), and 'My Views'. The main content area is titled 'System Configuration' and contains several sections: 'System' (Configure global settings and paths), 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), 'Tools' (Configure tools, their locations and automatic installers, which is highlighted with a yellow box), 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), and 'Appearance' (Configure the look and feel of Jenkins). A note at the top right says: 'Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See [the documentation](#).'. There are 'Manage' and 'Dismiss' buttons next to the note.

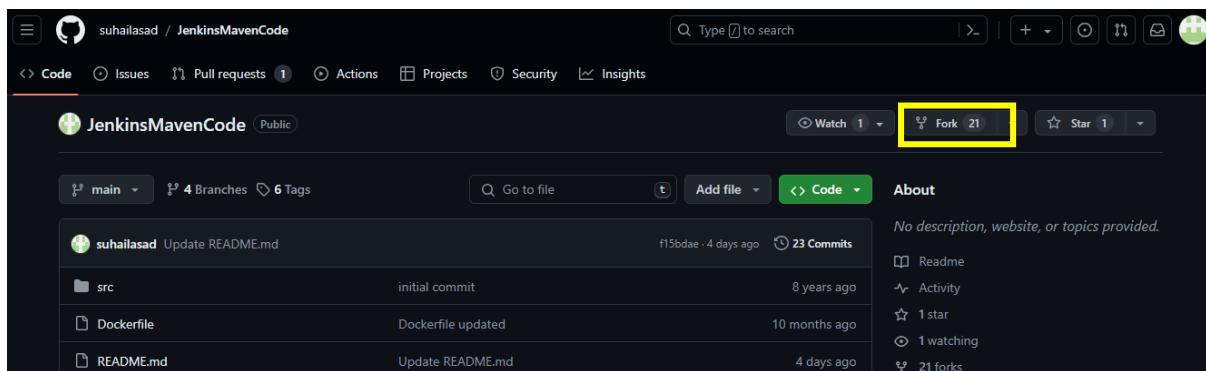
114. Maven > given name as Maven and version and save it.

The screenshot shows the 'Tools' configuration page for 'Maven'. A large yellow box highlights the 'Maven' section. Inside, the 'Name' field is set to 'maven' and the 'Install automatically' checkbox is checked. Under 'Install from Apache', the 'Version' dropdown is set to '3.9.7'. At the bottom, there are 'Add Maven' and 'Save' buttons, with 'Save' being highlighted by a blue box.

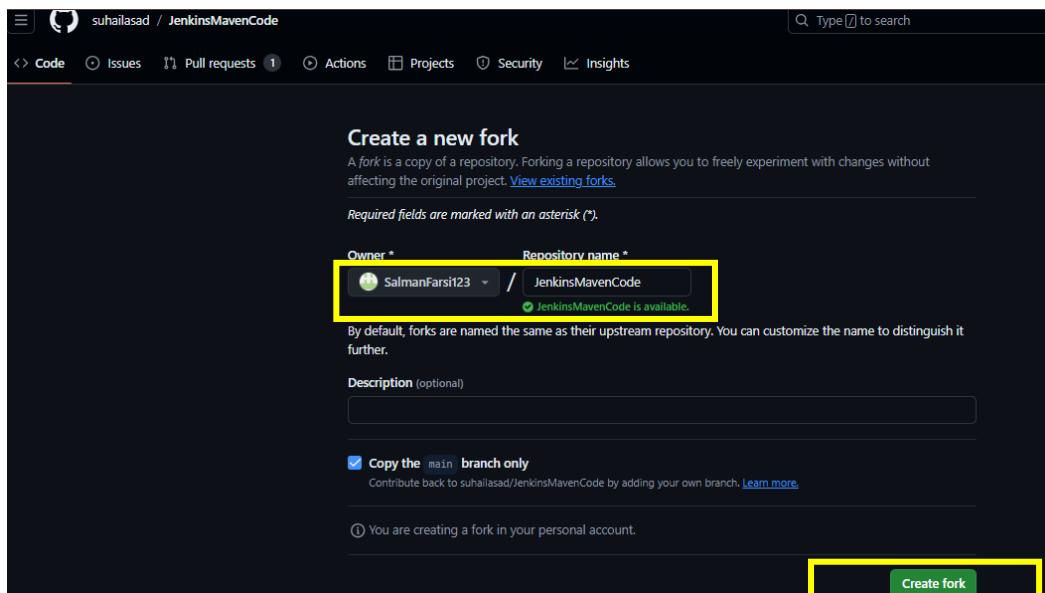
115. This is Suhailasad JenkinsMavenCode repository.



116. Click On Fork



117. Given Repository Name as ur wish.



118. This My Git-Hub Repository Fork the Main Branch only.

The screenshot shows a GitHub repository page for 'JenkinsMavenCode'. The repository is public and was forked from 'suhaillasad/JenkinsMavenCode'. The main branch is highlighted with a yellow box. The repository contains one branch ('main') and no tags. The code listing shows several files including Dockerfile, README.md, and build.xml. On the right side, there's a 'Clone' section with options for HTTPS, SSH, and GitHub CLI, and a download ZIP button. The repository has 0 stars, 0 forks, and 0 releases. It also has 0 packages published.

119. I came to Jenkins and In the SCM I am Pasting my git hub repo and Fetching the data from main branch

The screenshot shows the Jenkins configuration for a job named 'Job1'. Under the 'Source Code Management' section, the 'Repository URL' is set to 'https://github.com/SalmanFarsi123/JenkinsMavenCode.git'. The 'Branch Specifier' is set to '*/main'. The 'Save' button is visible at the bottom.

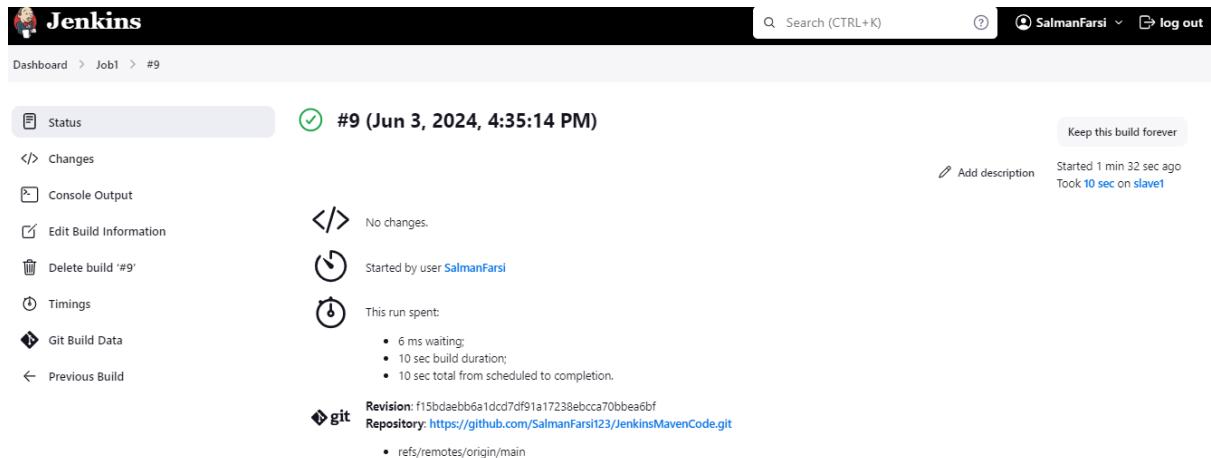
120. In Build Environment > Invoke top-level Maven targets

The screenshot shows the Jenkins job configuration interface. The left sidebar lists several sections: General, Source Code Management, Build Triggers, Build Environment (which is highlighted with a yellow box), Build Steps, and Post-build Actions. The main area is titled 'Configure' and contains a 'Build Steps' section. A dropdown menu titled 'Add build step ^' is open, showing various options like Execute Windows batch command, Execute shell, Invoke Ant, Invoke Gradle script, and 'Invoke top-level Maven targets', which is also highlighted with a yellow box.

121. Or Go to Build steps, Maven version > given named as maven and Goals Clean Install and save it.

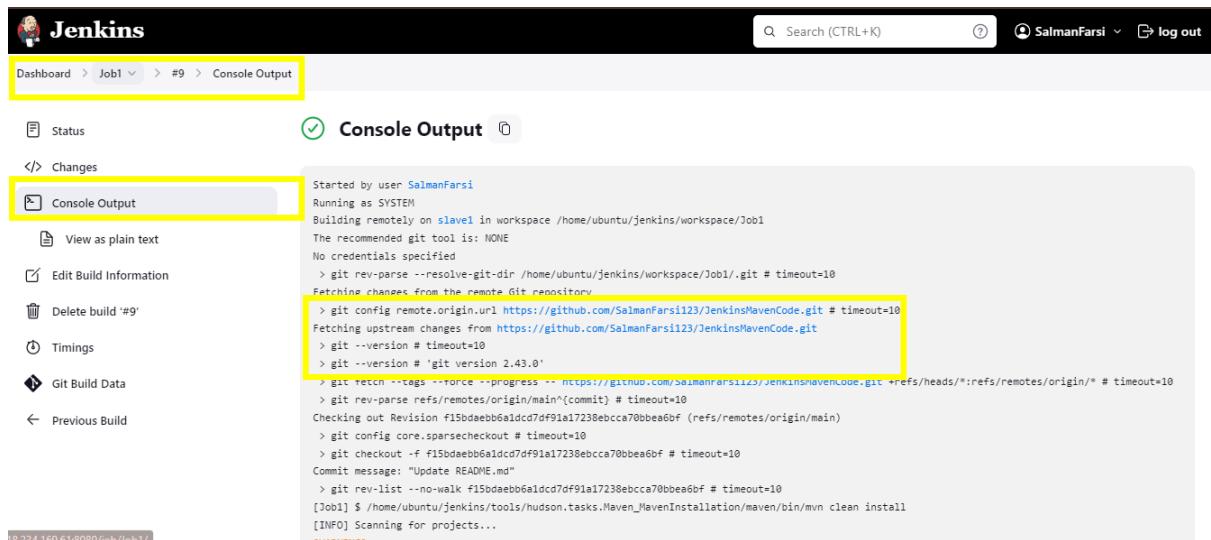
The screenshot shows the Jenkins job configuration interface. The left sidebar lists several sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps (which is highlighted with a yellow box), and Post-build Actions. The main area is titled 'Configure' and contains a 'Build Steps' section. A single step is defined, labeled 'Invoke top-level Maven targets'. Underneath it, the 'Maven Version' is set to 'maven' and the 'Goals' are set to 'clean install'. At the bottom of the screen, there are 'Save' and 'Apply' buttons.

122. And Build Now and its ran successfully



The screenshot shows the Jenkins interface for Job1 build #9. The status is green with a checkmark, indicating success. The build was started by user SalmanFarsi at 4:35:14 PM on Jun 3, 2024. It took 10 seconds on a slave. The build summary includes a link to the GitHub repository and the commit hash f15bdabbb6a1dc7df91a17238ebcca70bbea6bf. The console output section is collapsed.

123. Now Its fetching the data from github repo. And after that maven installed



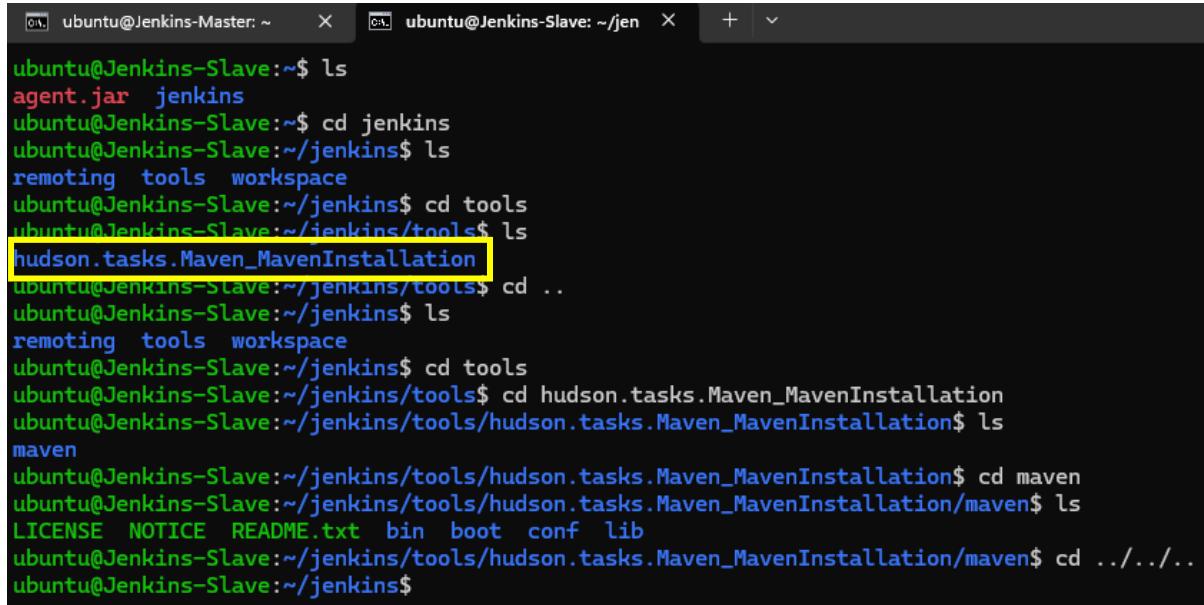
The screenshot shows the Jenkins interface for Job1 build #9. The status is green with a checkmark. The build was started by user SalmanFarsi. The console output shows the execution of git commands to fetch from the GitHub repository and the execution of Maven commands to install the project. The output ends with the command [INFO] Scanning for projects..

124. Here we can see success and created one new folder that target and that is executable format.

```
Tests run: 23, Failures: 0, Errors: 0, Skipped: 0

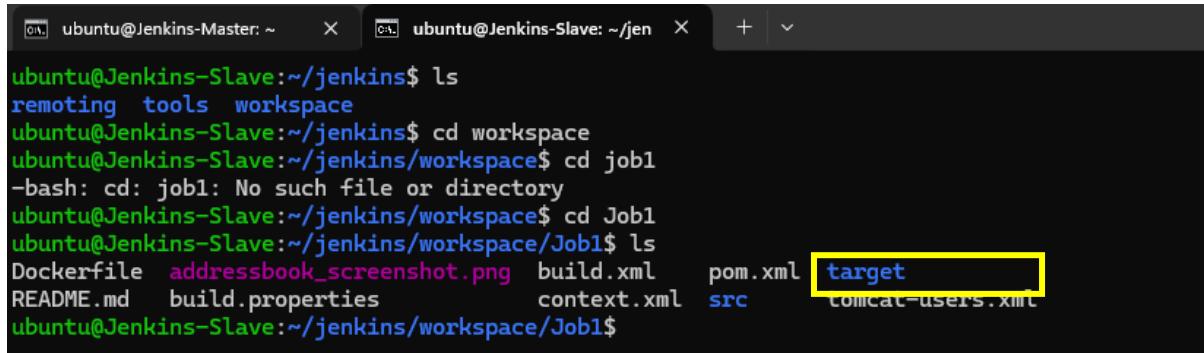
[INFO]
[INFO] --- war:3.4.0:war (default-war) @ addressbook ---
[INFO] Packaging webapp
[INFO] Assembling webapp [addressbook] in [/home/ubuntu/jenkins/workspace/Job1/target/addressbook]
[INFO] Processing war project
[INFO] Building war: /home/ubuntu/jenkins/workspace/Job1/target/addressbook.war
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ addressbook ---
[INFO] Installing /home/ubuntu/jenkins/workspace/Job1/pom.xml to /home/ubuntu/.m2/repository/com/edurekademo/tutorial/addressbook/2.0/addressbook-2.0.pom
[INFO] Installing /home/ubuntu/jenkins/workspace/Job1/target/addressbook.war to
/home/ubuntu/.m2/repository/com/edurekademo/tutorial/addressbook/2.0/addressbook-2.0.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  5.628 s
[INFO] Finished at: 2024-06-03T16:39:36Z
[INFO] -----
Finished: SUCCESS
```

125. we can see in the tools maven installed automatically.



```
ubuntu@Jenkins-Slave:~$ ls
agent.jar jenkins
ubuntu@Jenkins-Slave:~$ cd jenkins
ubuntu@Jenkins-Slave:~/jenkins$ ls
remoting tools workspace
ubuntu@Jenkins-Slave:~/jenkins$ cd tools
ubuntu@Jenkins-Slave:~/jenkins/tools$ ls
hudson.tasks.Maven_MavenInstallation
ubuntu@Jenkins-Slave:~/jenkins/tools$ cd ..
ubuntu@Jenkins-Slave:~/jenkins$ ls
remoting tools workspace
ubuntu@Jenkins-Slave:~/jenkins$ cd tools
ubuntu@Jenkins-Slave:~/jenkins/tools$ cd hudson.tasks.Maven_MavenInstallation
ubuntu@Jenkins-Slave:~/jenkins/tools/hudson.tasks.Maven_MavenInstallation$ ls
maven
ubuntu@Jenkins-Slave:~/jenkins/tools/hudson.tasks.Maven_MavenInstallation$ cd maven
ubuntu@Jenkins-Slave:~/jenkins/tools/hudson.tasks.Maven_MavenInstallation/maven$ ls
LICENSE NOTICE README.txt bin boot conf lib
ubuntu@Jenkins-Slave:~/jenkins/tools/hudson.tasks.Maven_MavenInstallation/maven$ cd ../../..
ubuntu@Jenkins-Slave:~/jenkins$
```

126. we can see in the workspace its cloned git hub data and newly its created target folder/directory.



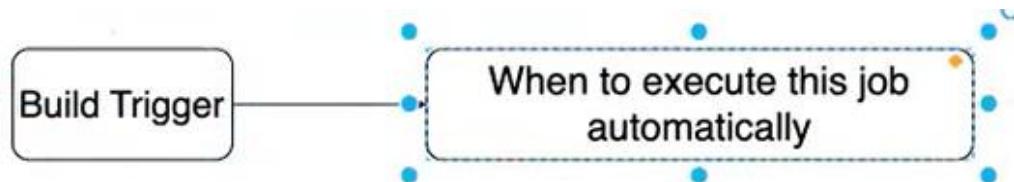
```
ubuntu@Jenkins-Slave:~/jenkins$ ls
remoting tools workspace
ubuntu@Jenkins-Slave:~/jenkins$ cd workspace
ubuntu@Jenkins-Slave:~/jenkins/workspace$ cd job1
-bash: cd: job1: No such file or directory
ubuntu@Jenkins-Slave:~/jenkins/workspace$ cd Job1
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$ ls
Dockerfile addressbook_screenshot.png build.xml pom.xml target
README.md build.properties context.xml src comcat-users.xml
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$
```

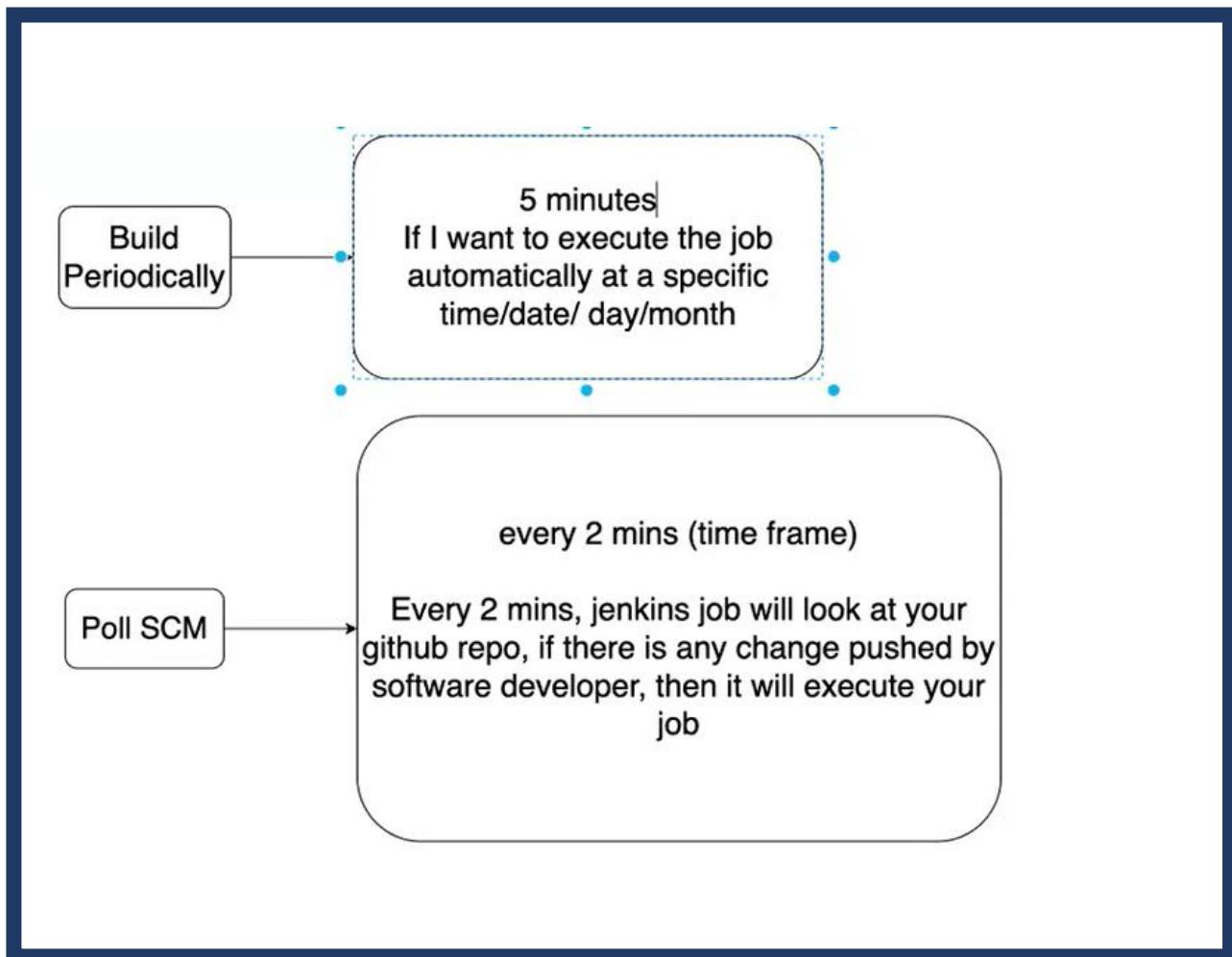
127. we can see those files in git-hub but not the target folder because that maven plugin build that target folder

128. See the within the target directory we can see the addressbook.war file

```
ubuntu@Jenkins-Slave:~/jenkins$ ls
remoting tools workspace
ubuntu@Jenkins-Slave:~/jenkins$ cd workspace
ubuntu@Jenkins-Slave:~/jenkins/workspace$ cd job1
-bash: cd: job1: No such file or directory
ubuntu@Jenkins-Slave:~/jenkins/workspace$ cd Job1
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$ ls
Dockerfile addressbook_screenshot.png build.xml pom.xml target
README.md build.properties context.xml src tomcat-users.xml
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1$ cd target
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1/target$ ls
addressbook classes generated-test-sources maven-status test-classes
addressbook.war generated-sources maven-archiver surefire-reports
ubuntu@Jenkins-Slave:~/jenkins/workspace/Job1/target$
```

129. Build trigger and its automatically execute the job automatically





130. Build Periodically means

Using "Build Trigger" Option:

Go to your Jenkins job configuration page.

Navigate to the "Build Triggers" section.

Select the checkbox next to "Build periodically".

In the schedule field, enter the cron expression defining your desired periodicity.

Cron Expression Examples:

Every hour: H * * * *

Every 20 minutes: H/20 * * * *

Daily at 8:30 AM: 30 8 * * *

Every weekday at noon: 0 12 * * 1-5

More complex schedules: Refer to cron documentation for detailed examples

Dashboard > job1 > Configuration

Configure

Build Triggers

Trigger builds remotely (e.g., from scripts) ?
 Build after other projects are built ?
 Build periodically 

Provides a cron -like feature to periodically execute this project. This feature is primarily for using Jenkins as a cron replacement, and it is **not ideal for continuously building software projects**. When people first start continuous integration, they are often so used to the idea of regularly scheduled builds like nightly/weekly that they use this feature. However, the point of continuous integration is to start a build as soon as a change is made, to provide a quick feedback to the change. To do that you need to [hook up SCM change notification to Jenkins](#).

So, before using this feature, stop and ask yourself if this is really what you want.

GitHub hook trigger for GITScm polling ?
 Poll SCM ?

Dashboard > job1 > Configuration

Configure

Build Triggers

Build periodically 

No schedules so will never run

This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:

MINUTE HOUR DOM MONTH DOW

MINUTE Minutes within the hour (0–59)
HOUR The hour of the day (0–23)
DOM The day of the month (1–31)
MONTH The month (1–12)
DOW The day of the week (0–7) where 0 and 7 are Sunday.

To specify multiple values for one field, the following operators are available. In the order of precedence,

- * specifies all valid values 
- M-N specifies a range of values
- M-N/X or */X steps by intervals of X through the specified range or whole valid range
- A,B,...,Z enumerates multiple values

← → ⌛ https://crontab.guru/#*_*_*_*_*

crontab guru

The quick and simple editor for cron schedule expressions by [Cronitor](#)

"At every minute."

next at 2024-05-03 08:08:00 random

*	*	*	*	*
minute	hour	day (month)	month	day (week)

* any value
,

value list separator
- range of values
/ step values
0-23 allowed values

The quick and simple editor for cron schedule expressions by [Cronitor](#)

"At every minute past hour 8."

next at 2024-05-03 08:08:00

* 8 * * *

minute	hour	day	month	day
		(month)		(week)
*	8	*	*	*

* any value
 , value list separator
 - range of values
 / step values
 0-23 allowed values

131. **GitHub hook trigger:** When you push code changes to your GitHub repository, it sends a webhook notification to Jenkins, triggering a build immediately (overrides polling interval).

Configure

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 ?

Build Environment

Delete workspace before build starts

suhailasad / JenkinsMavenCode Public

< Code Issues Pull requests Actions Projects Security Insights

main 4 Branches 6 Tags Go to file Code

About

No description, website, or topics provided.

Readme Activity 0 stars 1 watching 15 forks Report repository

Releases

6 tags

Packages

No packages published

Languages

README

Configure

General

Source Code Management

Build Triggers

Poll SCM ?

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

Build after other projects are built ?

Build periodically ?

GitHub hook trigger for GITScm polling ?

Build Environment

Build Steps

Post-build Actions

← → ⌛ https://crontab.guru/#*_*_16_*_*_4

crontab guru

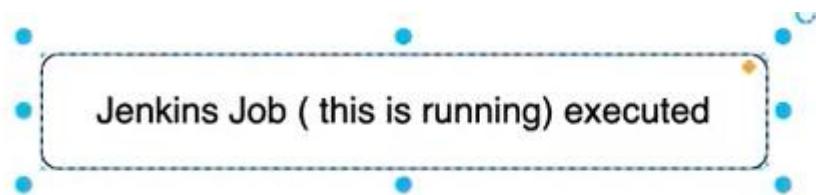
The quick and simple editor for cron schedule expressions by [Cronitor](#)

“At every minute past hour 16 on Thursday.”

next at 2024-05-09 16:00:00 random

*/16 * * * 4

minute	hour	day	month	day
			(month)	(week)
*	16	*	*	4
* any value				
, value list				
- separator				
- range of values				
/ step values				
@yearly (non-standard)				
@annually (non-standard)				
@monthly (non-standard)				
@weekly (non-standard)				
@daily (non-standard)				
@hourly (non-standard)				
@reboot (non-standard)				



after build (package) will deployed inside kubernetes cluster (software execution)

UpstreamJob (Job which is independent)

DownstreamJob (Job which is dependent)

CI

Task1 Job (successfully)

CD

Task2 Job (only then Task2 will execute automatically)

132. Currently we have One Job1 and Now creating another job.

The screenshot shows the Jenkins dashboard. On the left, there are links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are sections for 'Build Queue' (empty) and 'Build Executor Status' (one idle node). In the center, a table displays a single job entry: 'Job1' with a green checkmark icon, last success at '2 days 19 hr #10', and a duration of '8.8 sec'. A 'More' button is to the right of the table. At the top right, there's a search bar, user info, and a 'log out' link.

133. Named as DownstreamJob > Freestyle project > Ok

The screenshot shows the 'Enter an item name' dialog with 'DownstreamJob' entered. Below it, three project types are listed: 'Freestyle project' (selected), 'Maven project', and 'Pipeline'. Each has a brief description and an 'OK' button. The Jenkins header and sidebar are visible.

134. General > Description > Downstream Job

The screenshot shows the 'General' configuration page for the 'DownstreamJob' project. The 'Configure' sidebar on the left lists 'General', 'Source Code Management', 'Build Triggers', and 'Build Environment'. The main area shows the 'General' tab selected, with a 'Description' field containing 'Downstream Job'. A 'Enabled' checkbox is checked. The Jenkins header and sidebar are visible.

135. Build Steps > echo "Downstream Job Successfully done"

The screenshot shows the Jenkins 'Configuration' page for a job named 'DownstreamJob'. The left sidebar lists several configuration sections: General, Source Code Management, Build Triggers, Build Environment, **Build Steps**, and Post-build Actions. The 'Build Steps' section is highlighted with a yellow box. On the right, under 'Build Steps', there is a single step named 'Execute shell'. The 'Command' field contains the text 'echo "Downstream Job Successfully done"'. There is also a link to 'See the list of available environment variables'. At the bottom of the 'Build Steps' panel are 'Save' and 'Apply' buttons.

2 Ways to trigger the
Downstream Job

Either we do the configuration
of triggering DownstreamJob
in
UpstreamJob or
DownstreamJob

136. I wrote something in the description

The screenshot shows the Jenkins configuration interface for a job named 'Job1'. The 'General' tab is selected. In the 'Description' field, the text 'The Purpose of this Job Knowing About Upstream' is entered. A yellow box highlights this description field.

137. SCM > Git > Paste your github repo

The screenshot shows the Jenkins configuration interface for a job named 'Job1'. The 'Source Code Management' section is expanded, and the 'Git' option is selected. Under 'Repositories', the 'Repository URL' field contains the value 'https://github.com/SalmanFarsi123/JenkinsMavenCode.git', which is highlighted with a yellow box.

138. Build Steps > echo "upstreamJob"

The screenshot shows the Jenkins configuration interface for a job named 'Job1'. The 'Build Steps' section is expanded, showing a single 'Execute shell' step. The 'Command' field contains the text 'echo "UpstreamJob"', which is highlighted with a yellow box.

139. I renamed that Job1 to upstreamJob

The screenshot shows the Jenkins 'Rename Project Job1' dialog. On the left, there's a sidebar with options: Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename (which is highlighted with a yellow box). The main area has a 'New Name' input field containing 'UpstreamJob' (also highlighted with a yellow box), and a 'Rename' button below it.

140. Renamed Successfully.

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with New Item, Build History (which is highlighted with a yellow box), Manage Jenkins, My Views, Build Queue (with a note: 'No builds in the queue.'), Build Executor Status, and Built-In Node. The main area shows a 'Build History' table with two entries: 'DownstreamJob' and 'UpstreamJob'. 'UpstreamJob' is highlighted with a yellow box. The table columns include S (Status), W (Workstation), Name, Last Success, Last Failure, and Last Duration. The 'UpstreamJob' entry shows a green checkmark icon, a sun icon, and '2 days 20 hr #10' under Last Success.

S	W	Name	Last Success	Last Failure	Last Duration
(..)	sun	DownstreamJob	N/A	N/A	N/A
(green checkmark)	sun	UpstreamJob	2 days 20 hr #10	N/A	8.8 sec

141. Go to UpstreamJob > Configuration > Post-Build Actions > Build other projects > select downstreamjob

The screenshot shows the Jenkins configuration interface for an 'UpstreamJob'. The left sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Post-build Actions' section is highlighted with a yellow box. On the right, under 'Post-build Actions', there is a sub-section titled 'Build other projects' with a question mark icon. It contains a 'Projects to build' field containing 'DownstreamJob,' which is also highlighted with a yellow box. Below this, a red error message says 'No such project 'down'. Did you mean 'DownstreamJob'?' There are three radio button options for triggering: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. At the bottom right are 'Save' and 'Apply' buttons.

142. This DownstreamJob

The screenshot shows the Jenkins status page for a 'DownstreamJob'. The top navigation bar shows 'Dashboard > DownstreamJob >'. The left sidebar has links: Status (highlighted with a yellow box), Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area is titled 'DownstreamJob' and describes it as a 'Downstream Job'. It shows an 'Upstream Projects' section with a green checkmark next to 'UpstreamJob'. At the bottom are 'Permalinks' and 'Build History' (with a sun icon) and a 'trend' dropdown.

143. Build Now > its successfully ran

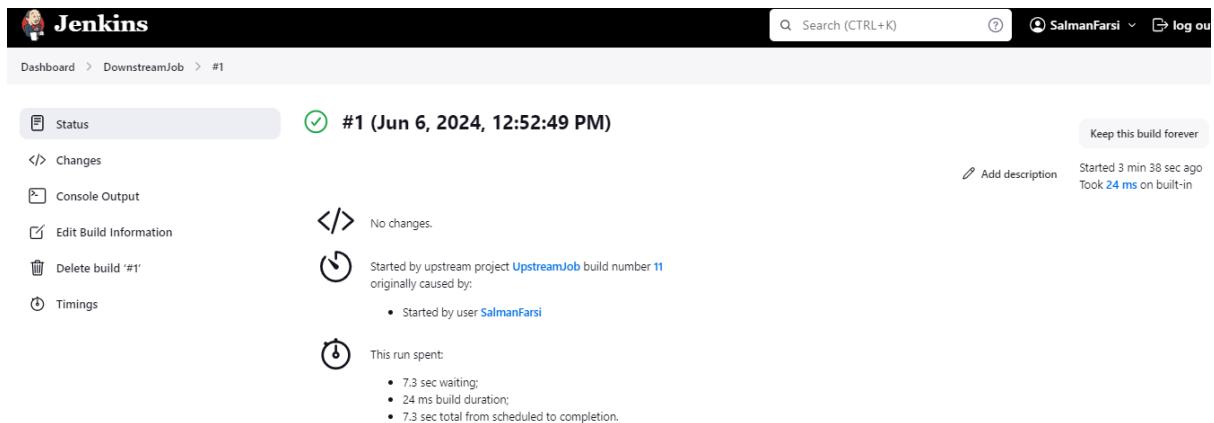
The screenshot shows the Jenkins interface for the 'UpstreamJob' configuration. The main title is 'UpstreamJob' with a green checkmark icon. Below it, a subtitle reads 'The Purpose of this Job Knowing About Upstream'. On the left, there's a sidebar with links: Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. To the right, under 'Downstream Projects', there's a link to 'DownstreamJob'. A section titled 'Permalinks' lists four build entries: 'Last build (#10), 2 days 20 hr ago', 'Last stable build (#10), 2 days 20 hr ago', 'Last successful build (#10), 2 days 20 hr ago', and 'Last completed build (#10), 2 days 20 hr ago'. A yellow box highlights the build history table, specifically the first two entries.

- Last build (#10), 2 days 20 hr ago
- Last stable build (#10), 2 days 20 hr ago
- Last successful build (#10), 2 days 20 hr ago
- Last completed build (#10), 2 days 20 hr ago

144. Like DownstreamJob ran

The screenshot shows the Jenkins interface for the 'DownstreamJob' configuration. The main title is 'DownstreamJob' with a green checkmark icon. Below it, a subtitle reads 'Downstream Job'. On the left, there's a sidebar with links: Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. To the right, under 'Upstream Projects', there's a link to 'UpstreamJob'. A section titled 'Permalinks' lists one build entry: '#1 Jun 6, 2024, 12:52 PM'. A yellow box highlights this single build entry in the build history table.

145. status



The screenshot shows the Jenkins job status page for 'DownstreamJob #1'. The main title is '#1 (Jun 6, 2024, 12:52:49 PM)'. On the left, there's a sidebar with links: Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#1', and Timings. The 'Console Output' section shows the following log:

```
</> No changes.  
Started by upstream project UpstreamJob build number 11  
originally caused by:

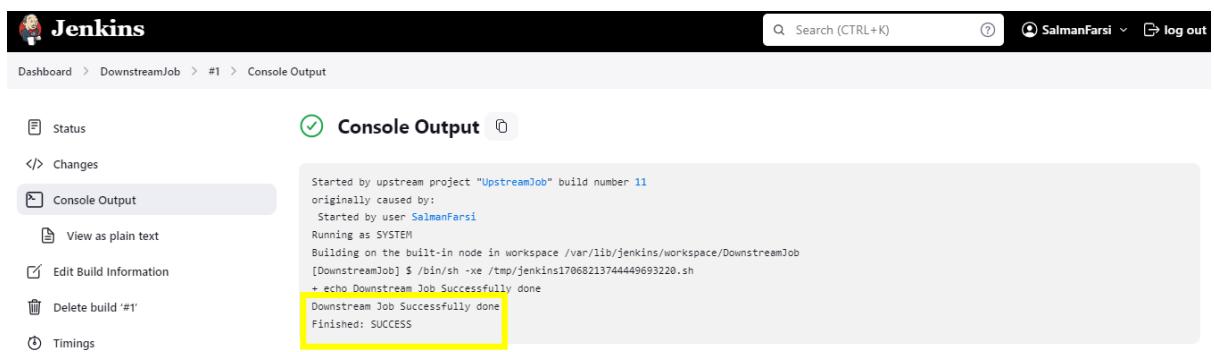
- Started by user SalmanFarsi

This run spent:

- 7.3 sec waiting;
- 24 ms build duration;
- 7.3 sec total from scheduled to completion.

```

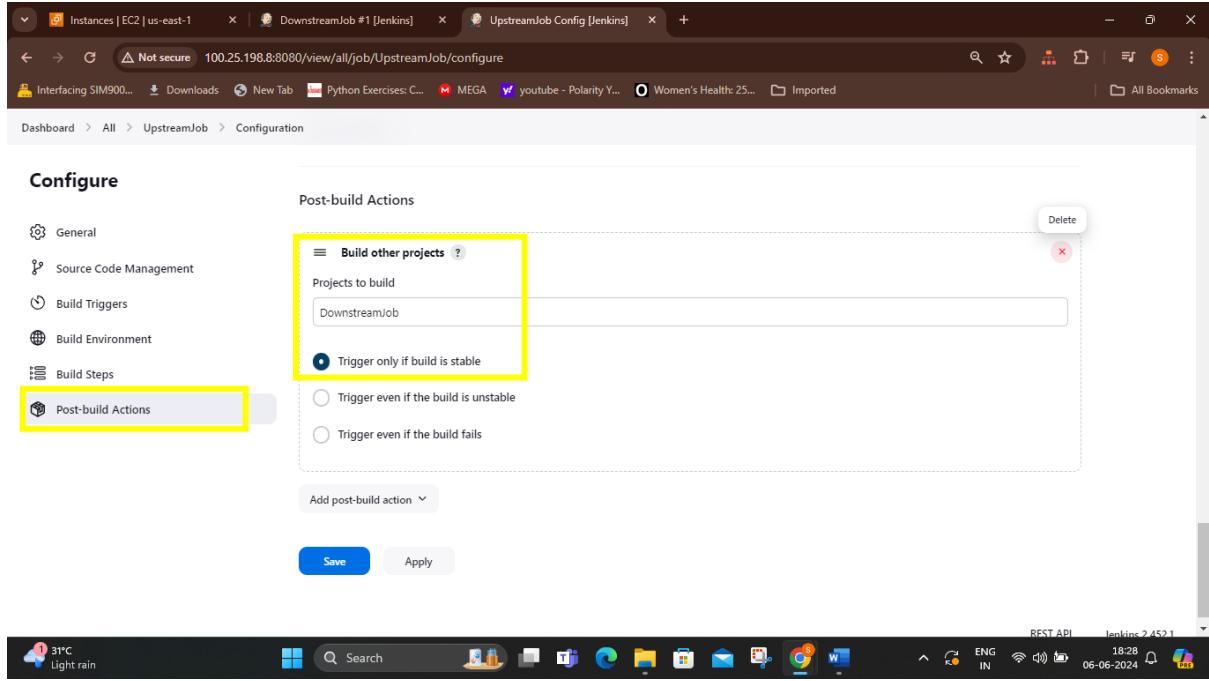
146. Echo it print downstream job and this 1st method.



The screenshot shows the Jenkins console output page for 'DownstreamJob #1'. The title is 'Console Output'. The log output is:

```
Started by upstream project "UpstreamJob" build number 11  
originally caused by:  
Started by user SalmanFarsi  
Running as SYSTEM  
Building on the built-in node in workspace /var/lib/jenkins/workspace/DownstreamJob  
[DownstreamJob] $ /bin/sh -xe /tmp/jenkins17068213744449693220.sh  
+ echo Downstream Job Successfully done  
Downstream Job Successfully done  
Finished: SUCCESS
```

147. This 2nd Method Go to UpstreamJob > Post-Build Actions > Build other projects and select downstreamJob



The screenshot shows the Jenkins configuration page for 'UpstreamJob'. The left sidebar has sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions (which is selected). The main area shows 'Post-build Actions' with one entry highlighted:

Build other projects

Projects to build: DownstreamJob

Trigger only if build is stable (radio button selected)

Other options: Trigger even if the build is unstable, Trigger even if the build fails

Buttons at the bottom: Save, Apply, and a 'Delete' button.

148. In the DownstreamJob go to SCM > Build After other Projects are built and select UpstreamJob.

The screenshot shows the 'Configure' screen for a 'DownstreamJob'. On the left, there's a sidebar with options: General, Source Code Management (highlighted with a yellow box), Build Triggers, Build Environment, Build Steps, and Post-build Actions. Under 'Source Code Management', there's a 'Build Triggers' section. It contains a checkbox for 'Build after other projects are built' which is checked. Below it is a 'Projects to watch' input field containing 'up' and 'UpstreamJob' (highlighted with a yellow box). There are also four radio button options: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', 'Trigger even if the build fails', and 'Always trigger, even if the build is aborted'. At the bottom are 'Save' and 'Apply' buttons.

149. Build Now UpstreamJob

The screenshot shows the 'UpstreamJob' project page. On the left, there's a sidebar with options: Status (highlighted with a yellow box), Changes, Workspace, Build Now (highlighted with a yellow box), Configure, Delete Project, and Rename. The main area has a green checkmark icon and the title 'UpstreamJob'. Below it is a subtitle 'The Purpose of this Job Knowing About Upstream'. To the right are 'Edit description' and 'Disable Project' buttons. A 'Downstream Projects' section shows a green circle with a checkmark and 'DownstreamJob'. A 'Permalinks' section lists recent builds: 'Last build (#11), 9 min 35 sec ago', 'Last stable build (#11), 9 min 35 sec ago', 'Last successful build (#11), 9 min 35 sec ago', and 'Last completed build (#11), 9 min 35 sec ago'. At the bottom is a 'Build History' table with columns 'Build #', 'Timestamp', and 'Status'.

150. its ran downstreamJob too.

The screenshot shows the Jenkins interface for a 'DownstreamJob' project. At the top, there's a navigation bar with a user icon, search bar, and log out link. Below it is a breadcrumb trail: Dashboard > DownstreamJob. A yellow box highlights this breadcrumb trail. On the left, a sidebar lists project actions: Status, Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main content area has a title 'DownstreamJob' with a green checkmark icon. It includes a 'Permalinks' section with a bulleted list of recent builds. Below that is a 'Build History' section with a table showing two builds: '#2' (Jun 6, 2024, 1:02 PM) and '#1' (Jun 6, 2024, 12:52 PM). A yellow box highlights the table row for build '#2'. There are also 'Edit description' and 'Disable Project' buttons on the right.

151. DownstreamJob status

The screenshot shows the status page for build '#2' of the 'DownstreamJob'. The top navigation bar and breadcrumb trail are identical to the previous screenshot. The main content area shows the build number '#2 (Jun 6, 2024, 1:02:53 PM)' with a green checkmark icon. To the right are buttons for 'Keep this build forever', 'Add description', and status information ('Started 2 min 28 sec ago', 'Took 21 ms on built-in'). On the left, a sidebar lists actions: Status, Changes, Console Output, Edit Build Information, Delete build '#2', Timings, and Previous Build. The 'Console Output' item is highlighted with a yellow box. The main content area displays the build log, which starts with 'Started by upstream project "UpstreamJob" build number 12 originally caused by: Started by user SalmanFarsi'. It then shows the command being run: 'Running as SYSTEM Building on the built-in node in workspace /var/lib/jenkins/workspace/DownstreamJob [DownstreamJob] \$ /bin/sh -xe /tmp/jenkins7611715843329947703.sh + echo Downstream Job Successfully done'. A yellow box highlights the output 'Downstream Job Successfully done'. Below that, it says 'Finished: SUCCESS'.

152. Console Output of DownstreamJob

The screenshot shows the 'Console Output' page for build '#2'. The top navigation bar and breadcrumb trail are identical. The main content area shows the build number '#2' with a green checkmark icon. The 'Console Output' item in the sidebar is highlighted with a yellow box. The main content area displays the build log, which starts with 'Started by upstream project "UpstreamJob" build number 12 originally caused by: Started by user SalmanFarsi'. It then shows the command being run: 'Running as SYSTEM Building on the built-in node in workspace /var/lib/jenkins/workspace/DownstreamJob [DownstreamJob] \$ /bin/sh -xe /tmp/jenkins7611715843329947703.sh + echo Downstream Job Successfully done'. A yellow box highlights the output 'Downstream Job Successfully done'. Below that, it says 'Finished: SUCCESS'.