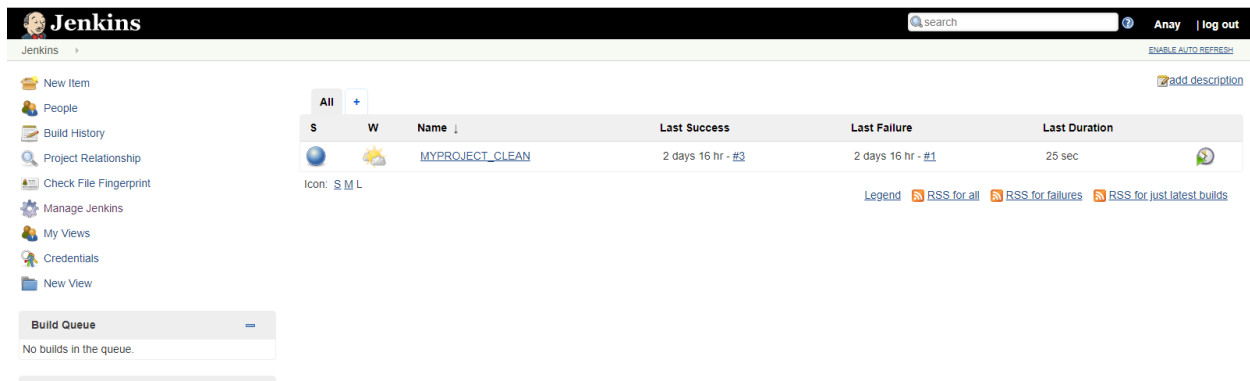


MASTER-SLAVE Configuration for Jenkins.

Step1: Install Jenkins in a host “Master_host”.

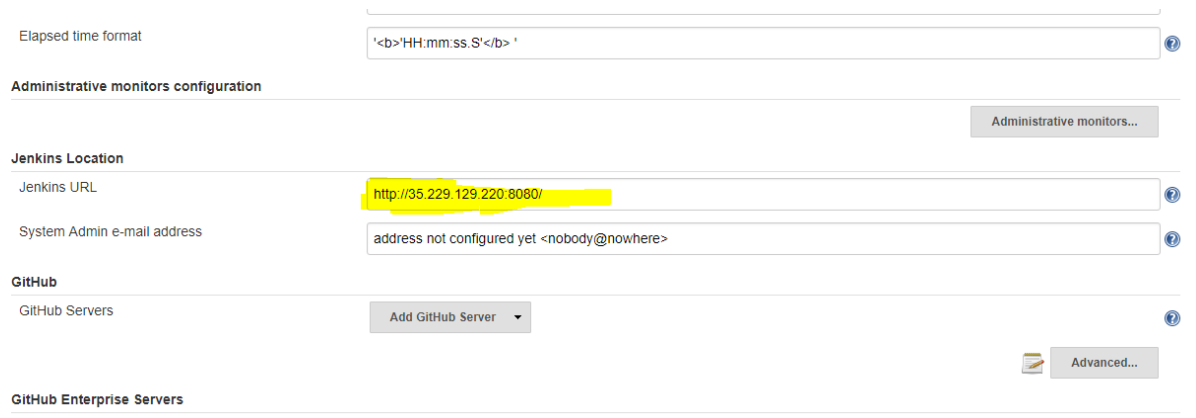


The screenshot shows the Jenkins dashboard. On the left is a sidebar with navigation links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Credentials, and New View. Below these is a 'Build Queue' section indicating 'No builds in the queue.' The main area displays a table of builds. The table has columns for 'S' (Status), 'W' (Workspace), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. One build is listed with the name 'MYPROJECT_CLEAN', a success status, and a duration of 25 seconds. Below the table are links for 'Icon: S M L' and 'Legend' with RSS feeds for all, failures, and just latest builds. At the top right, there is a search bar, 'Any', 'log out', and an 'ENABLE AUTO REFRESH' link.

S	W	Name	Last Success	Last Failure	Last Duration
		MYPROJECT_CLEAN	2 days 16 hr - #3	2 days 16 hr - #1	25 sec

Step2: Navigate to manage Jenkins → configure system and check the configuration on tab Jenkins location.

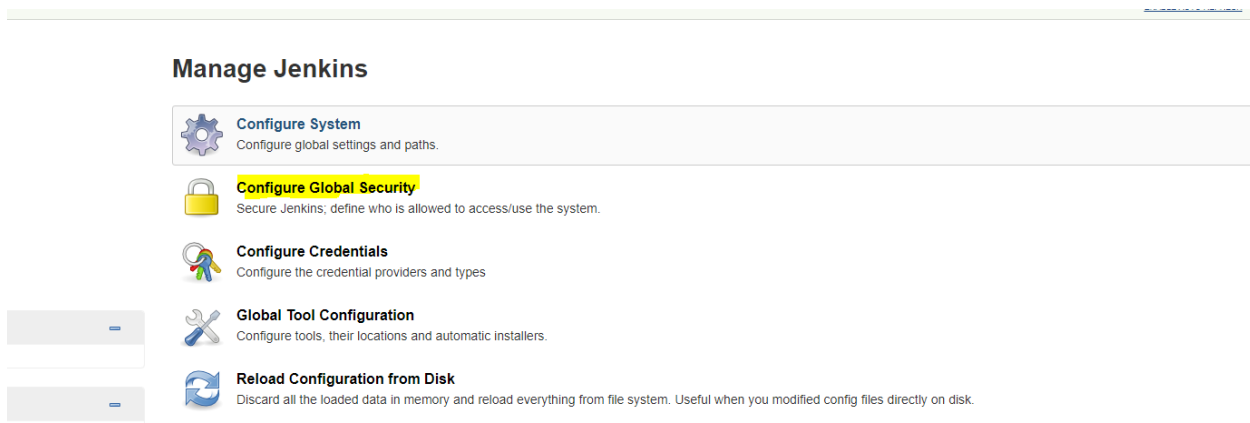
We should have entered our Jenkins URL



The screenshot shows the 'Configure System' page in Jenkins. It includes several configuration sections: 'Elapsed time format' with a text input field containing 'HH:mm:ss.S'; 'Administrative monitors configuration' with a button 'Administrative monitors...'; 'Jenkins Location' with 'Jenkins URL' set to 'http://35.229.129.220:8080/' and 'System Admin e-mail address' set to 'address not configured yet <nobody@nowhere>'; 'GitHub' with a button 'Add GitHub Server'; and 'GitHub Enterprise Servers'. An 'Advanced...' button is visible at the bottom right.

Step3: we will configure or global security settings.

Navigate to manage Jenkins → global security



The screenshot shows the 'Manage Jenkins' page. It features a list of configuration options, each with an icon and a description: 'Configure System' (gear icon) for global settings and paths; 'Configure Global Security' (lock icon) for securing Jenkins and defining access; 'Configure Credentials' (key icon) for credential providers and types; 'Global Tool Configuration' (wrench icon) for tool locations and installers; and 'Reload Configuration from Disk' (refresh icon) for reloading config files. The 'Configure Global Security' option is highlighted with a yellow background.

Next, click on this and scroll down to Agent tab and select random if it is disabled.

Markup Formatter

Markup Formatter

Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.

Agents









TCP port for JNLP agents ☐ Fixed : ☒ Random ☐ Disable

CSRF Protection

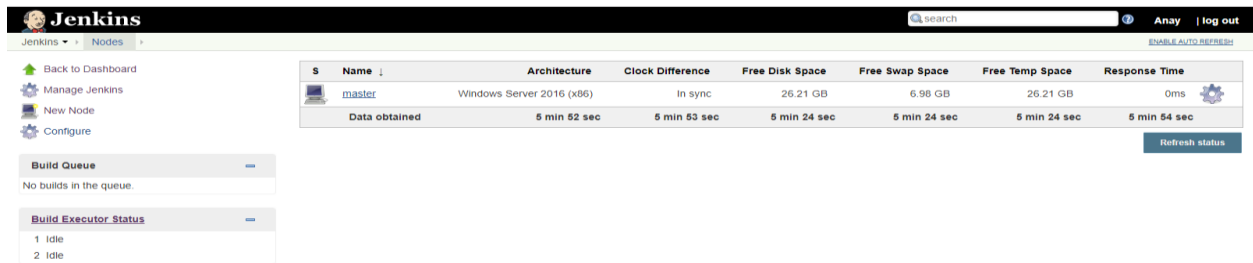
☒ Prevent Cross Site Request Forgery exploits

Crumbs **Crumb Algorithm**

Step4: Navigate to manage Jenkins → Manage Nodes

-  System log captures output from `java.util.logging` output related to Jenkins.
-  **Load Statistics**
Check your resource utilization and see if you need more computers for your builds.
-  **Jenkins CLI**
Access/manage Jenkins from your shell, or from your script.
-  **Script Console**
Executes arbitrary script for administration/trouble-shooting/diagnostics.
-  **Manage Nodes**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
-  **About Jenkins**
See the version and license information.
-  **Manage Old Data**
Scrub configuration files to remove remnants from old plugins and earlier versions.
-  **Manage Users**

Next, click on this and we will get master node as given below:



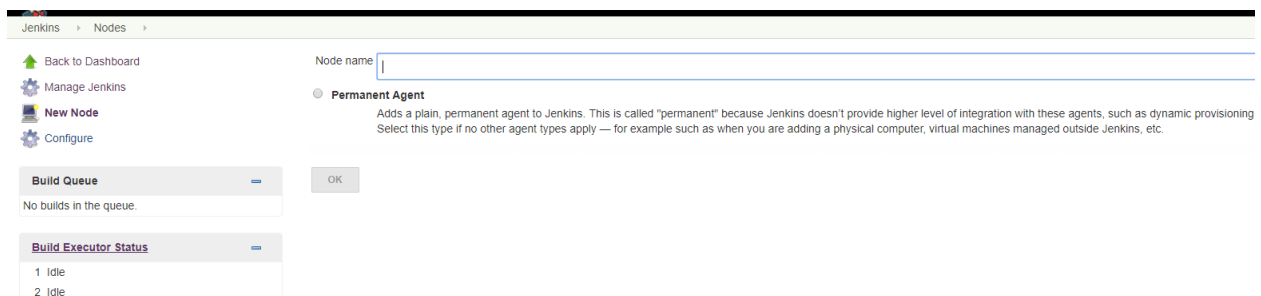
The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: Back to Dashboard, Manage Jenkins, New Node, and Configure. Below these are two panels: 'Build Queue' showing 'No builds in the queue' and 'Build Executor Status' showing two idle executors. The main area displays a table of nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	master	Windows Server 2016 (x86)	In sync	26.21 GB	6.98 GB	26.21 GB	0ms
Data obtained		5 min 52 sec	5 min 53 sec	5 min 24 sec	5 min 24 sec	5 min 24 sec	5 min 54 sec

At the bottom right of the table is a 'Refresh status' button.

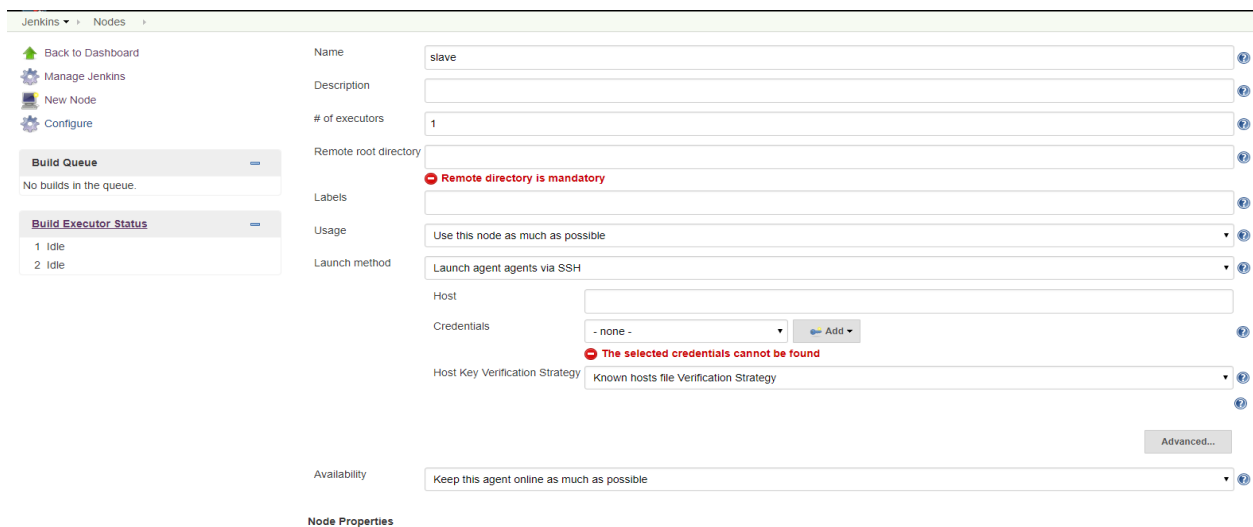
Step5: We will configure slave node and launch it.

Now, click on new node and we will get a page like this:



The screenshot shows the 'New Node' configuration page. The 'Node name' field is empty. Below it, the 'Permanent Agent' radio button is selected. A description explains that this is a plain, permanent agent. On the left sidebar, the 'Build Queue' and 'Build Executor Status' panels are visible. An 'OK' button is located next to the 'Node name' field.

Next, select permanent agent and give a name to agent and click on ok.



The screenshot shows the 'Configure' page for a new node named 'slave'. Fields include: Name (slave), Description, # of executors (1), Remote root directory (with an error message 'Remote directory is mandatory'), Labels, Usage (Use this node as much as possible), Launch method (Launch agent agents via SSH), Host, Credentials (set to '- none -' with an error message 'The selected credentials cannot be found'), Host Key Verification Strategy (Known hosts file Verification Strategy), and Availability (Keep this agent online as much as possible). There is an 'Advanced...' button at the bottom right.

Now, we need to have another virtual machine which we want to configure as node.

1. We must have a dir in slave node, So create a dir "C:\Jenkins"
2. Make sure we have opened our Jenkins master server in that slave node because later in this tutorial we need to download some software's.

Name	slave
Description	slave
# of executors	2
Remote root directory	C:\Jenkins
Labels	slave
Usage	Use this node as much as possible
Launch method	Launch agent via Java Web Start
Disable WorkDir	<input type="checkbox"/>
Custom WorkDir path	C:\Jenkins
Internal data directory	remoting
Fail if workspace is missing	<input type="checkbox"/>
Availability	Keep this agent online as much as possible

Node Properties

☐ Environment variables

Fill all the labels as given above and click on save.

ENABLE AUTO REFRESH

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Windows Server 2016 (x86)	In sync	26.21 GB	6.99 GB	26.21 GB	0ms
	slave		N/A	N/A	N/A	N/A	N/A
	Data obtained	15 ms	15 ms	15 ms	0 ms	0 ms	0 ms

Next, we need to launch this slave node to launch this we need to visit at <http://<masterurl>/computer/slave/> and we will get:

Agent slave (slave)

Connect agent to Jenkins one of these ways:

- Launch** Launch agent from browser
- Run from agent command line:

```
java -jar agent.jar -jnlpUrl http://35.229.129.220:8080/computer/slave/slave-agent.jnlp -secret 033fdf5bafc37720695b6ccc7af02d7468e85596e2246b3ecc71a8150946d508 -workDir "C:\Jenkins"
```

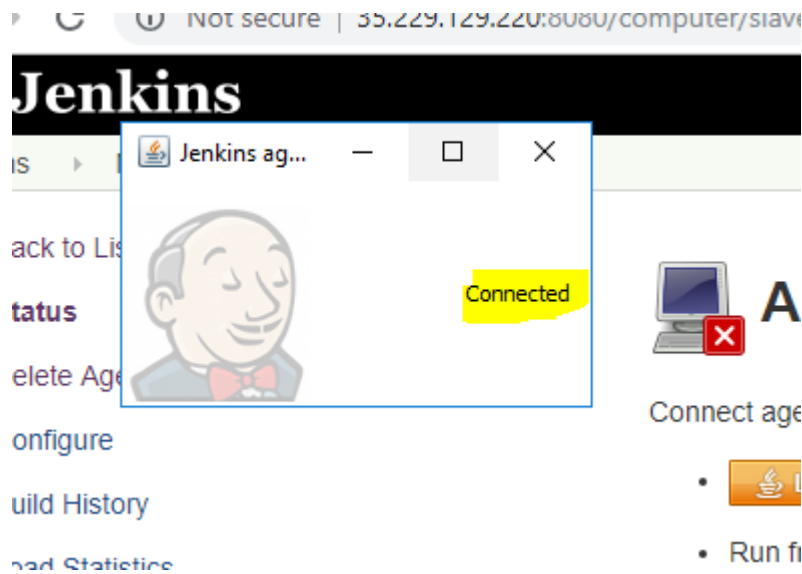
Projects tied to slave

None

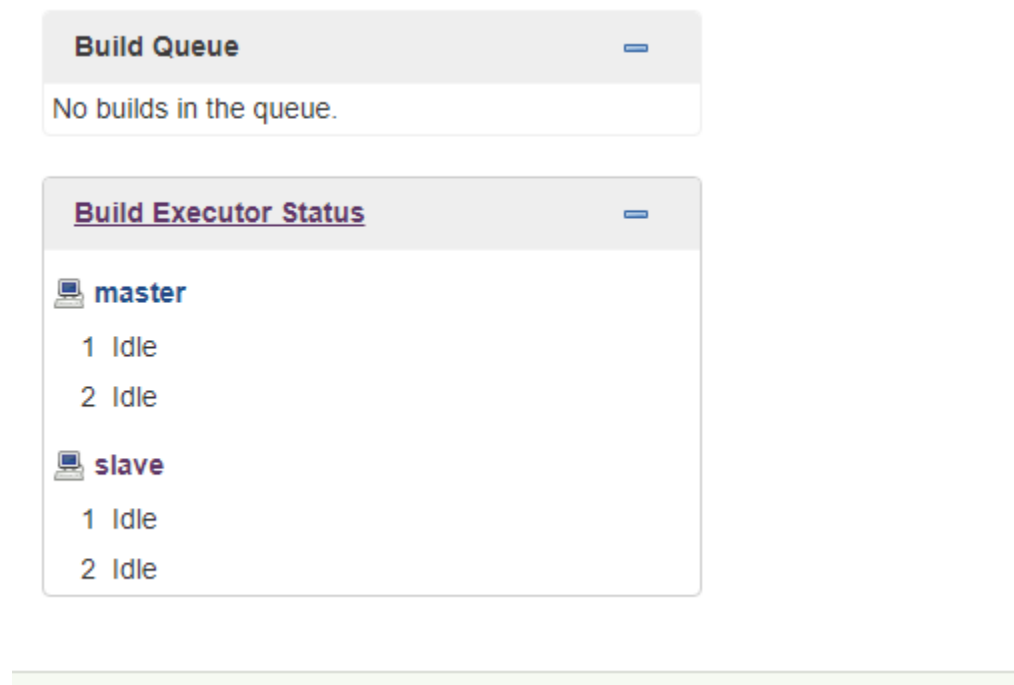
Now, click on launch and it will download a .jnlp file.

Note: The slave node must be having JDK.

Next, double click on that file and if we have jdk and the Dir that we have created previously, we will get the output as connected.



Now, refresh the master url and we can see that slave node is ready.



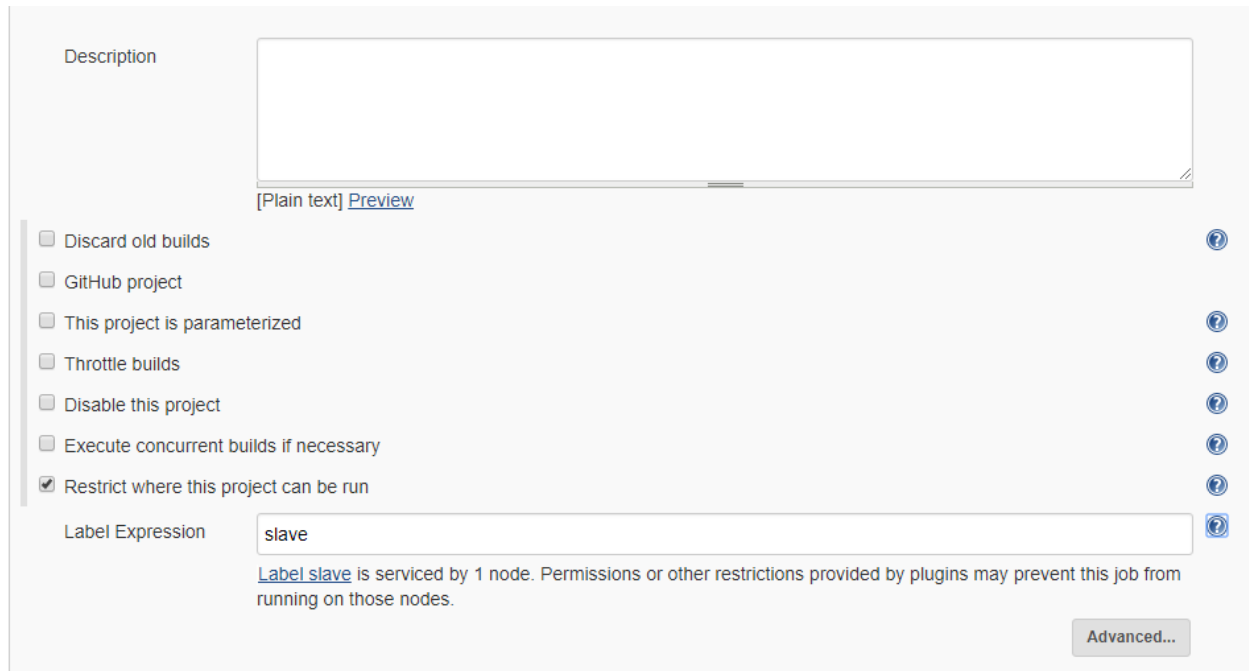
Example:

Now in this example we will restrict our Jenkins server to build a job on slave node only.

I'm assuming that you have gone through my pervious tutorials. And you are aware about how we can build a maven project in Jenkins.

We will be building a maven project. So, click on new item and configure all the steps.

General:



The image shows the 'General' tab of a Jenkins job configuration. It features a 'Description' text area at the top. Below it is a list of checkboxes for various options: 'Discard old builds', 'GitHub project', 'This project is parameterized', 'Throttle builds', 'Disable this project', 'Execute concurrent builds if necessary', and 'Restrict where this project can be run' (which is checked). To the right of these checkboxes are help icons. Below the checkboxes is a 'Label Expression' field containing the text 'slave'. A note below this field states: 'Label slave is serviced by 1 node. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.' At the bottom right is an 'Advanced...' button.

Description

[Plain text] [Preview](#)

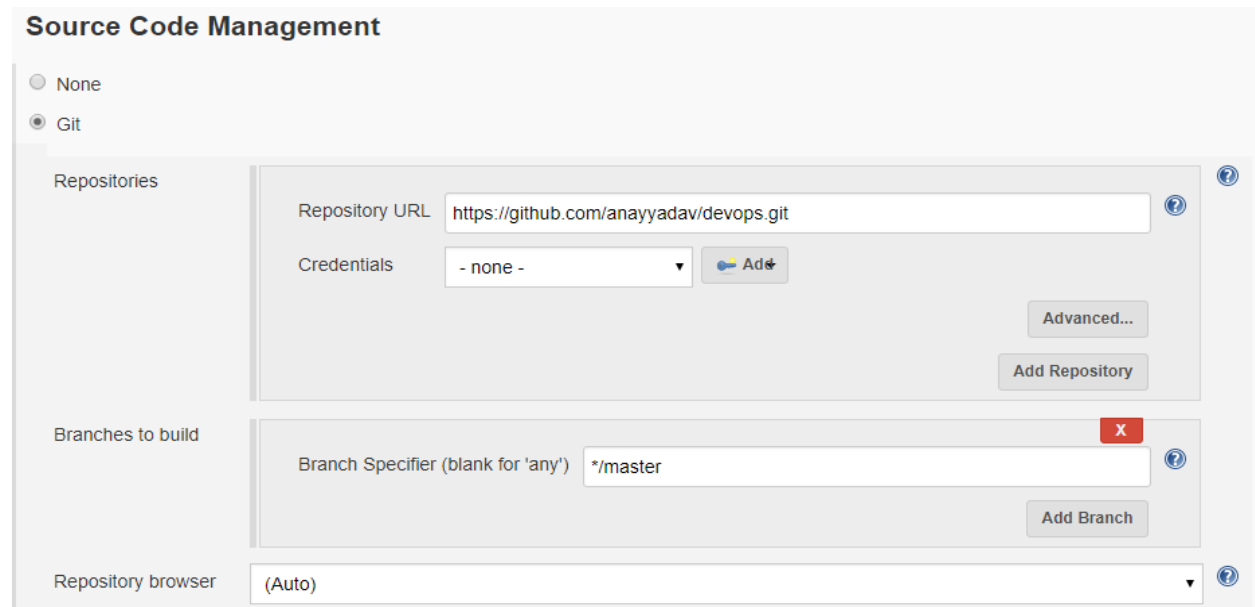
- ☐ Discard old builds
- ☐ GitHub project
- ☐ This project is parameterized
- ☐ Throttle builds
- ☐ Disable this project
- ☐ Execute concurrent builds if necessary
- ☒ Restrict where this project can be run

Label Expression:

[Label slave](#) is serviced by 1 node. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.

[Advanced...](#)

SCM:



The image shows the 'Source Code Management' section of a Jenkins job configuration. It has two radio buttons: 'None' and 'Git' (which is selected). Below this is the 'Repositories' section, which includes a 'Repository URL' field with the value 'https://github.com/anayyadav/devops.git', a 'Credentials' dropdown menu set to '- none -', and an 'Add' button. There are 'Advanced...' and 'Add Repository' buttons to the right. Below the 'Repositories' section is the 'Branches to build' section, which has a 'Branch Specifier (blank for \'any\')' field with the value '*/master' and an 'Add Branch' button. At the bottom is the 'Repository browser' dropdown menu, which is set to '(Auto)'. Help icons are present on the right side of the 'Repositories' and 'Branches to build' sections.

Source Code Management

☐ None
☒ Git

Repositories

Repository URL:

Credentials: [Add](#)

[Advanced...](#)
[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any'): [Add Branch](#)

Repository browser:

BUILD:

Pre Steps

Add pre-build step ▾

Build

Root POM

?

Goals and options

?

Advanced...

Finally save this job and click on build.

We can see that this process is running on slave.

Console Output

```
Started by user Anay.  
Building remotely on slave in workspace C:\Jenkins\workspace\SLAVE_NODE_TEST  
Unpacking https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.3.9/apache-maven-3.3.9-bin.zip to  
C:\Jenkins\tools\hudson.tasks.Maven_MavenInstallation\maven339 on slave
```

NOTE: Make sure one must be having Git on their slave node.

Verification:

We can also verify that this job is running on Slave node. To verify we can check the main page of Jenkins server.

New View

Build Queue

No builds in the queue.

Build Executor Status

master

1 Idle

2 Idle

slave

1 Idle

2 SLAVE_NODE_TEST #2

