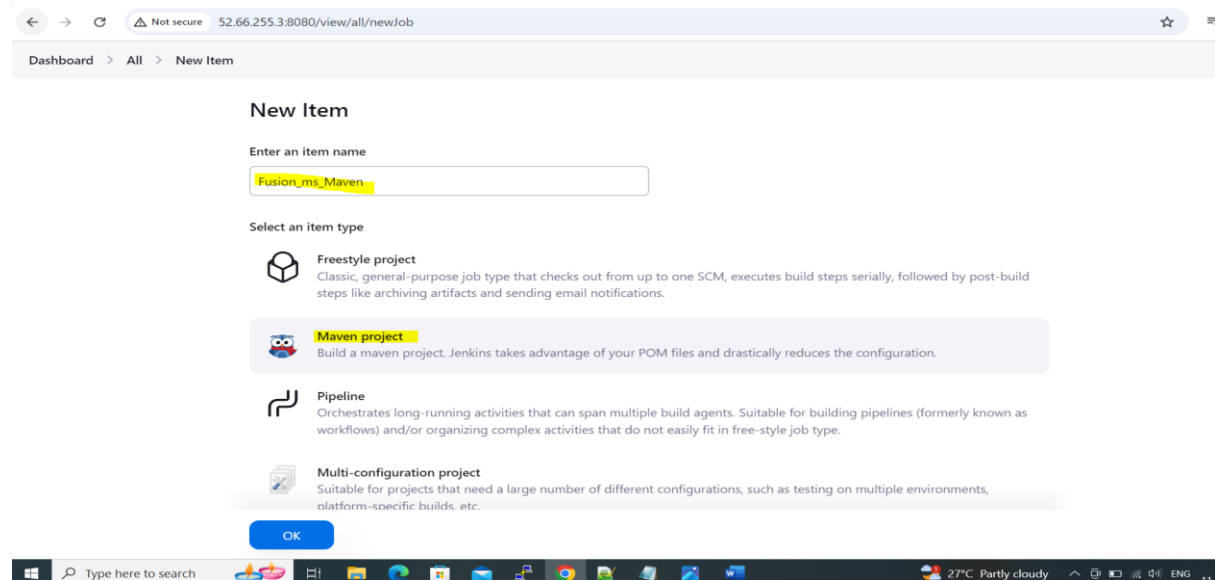
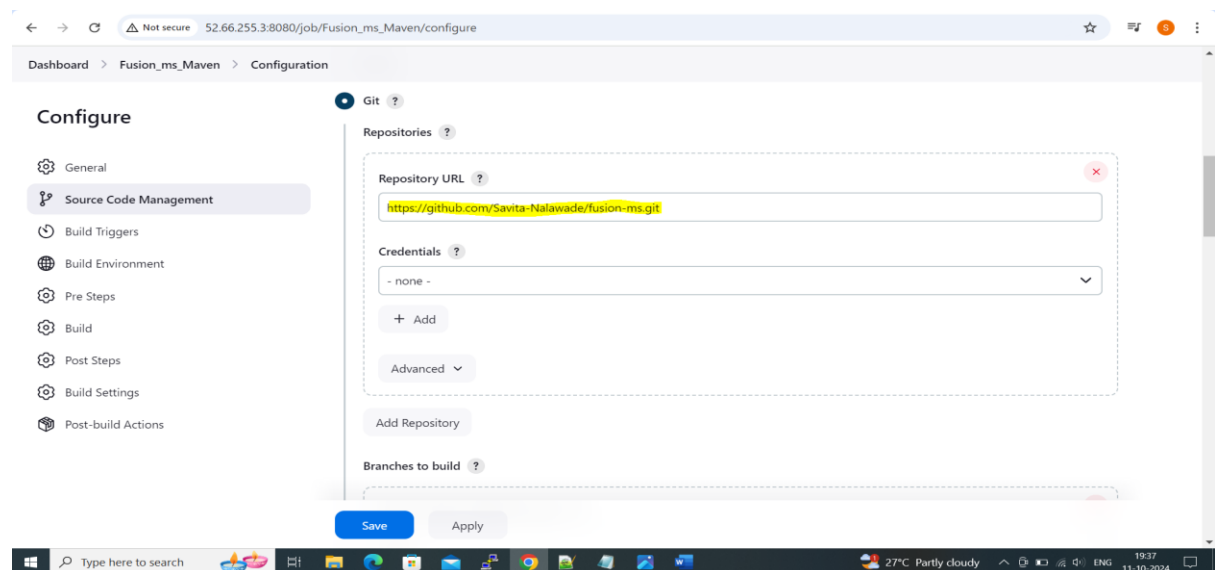


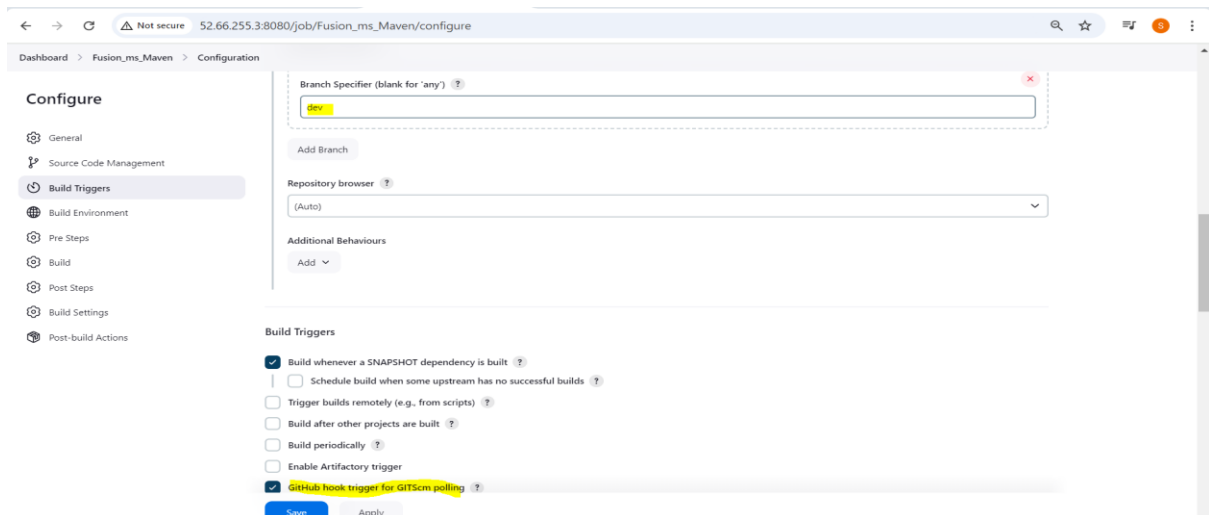
## 1)Create maven job(Fusion\_ms\_Maven) in jenkins



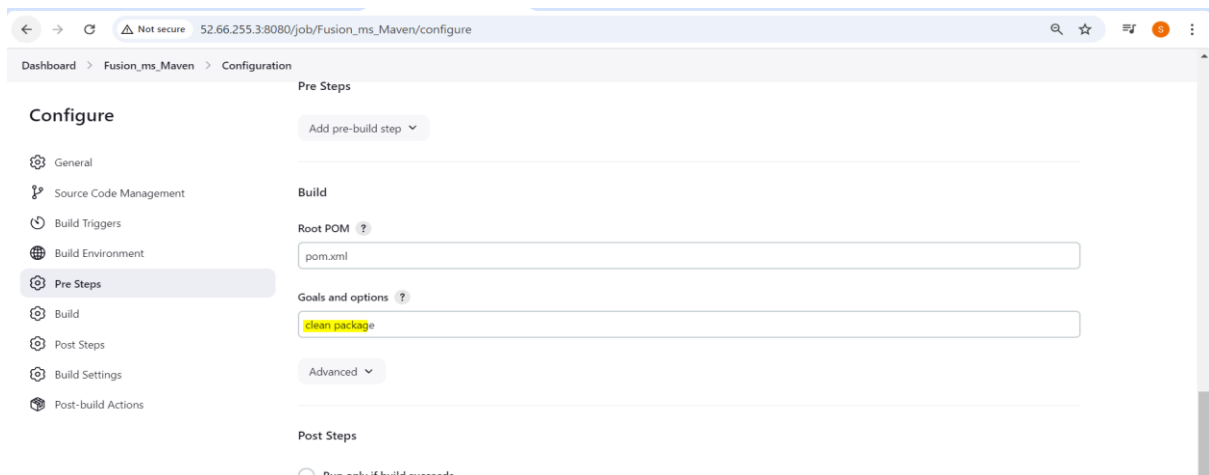
## 2)add url of your repository



3)provide branch (dev) it means when we do any changes in dev branch job will be run



4)give gols as “clean package” –it will compile and package code



5)Now, open IntelliJ and check in which branch we are currently move to dev branch

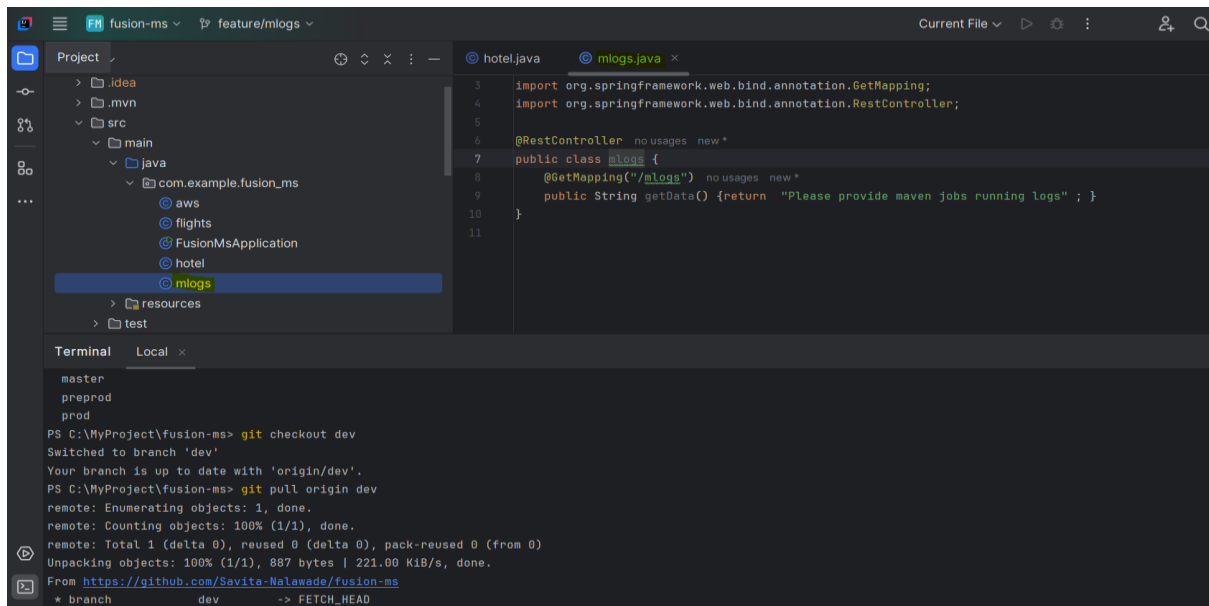
```

Terminal  Local x + v
PS C:\MyProject\fusion-ms> git branch
dev
* feature/aws
main
master
preprod
prod

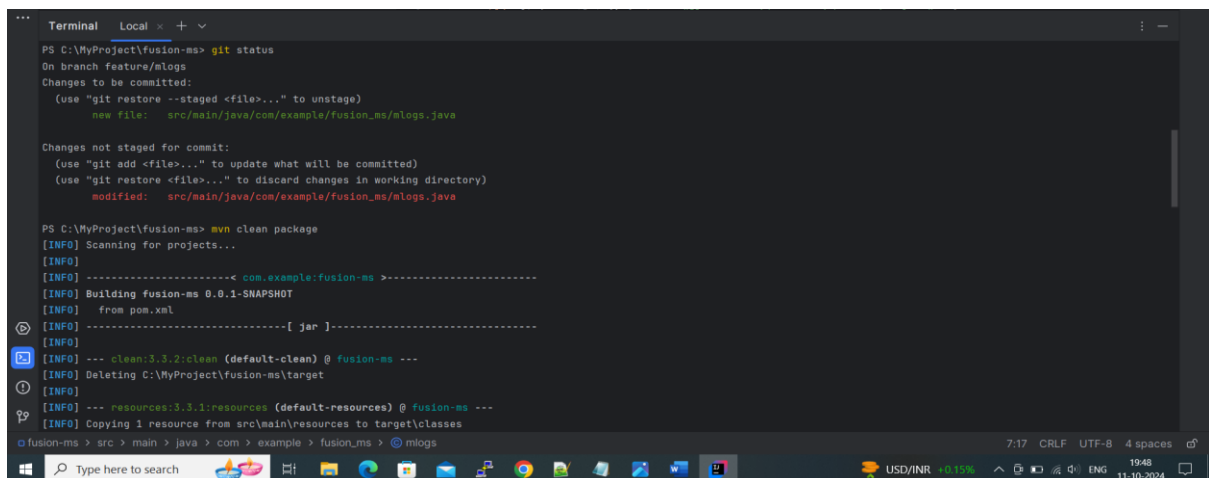
PS C:\MyProject\fusion-ms> git checkout dev
Switched to branch 'dev'
Your branch is up to date with 'origin/dev'.
PS C:\MyProject\fusion-ms> git pull origin dev
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (1/1), 887 bytes | 221.00 KiB/s, done.
From https://github.com/Savita-Nalawade/fusion-ms
* branch      dev      -> FETCH_HEAD
  715fe26..05726b2 dev    -> origin/dev
Updating 715fe26..05726b2
Fast-forward
 src/main/java/com/example/fusion_ms/aws.java | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 src/main/java/com/example/fusion_ms/aws.java
PS C:\MyProject\fusion-ms> git checkout -b feature/mlogs
Switched to a new branch 'feature/mlogs'
PS C:\MyProject\fusion-ms>

```

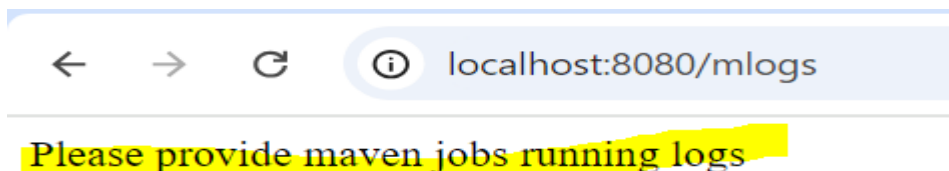
6) Create one new branch as “feature/mlogs” and create one java class for same



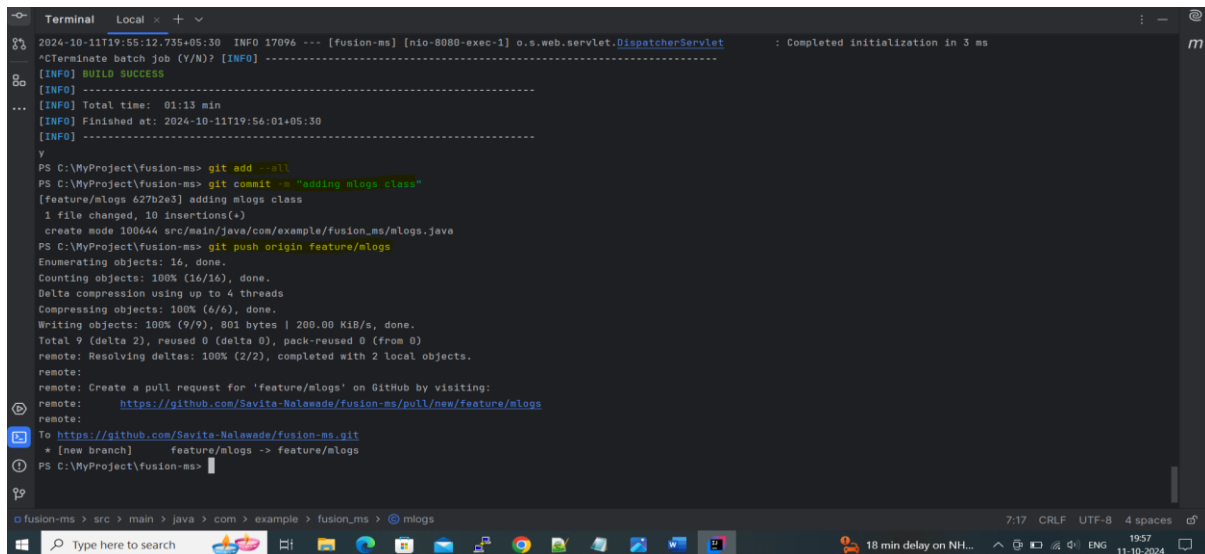
7) Check git status and do mvn clean package command, It will compile successfully



8) We can see output here

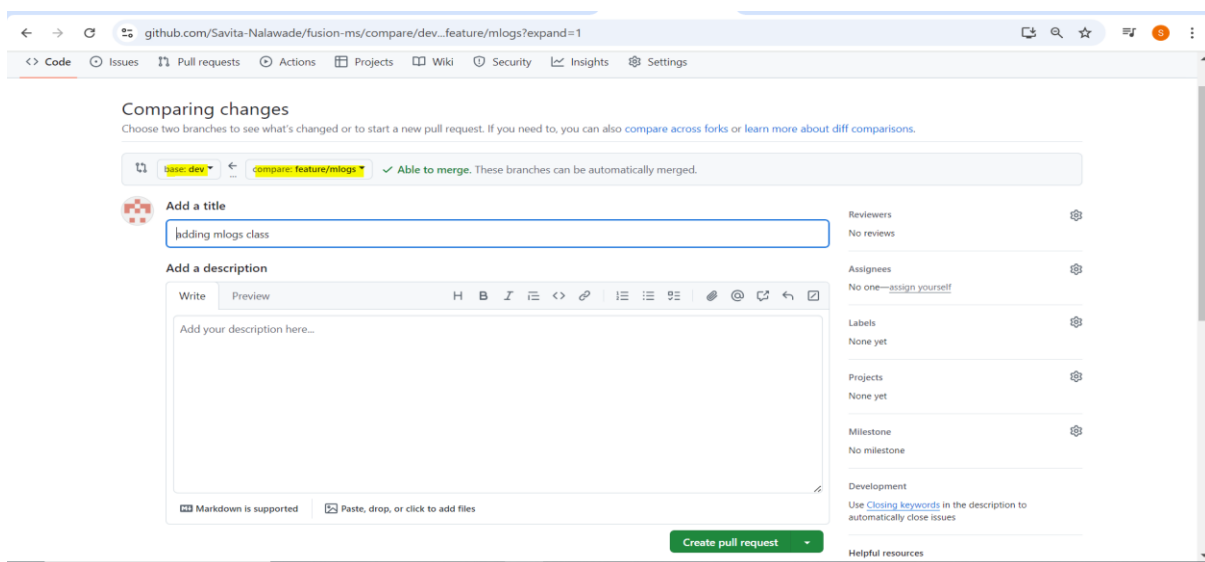


9) Now pass one message and push code to “feature/mlogs”

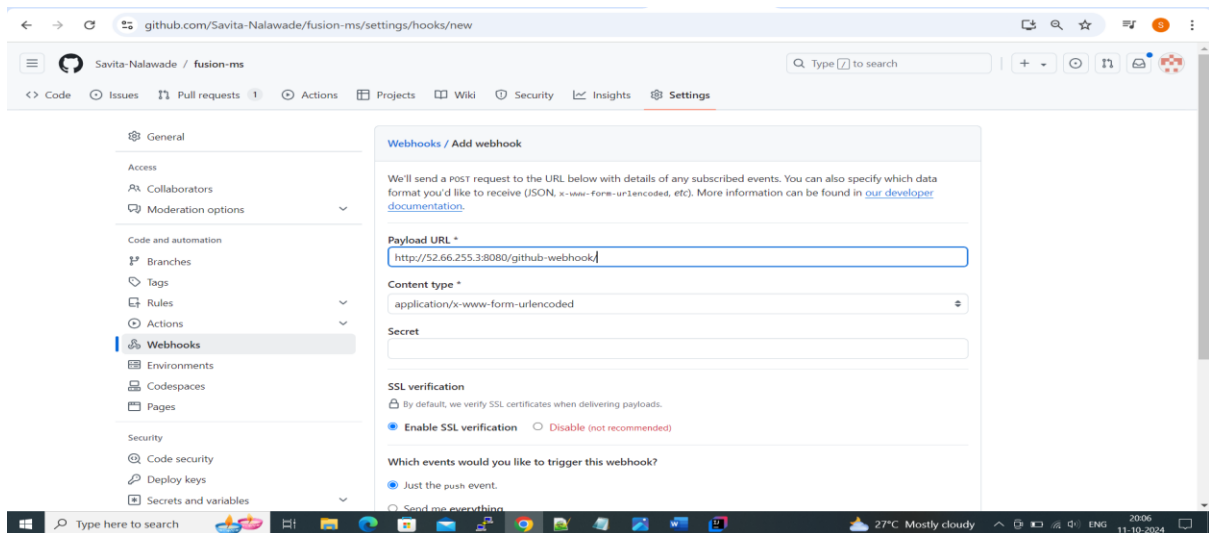


```
2024-10-11T19:55:12.735+05:30 INFO 17096 --- [fusion-ms] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
^CTerminate batch job (Y/N)? [INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:13 min
[INFO] Finished at: 2024-10-11T19:56:01+05:30
[INFO] -----
y
PS C:\MyProject\fusion-ms> git add -all
PS C:\MyProject\fusion-ms> git commit -m "adding mlogs class"
[feature/mlogs 627b2e3] adding mlogs class
1 file changed, 10 insertions(+)
create mode 100644 src/main/java/com/example/fusion_ms/mlogs.java
PS C:\MyProject\fusion-ms> git push origin feature/mlogs
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 801 bytes | 200.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'feature/mlogs' on GitHub by visiting:
remote:   https://github.com/Savita-Nalawade/fusion-ms/pull/new/feature/mlogs
remote:
To https://github.com/Savita-Nalawade/fusion-ms.git
 * [new branch] feature/mlogs -> feature/mlogs
PS C:\MyProject\fusion-ms>
```

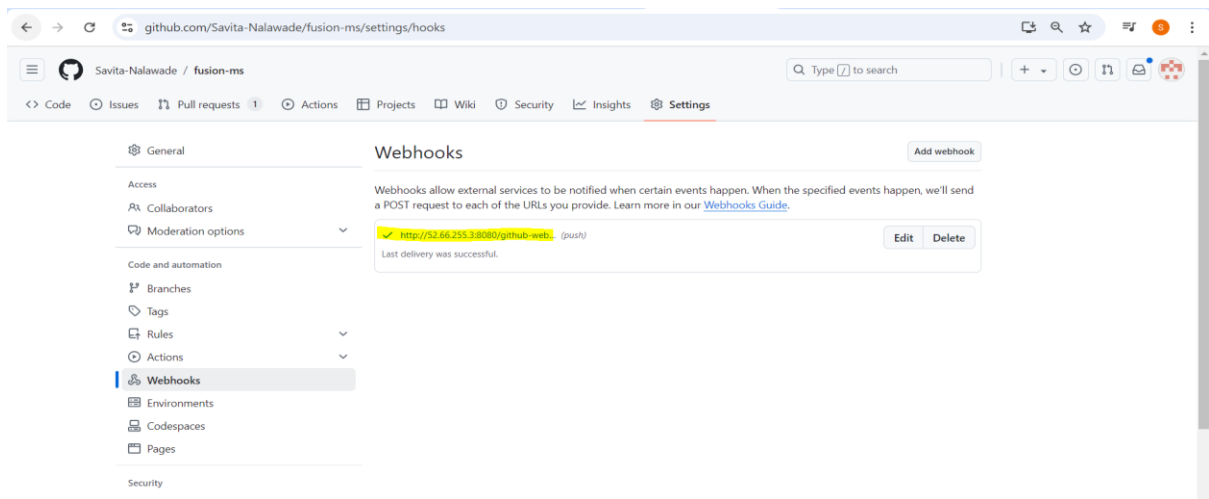
10) Here we can get popup and create pull request of (dev ← feature/mlogs)



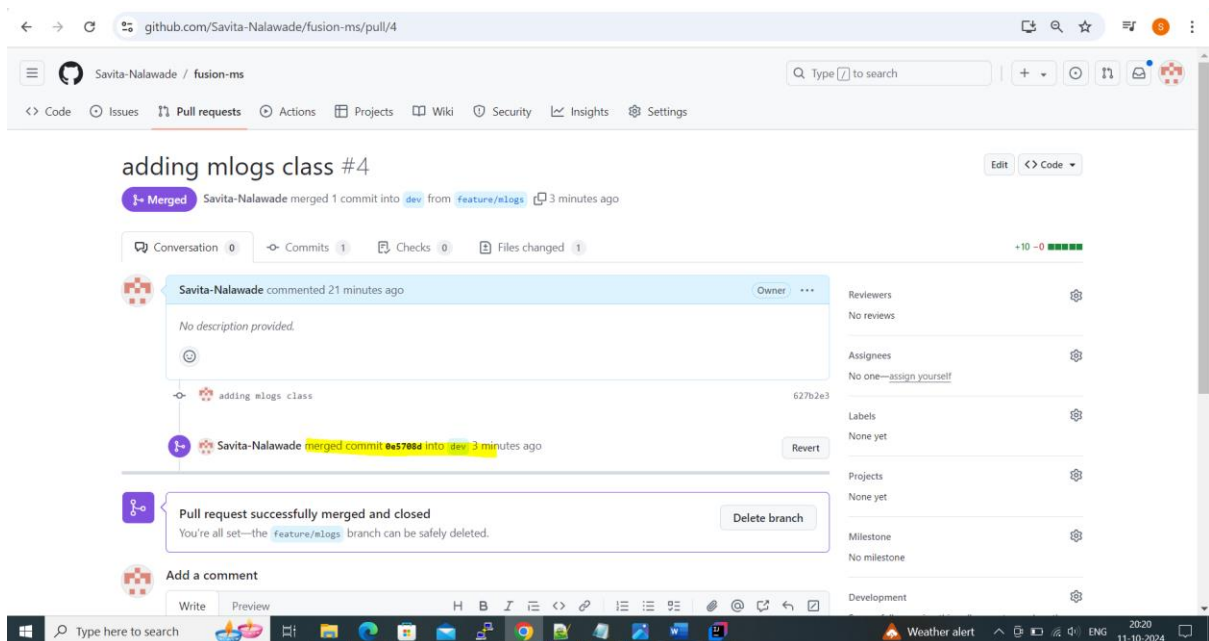
11) Now, Create workbooks to access code in Jenkins application



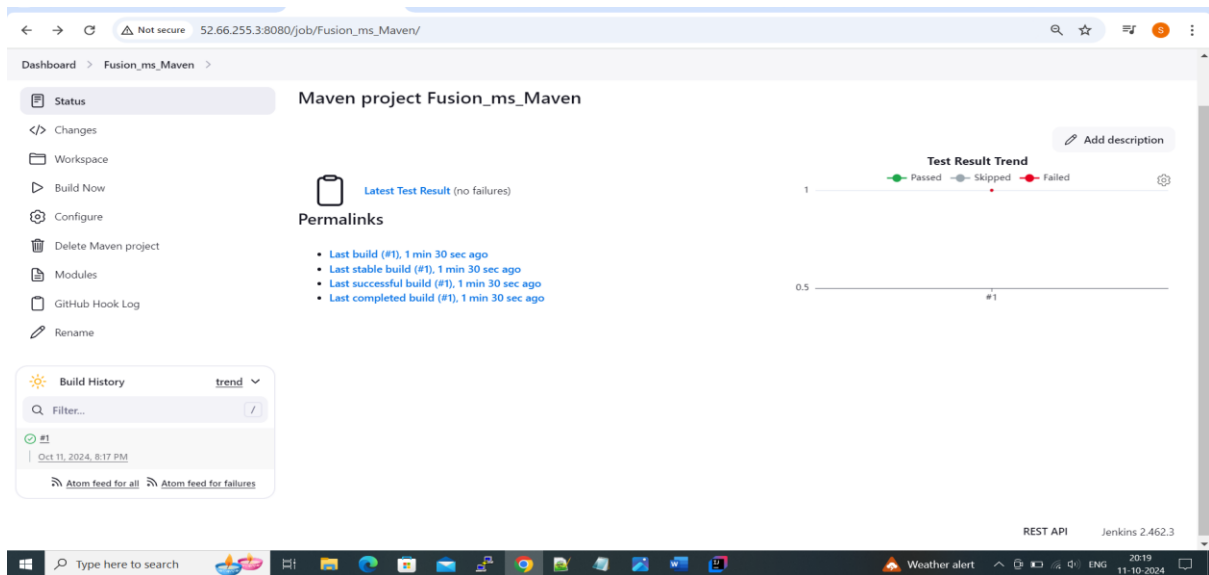
12)Check if webhooks created successfully.



13)after that when we merge branch job should run/trigger

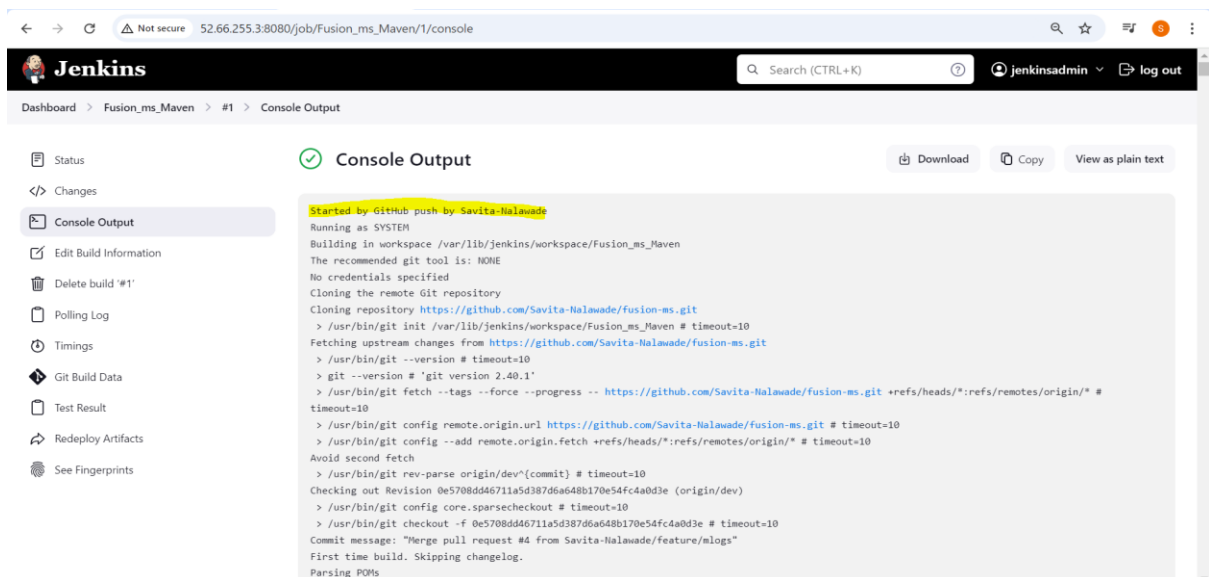


14)We can see in Jenkins your job have run



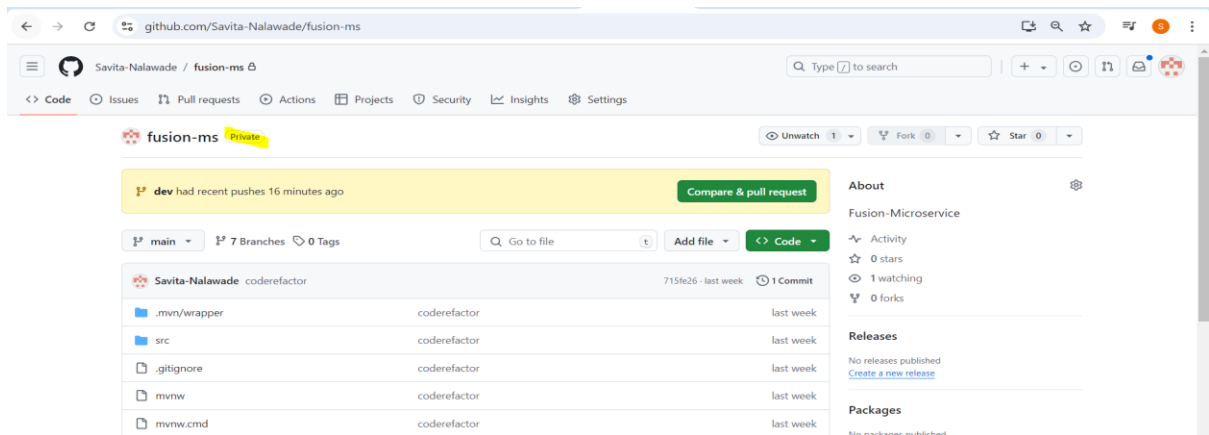
The screenshot shows the Jenkins web interface for a job named 'Fusion\_ms\_Maven'. The left sidebar contains navigation links: Status, Changes, Workspace, Build Now, Configure, Delete Maven project, Modules, GitHub Hook Log, and Rename. The main content area is titled 'Maven project Fusion\_ms\_Maven' and includes a 'Test Result Trend' graph showing a single green dot for 'Passed'. Below the graph, there are 'Permalinks' for the last build, last stable build, last successful build, and last completed build, all indicating they ran 1 minute and 30 seconds ago. A 'Build History' table at the bottom left shows a single build (#1) from Oct 11, 2024, at 8:17 PM, with links to 'Atom feed for all' and 'Atom feed for failures'. The bottom of the browser window shows a Windows taskbar with various application icons and a system clock indicating 20:19 on 11-10-2024.

15)see logs of output

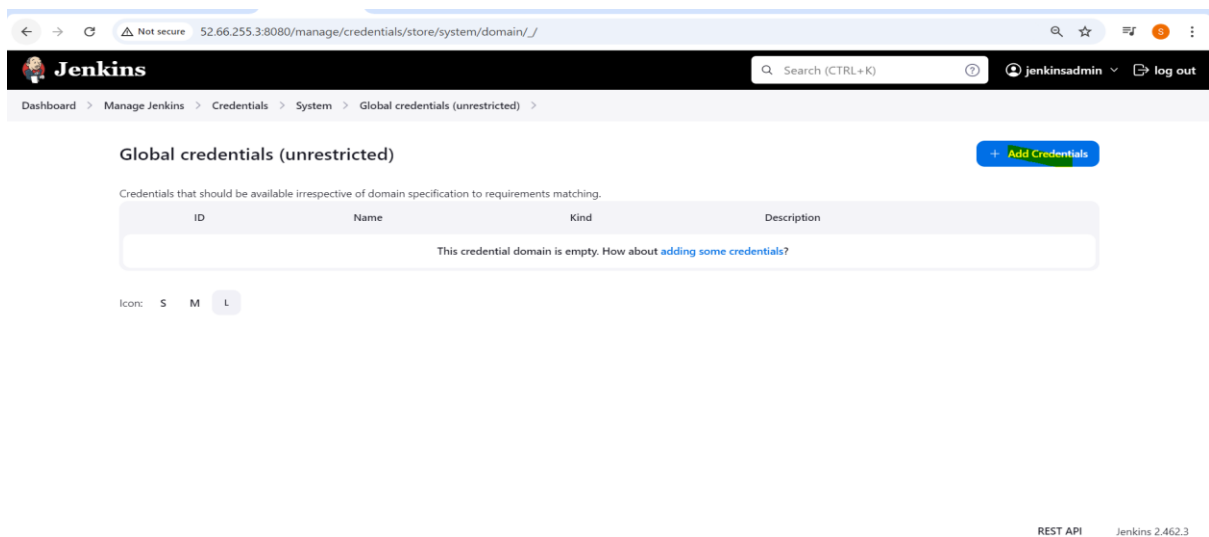


The screenshot displays the 'Console Output' of the Jenkins job. The left sidebar shows navigation links: Status, Changes, Console Output, Edit Build Information, Delete build '#1', Polling Log, Timings, Git Build Data, Test Result, Redeploy Artifacts, and See Fingerprints. The main content area shows the console output text, which begins with 'Started by Github push by Savita-Nalawade' and 'Running as SYSTEM'. The output details the process of cloning the repository 'https://github.com/Savita-Nalawade/fusion-ms.git', fetching upstream changes, and checking out revision '0e5708dd46711a5d387d6a648b170e54fc4a0d3e'. The commit message is 'Merge pull request #4 from Savita-Nalawade/feature/mlogs'. The output ends with 'First time build. Skipping changelog.' and 'Parsing POMs'. The bottom of the browser window shows the same Windows taskbar as the previous screenshot.

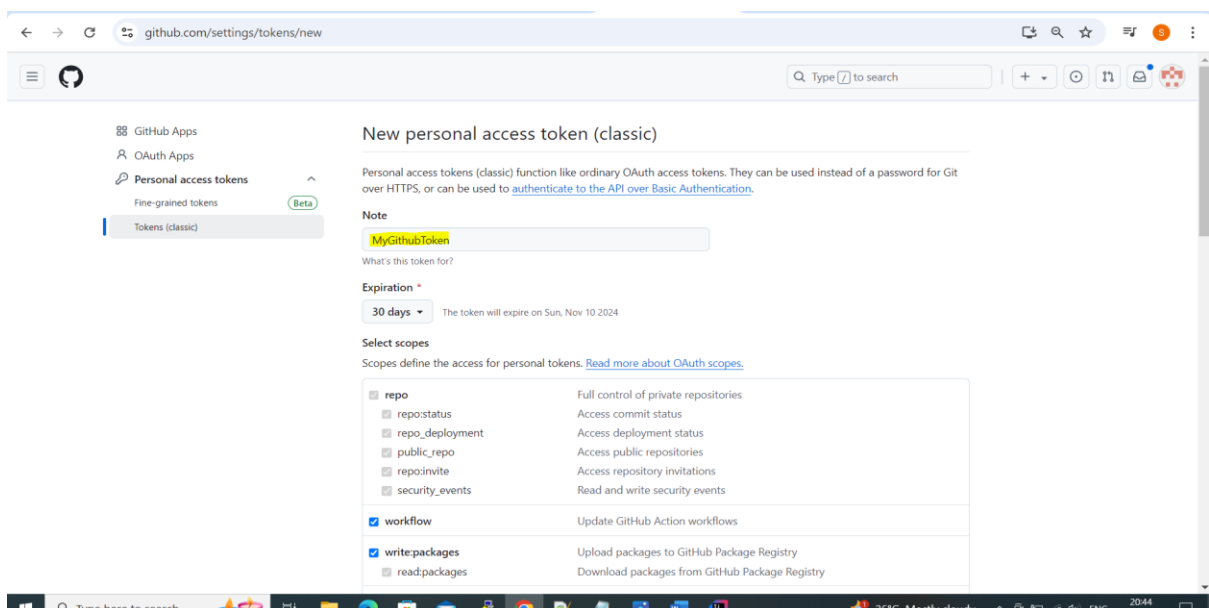
16)Now, we will made repository as private



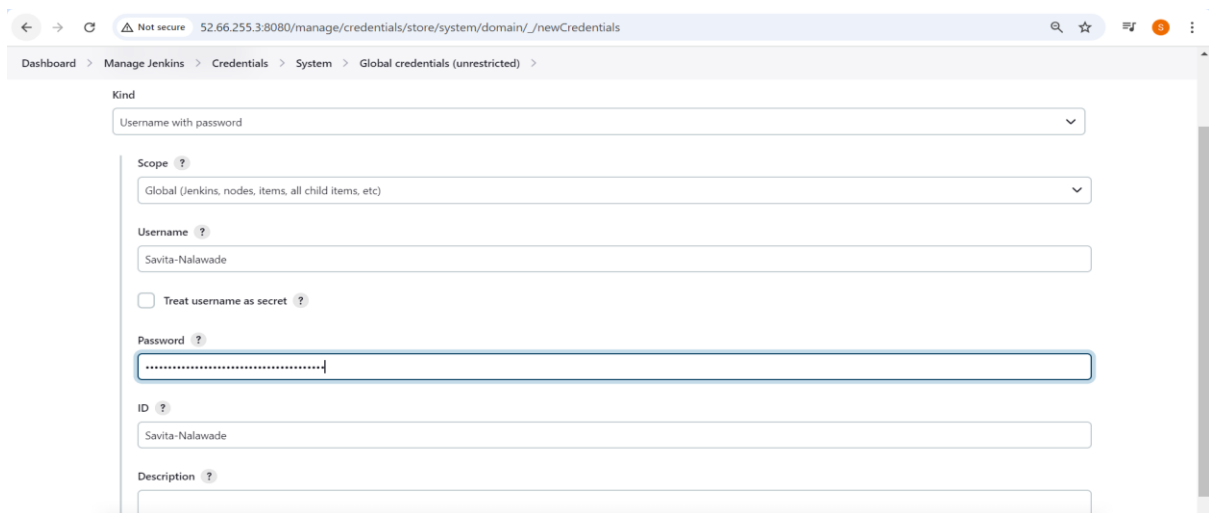
17)when we make repo as private we can not access in Jenkins for that we need to create creds



18)From Github we will create one Token

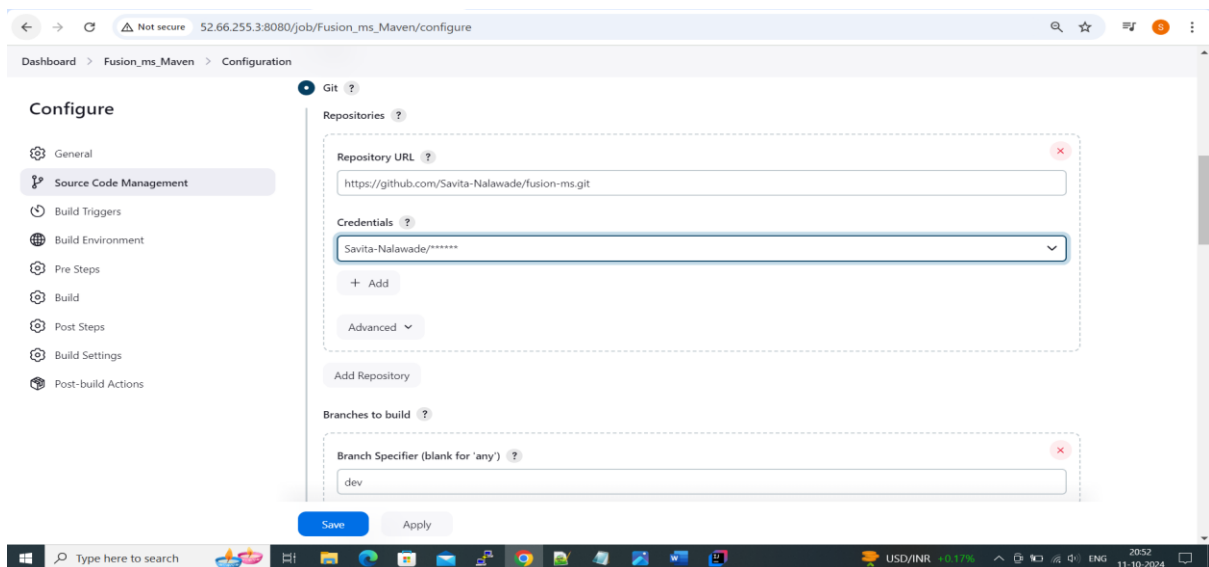


19) paste here password which we get from Token



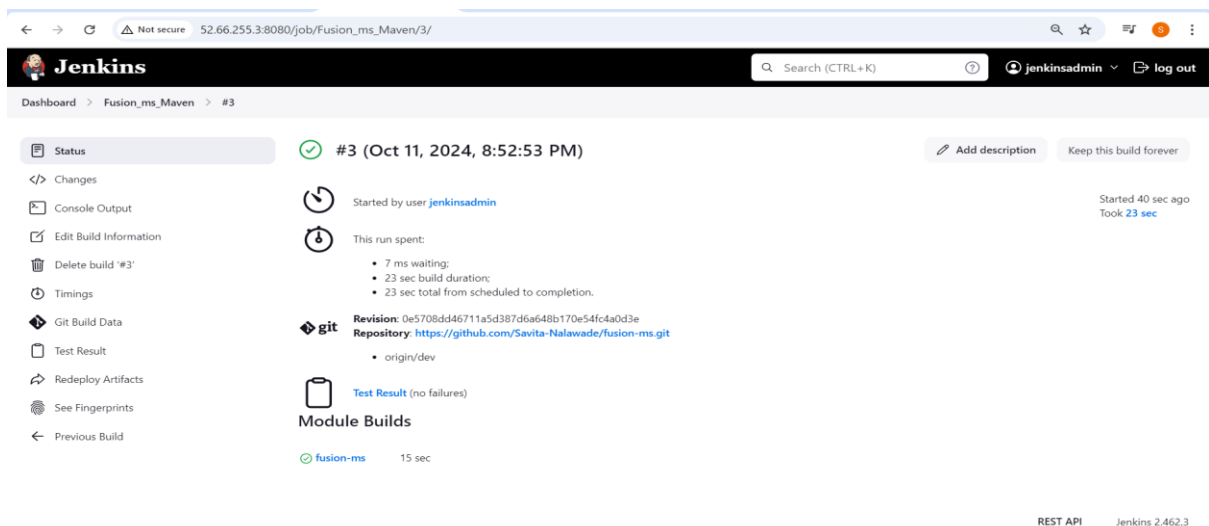
The screenshot shows the Jenkins 'New Credentials' configuration page. The 'Kind' is set to 'Username with password'. The 'Scope' is 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' is 'Savita-Nalawade'. The 'Password' field is filled with a masked password. The 'ID' is also 'Savita-Nalawade'. The 'Description' field is empty.

20) Now we can add creds in Jenkins job



The screenshot shows the Jenkins 'Configure' page for the 'Fusion\_ms\_Maven' job. The 'Source Code Management' section is expanded, showing 'Git' as the provider. The 'Repository URL' is 'https://github.com/Savita-Nalawade/fusion-ms.git'. The 'Credentials' dropdown is set to 'Savita-Nalawade/\*\*\*\*\*'. The 'Branches to build' section is expanded, showing 'Branch Specifier (blank for 'any')' set to 'dev'. The 'Save' button is highlighted.

21) Job have been run successfully



The screenshot shows the Jenkins job '#3' completion page. The job status is 'Success'. The build was started by 'jenkinsadmin' on Oct 11, 2024, at 8:52:53 PM. The build took 23 seconds. The console output shows the build steps: 'git' (revision: 0e5708dd46711a5d387d6a648b170e54f4a0d3e, repository: https://github.com/Savita-Nalawade/fusion-ms.git) and 'Test Result' (no failures). The 'Module Builds' section shows 'fusion-ms' completed in 15 seconds. The page includes a sidebar with navigation links like 'Status', 'Changes', 'Console Output', etc.



22)Below are the Jobs which we have created till now....

The screenshot shows the Jenkins Dashboard interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user information (jenkinsadmin). Below the navigation bar, the main content area displays a list of jobs. On the left, there's a sidebar with navigation links: New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main area shows a table of jobs with columns for status, name, last success, last failure, and last duration. Two jobs are listed: 'Freestyle\_Ping\_every\_5min' and 'Fusion\_ms\_Maven'. Below the table, there's a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing 'built-in node (0 of 2 executors busy)'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Freestyle_Ping_every_5min	8 hr 53 min #19	9 hr 1 min #11	8.2 sec
✓	☀	Fusion_ms_Maven	32 min #1	N/A	35 sec

THANK YOU!!!!!!