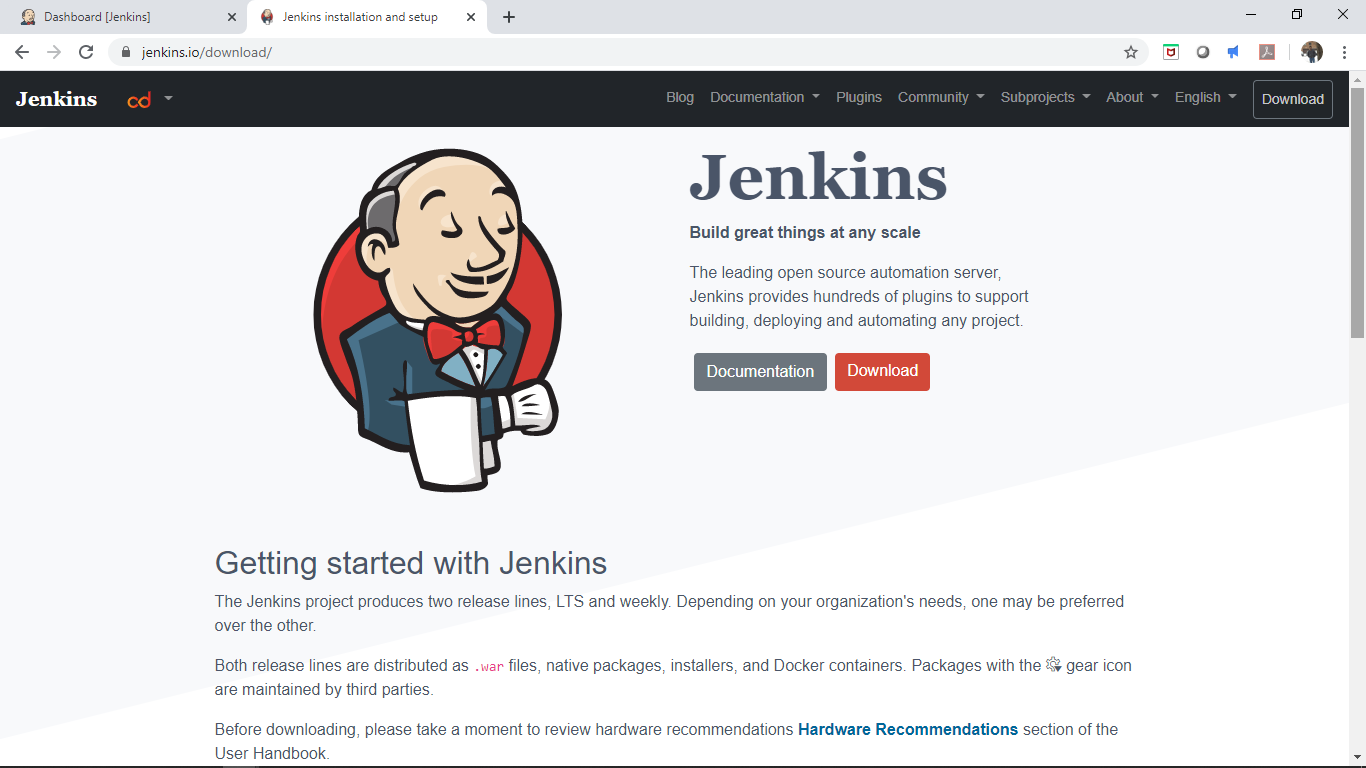
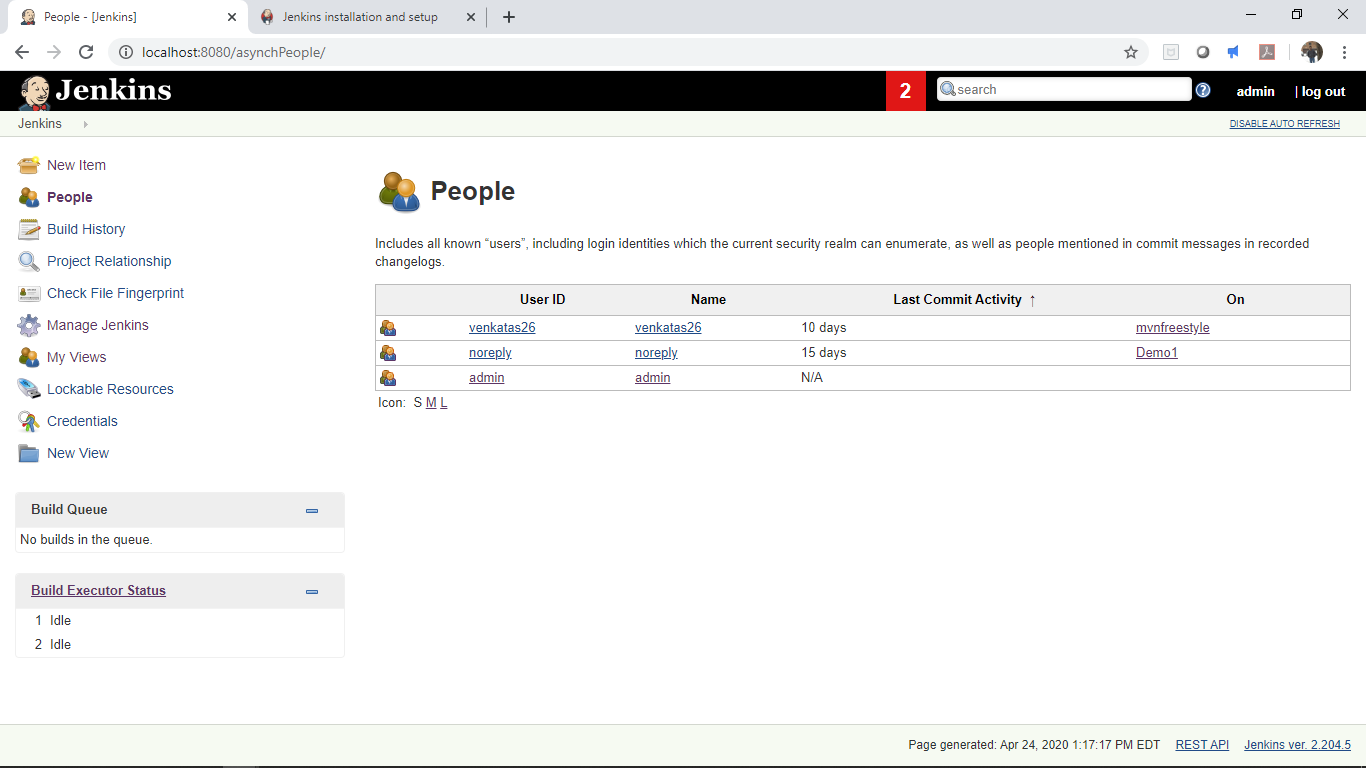
**1.Jenkins Installation:**

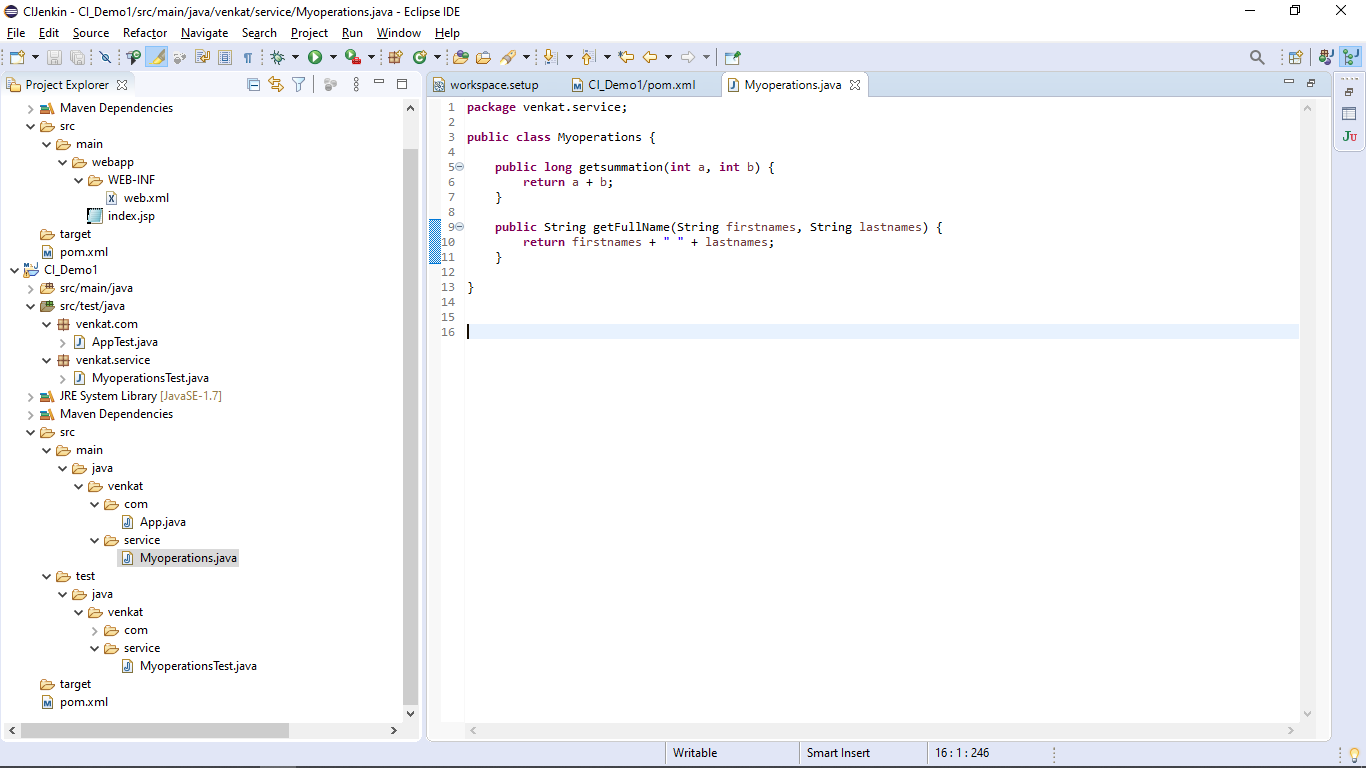
Jenkins installed using official website as shown below:

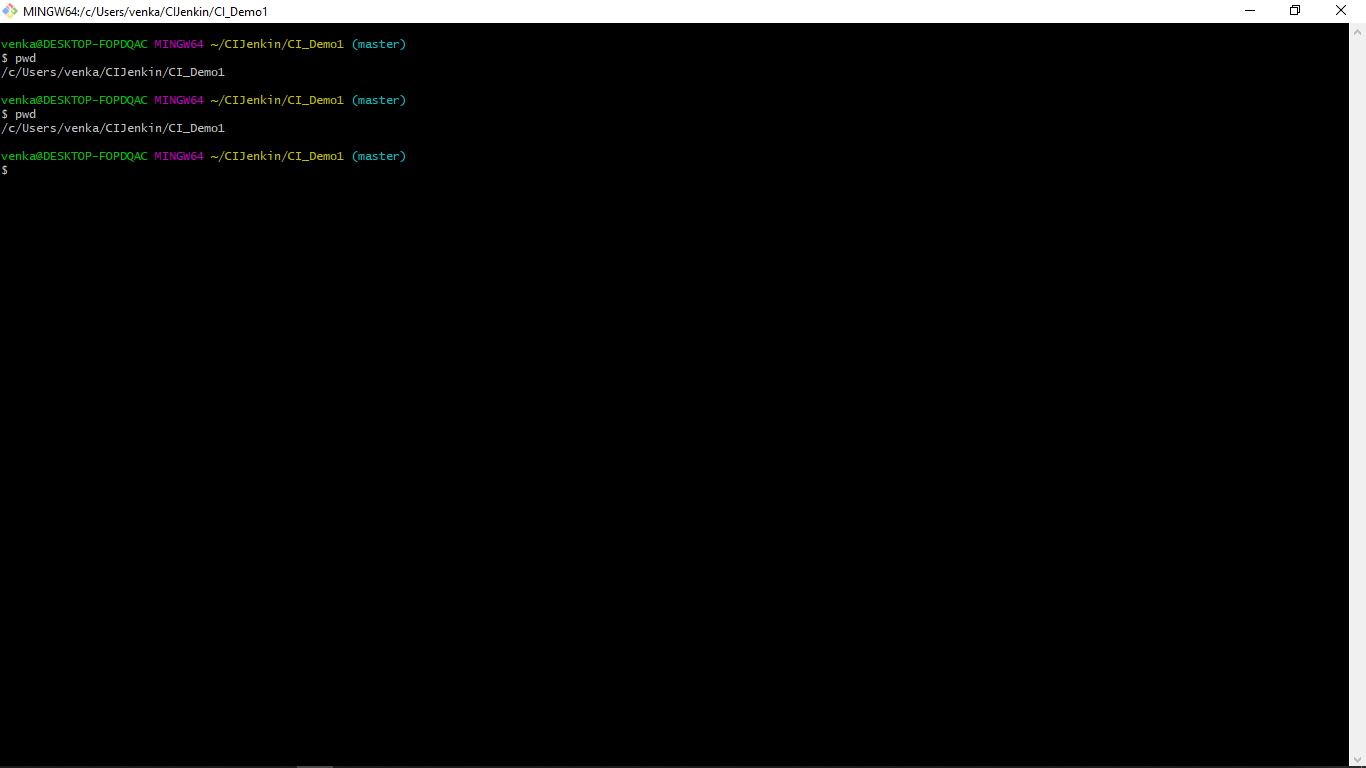




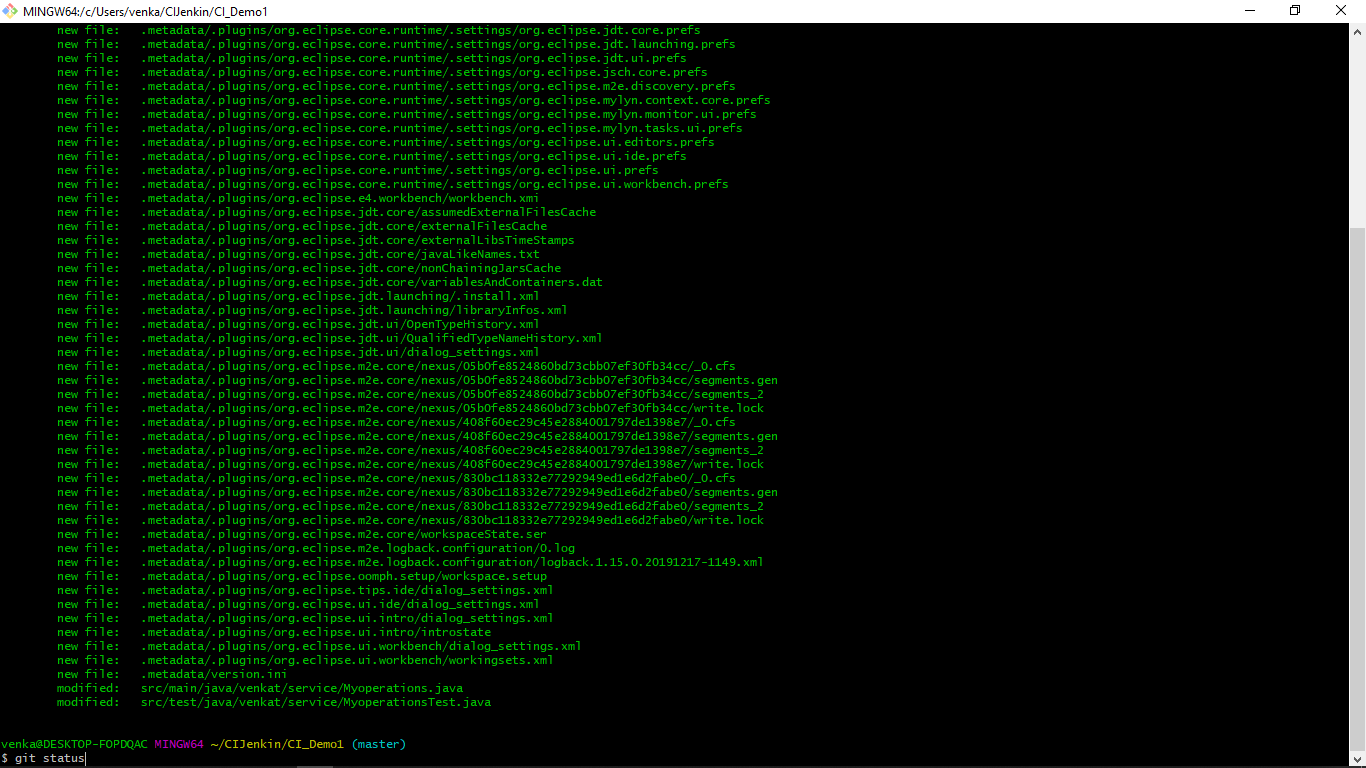
**2. Job Creation in Jenkins**

Created a Java application in eclipse as shown below and the source is in local repo in the path “C:\Users\venka\CIJenkin\CI\_Demo1”

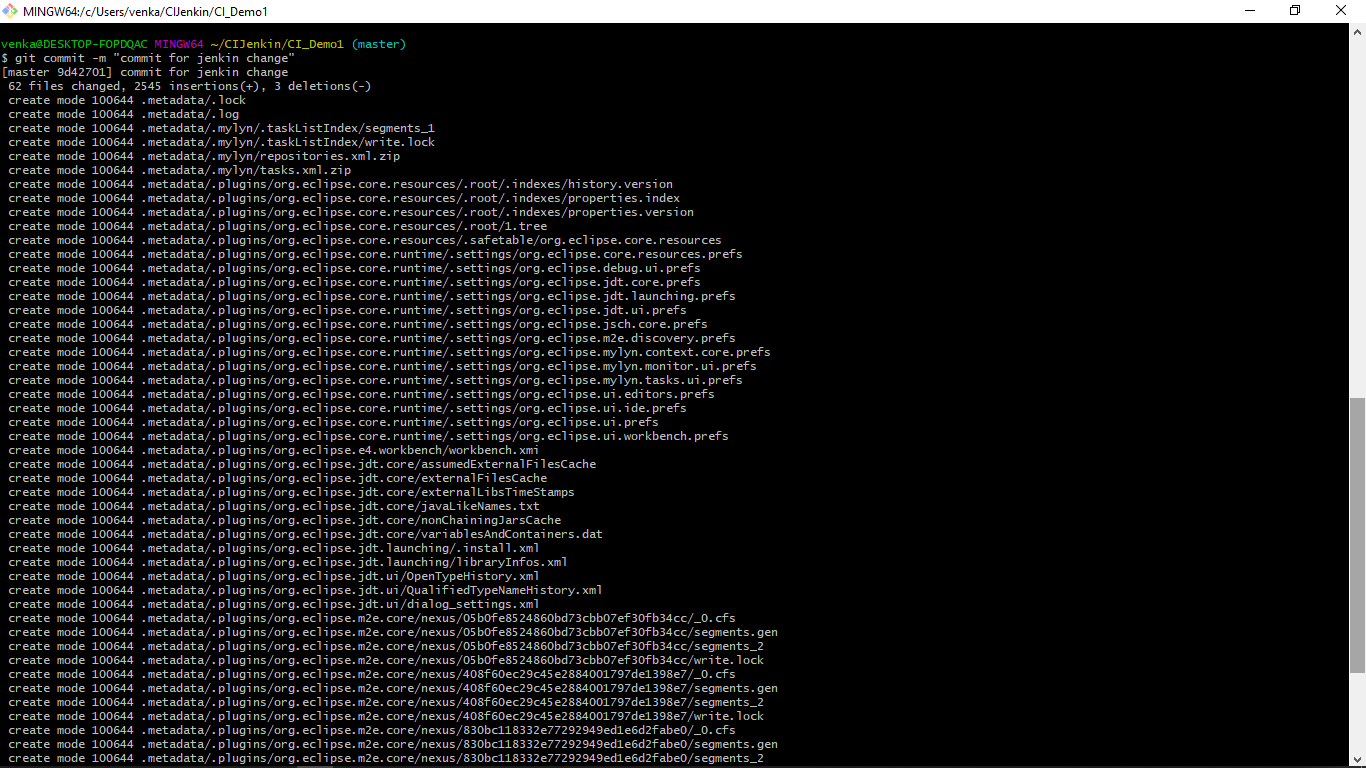




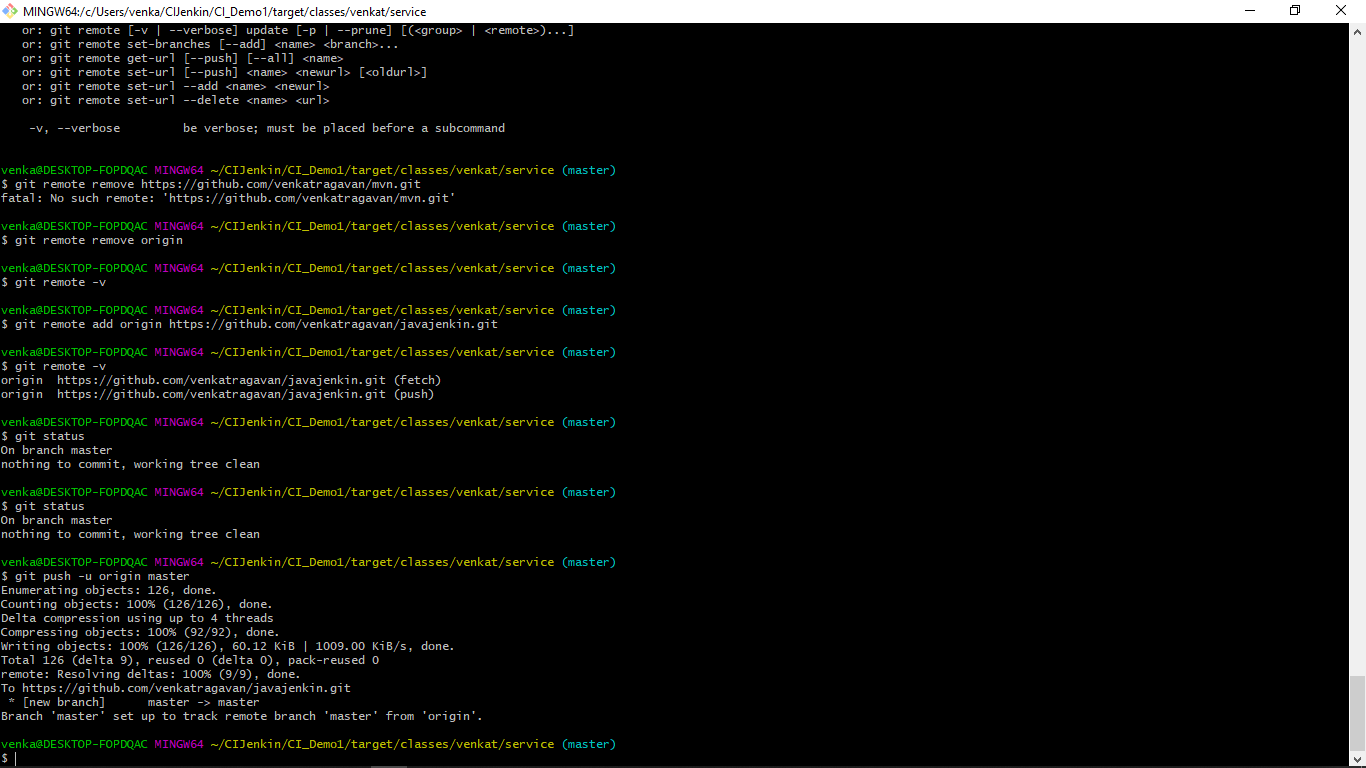
Added the source in local git :

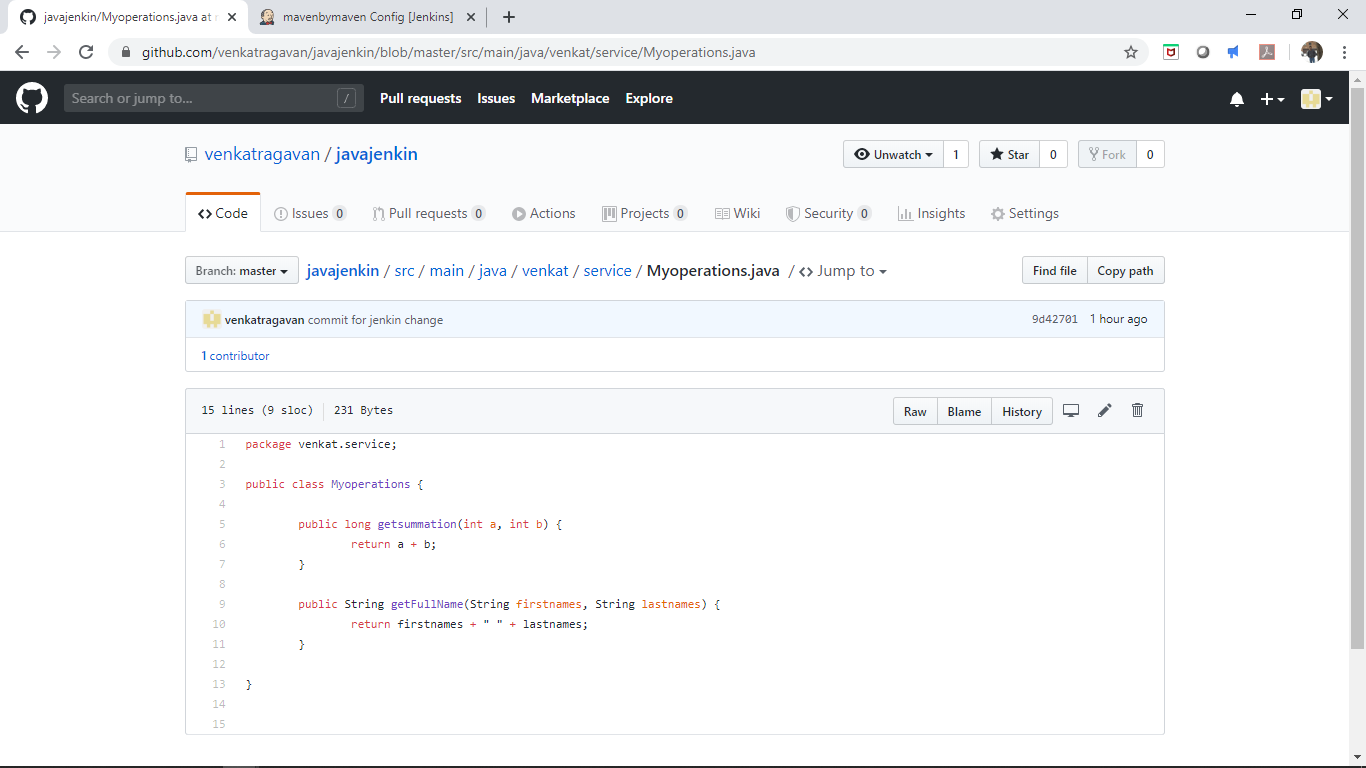


Committed to the local repositry:

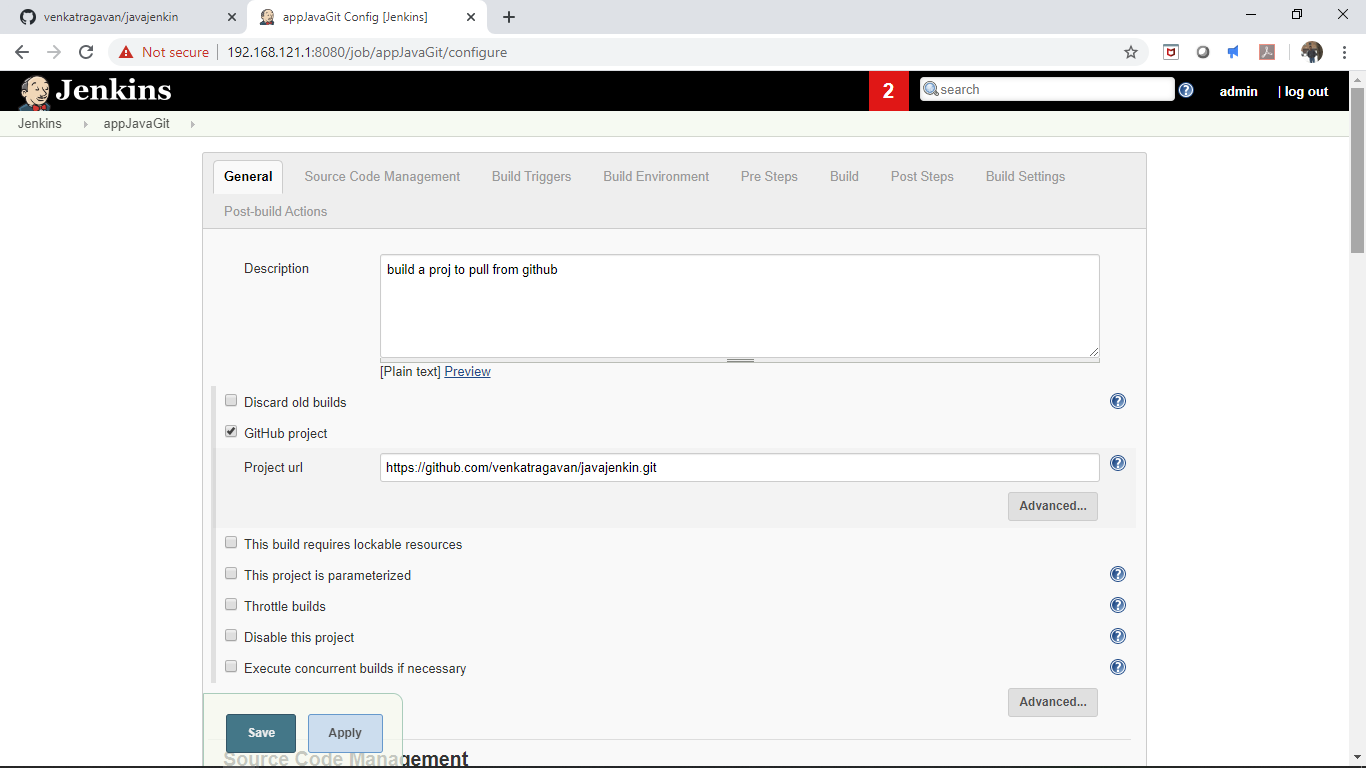


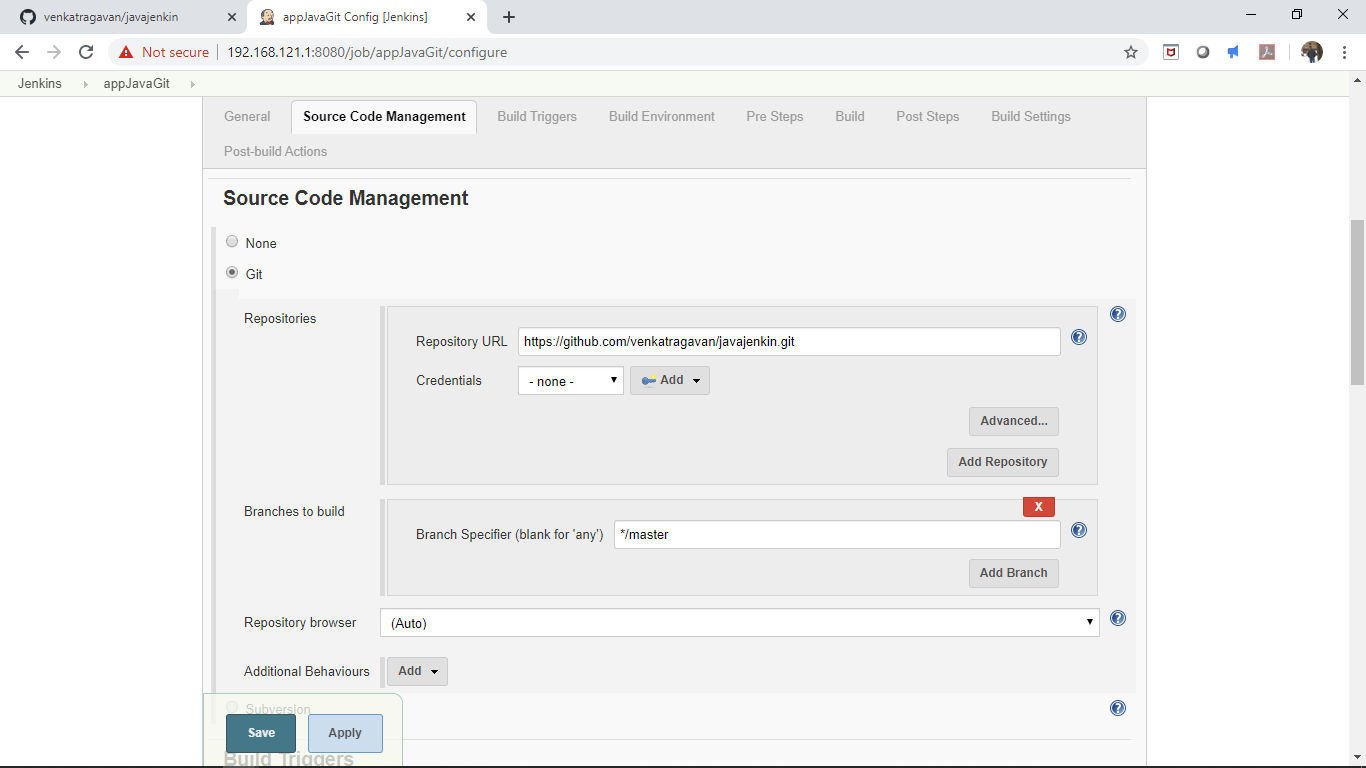
Pushed the changes to remote repositry in github:

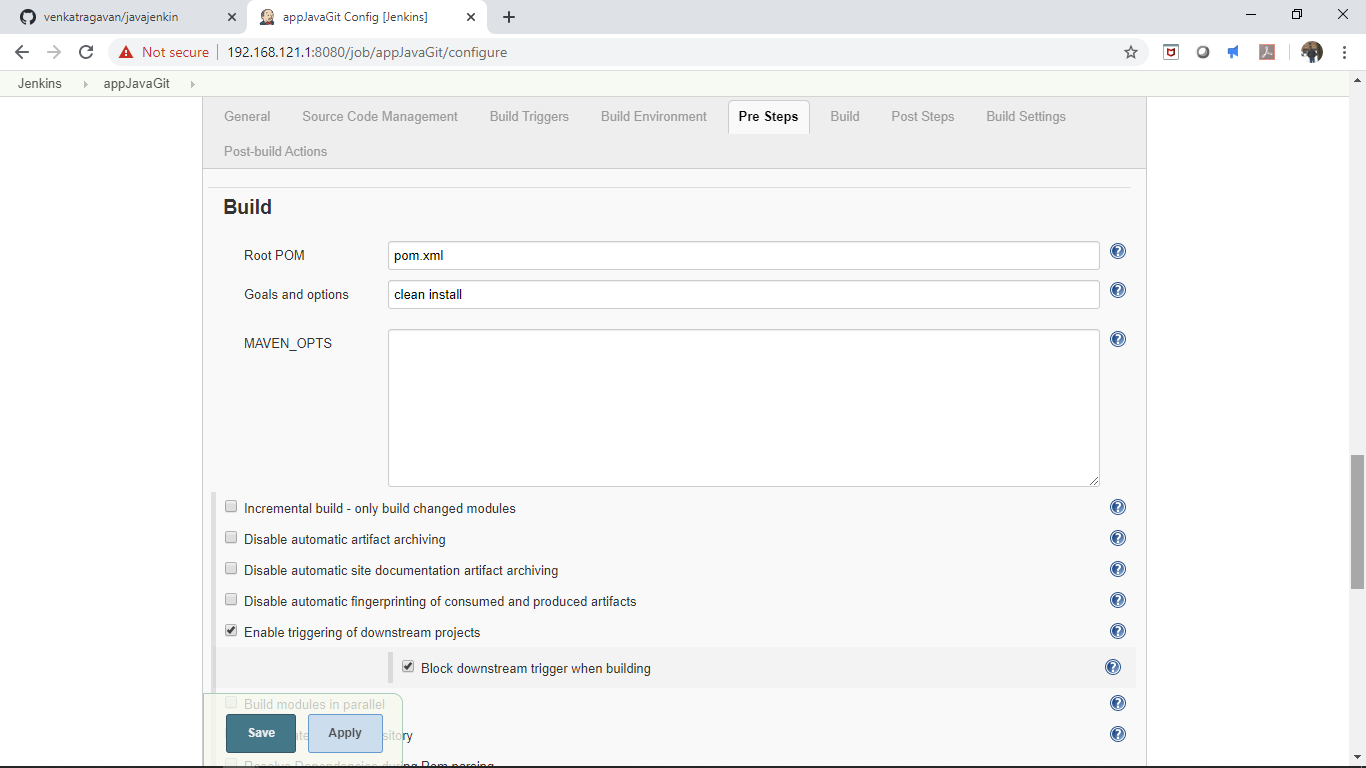


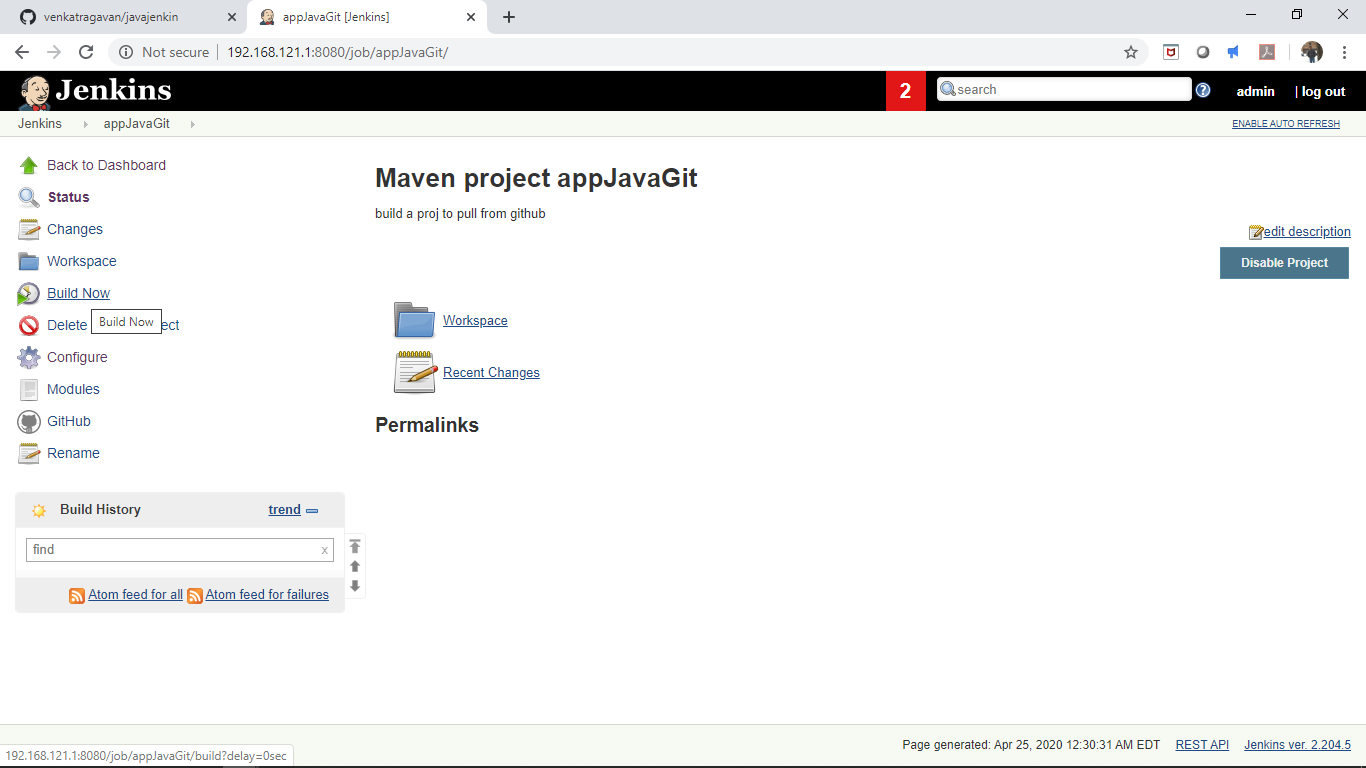


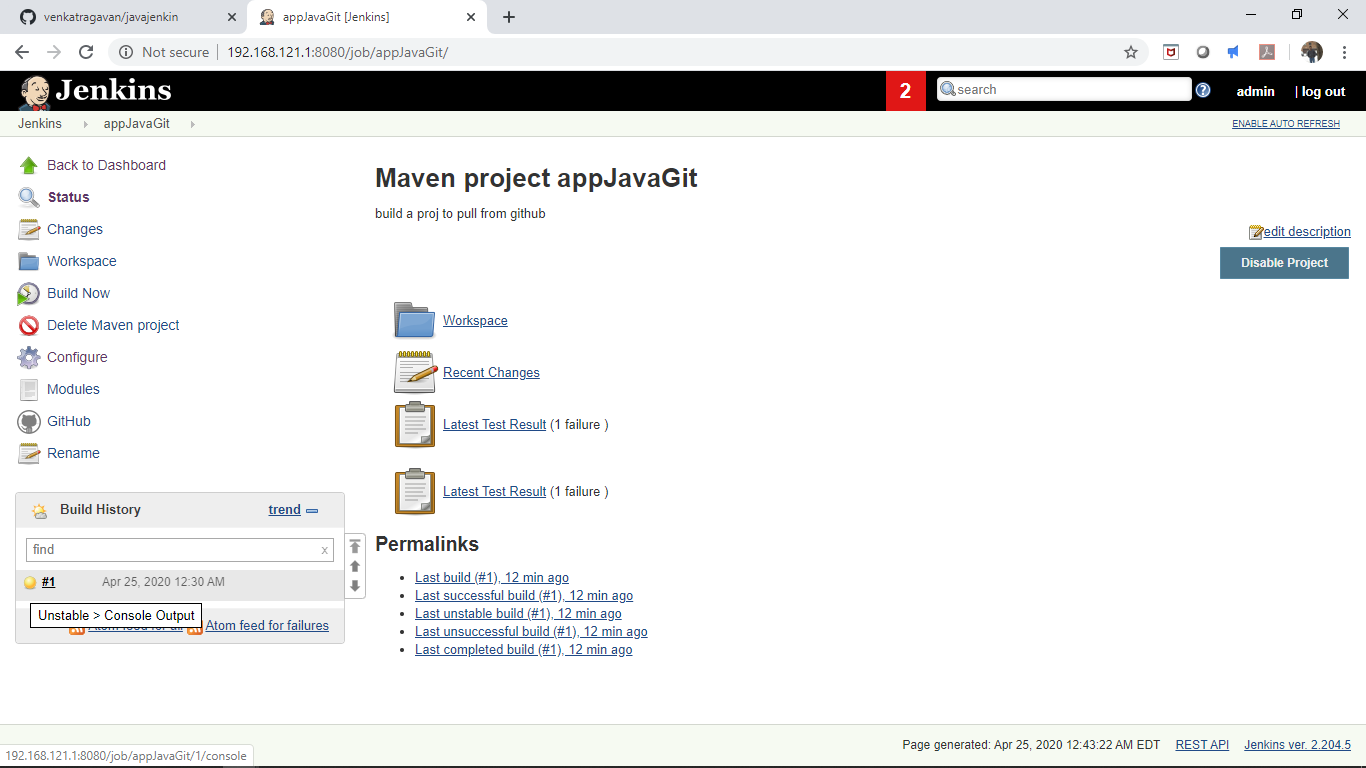


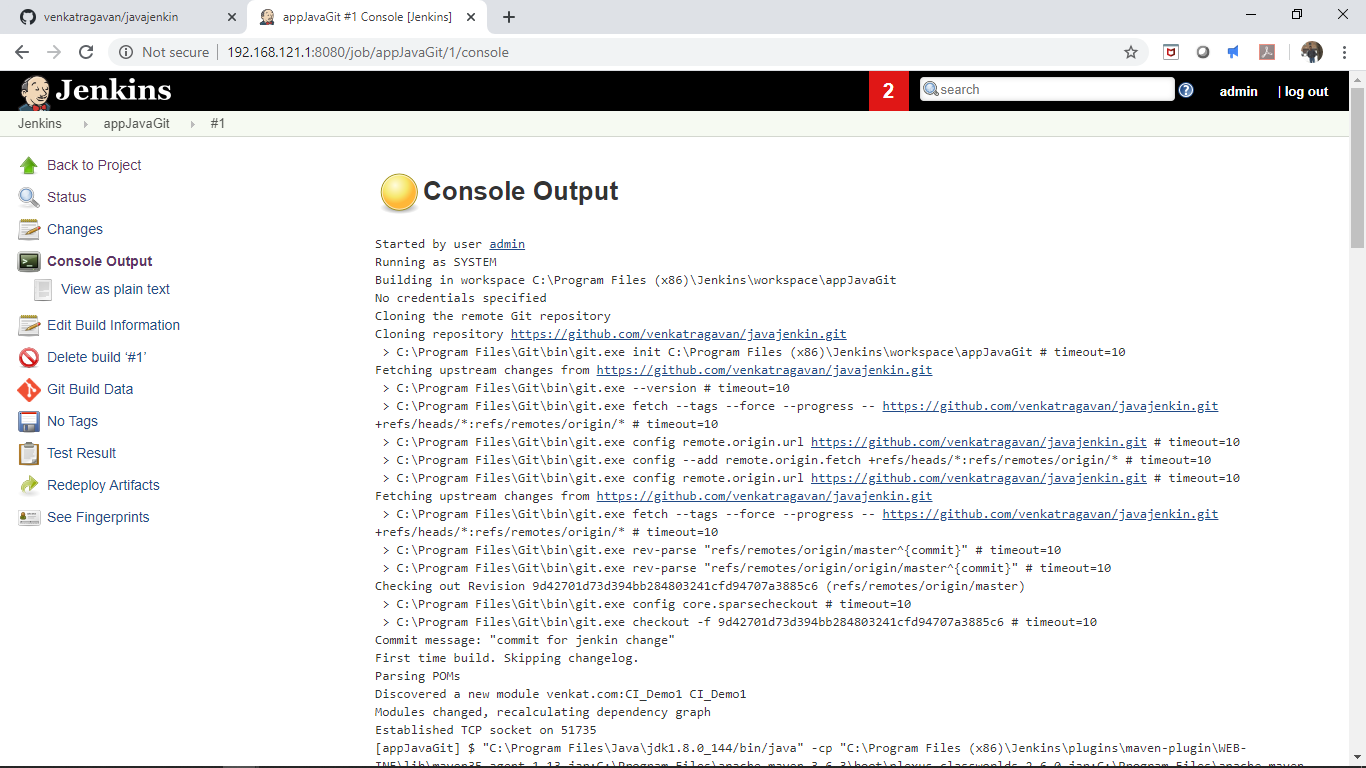




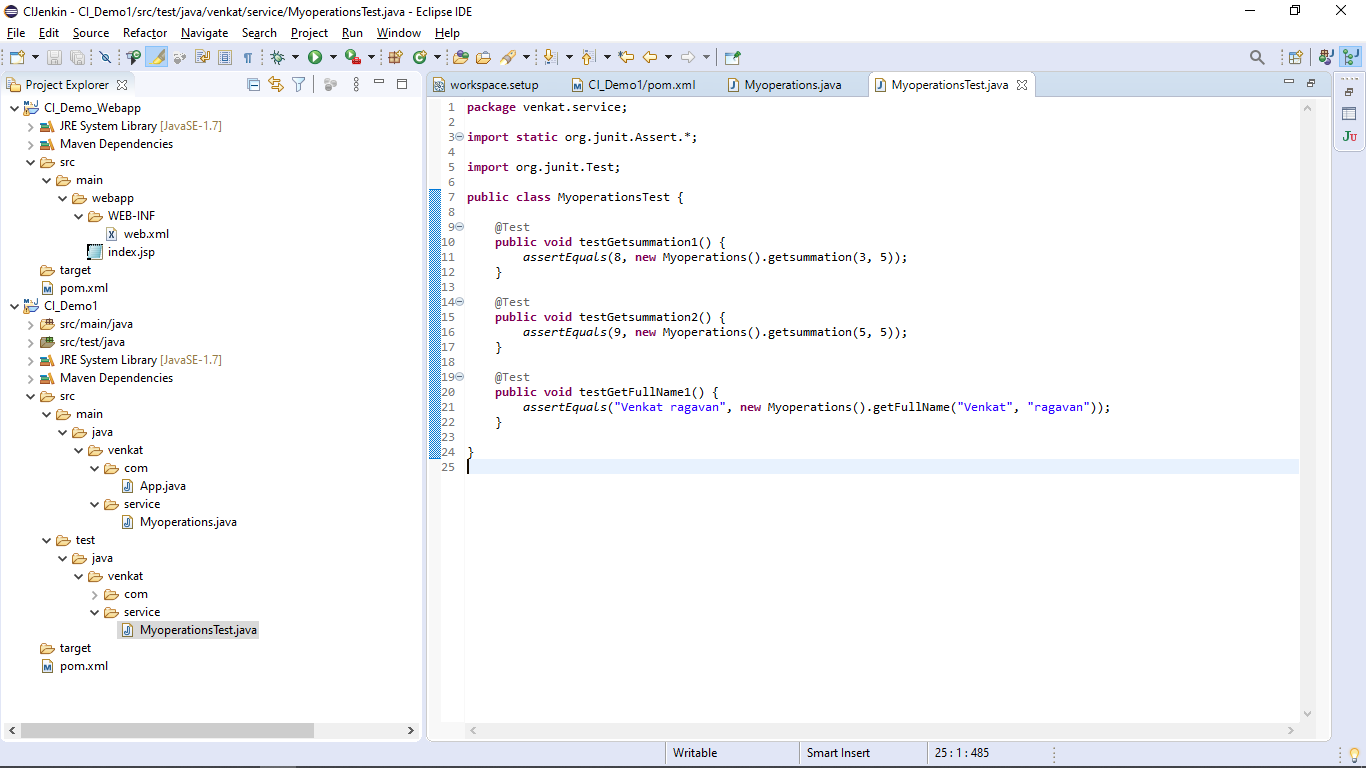


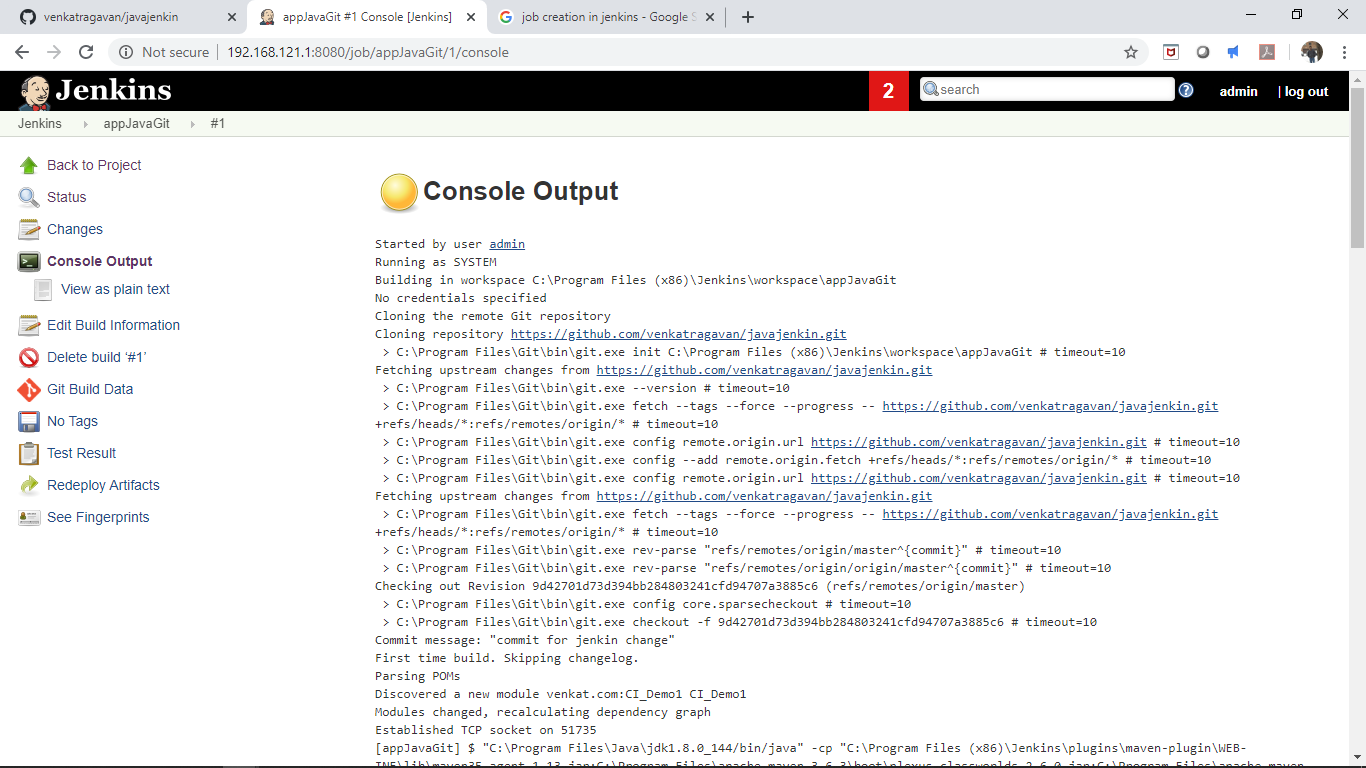


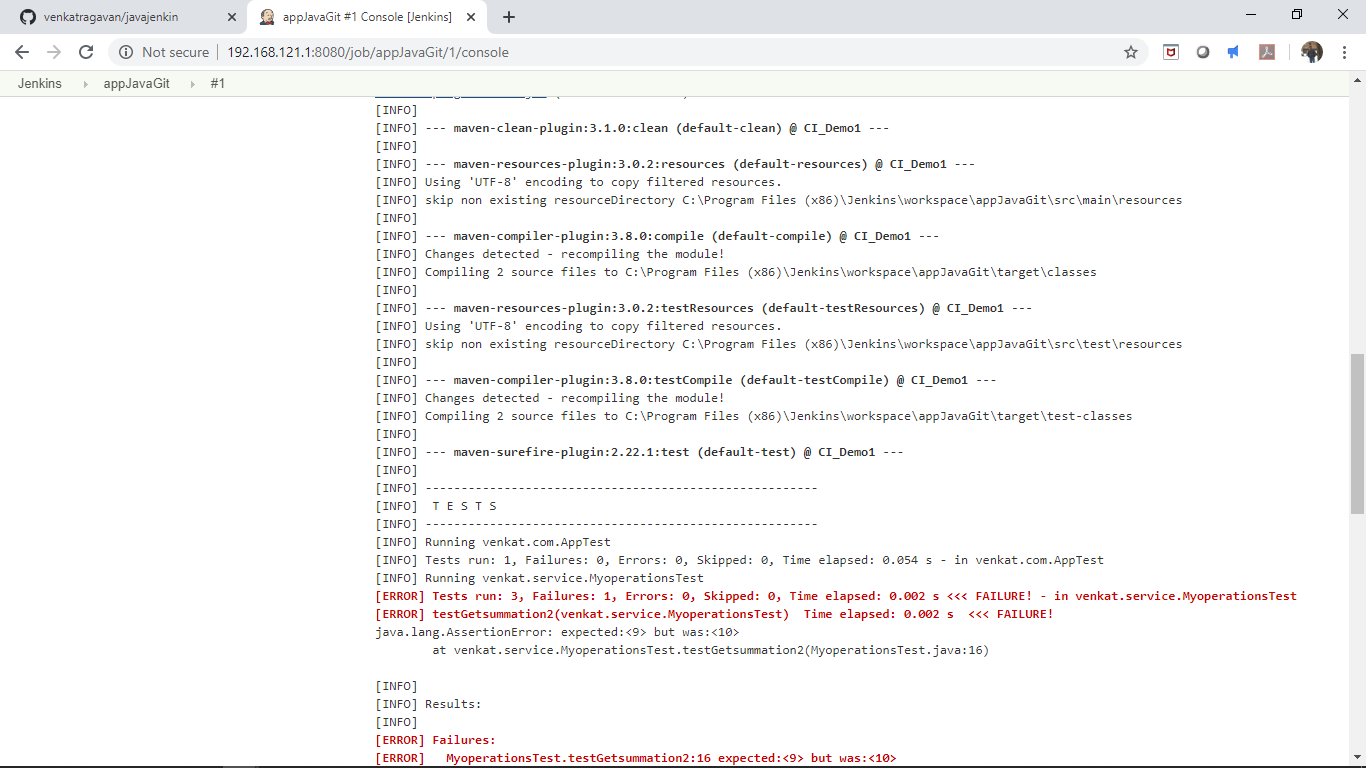


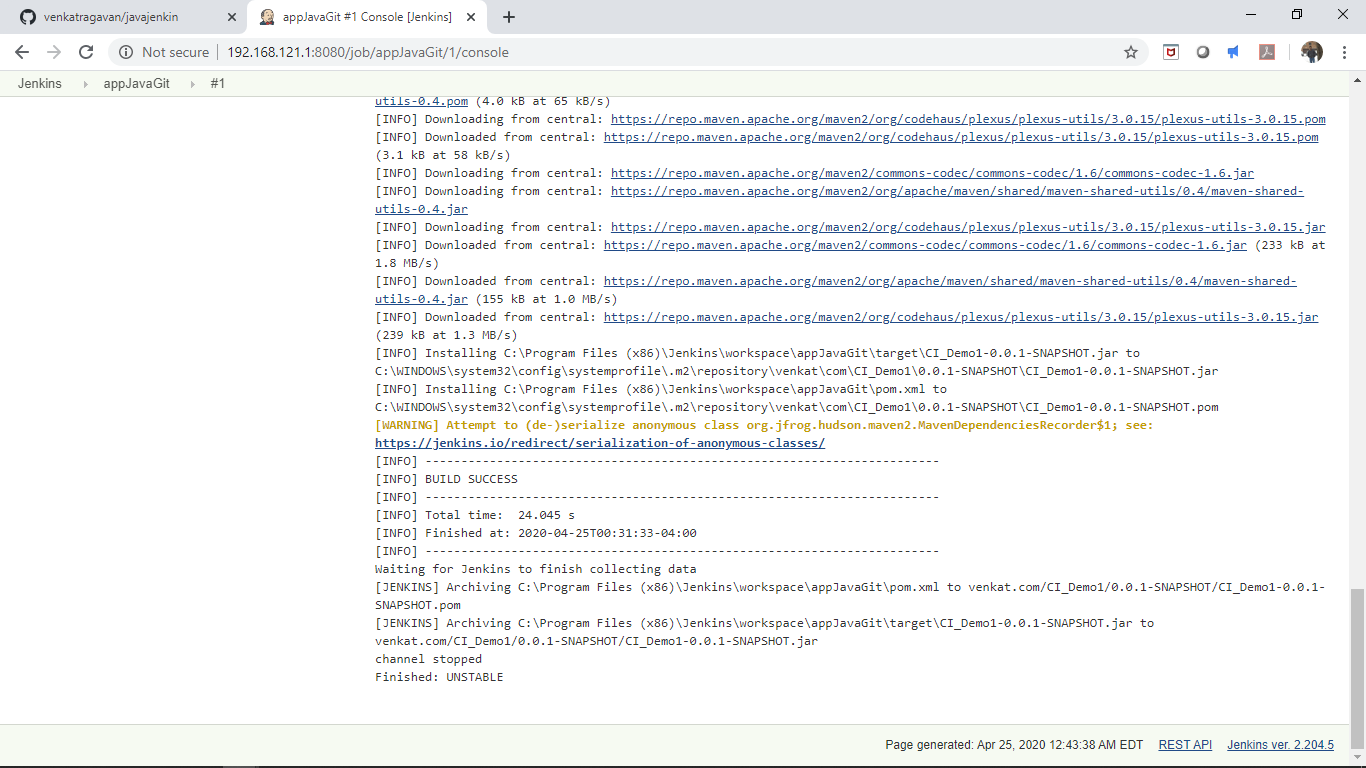


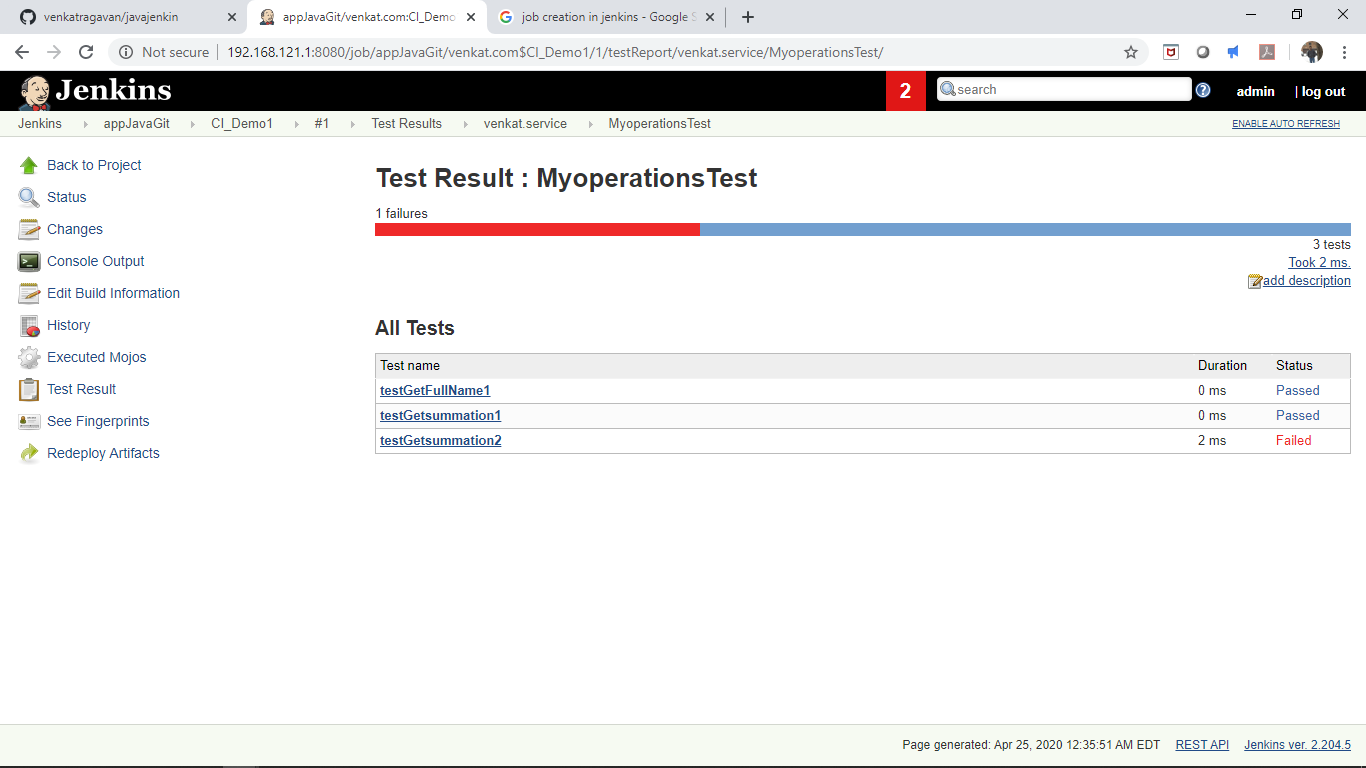
The build was unstable as one of the test result in junit failed which was done intentionally to check the build stability.











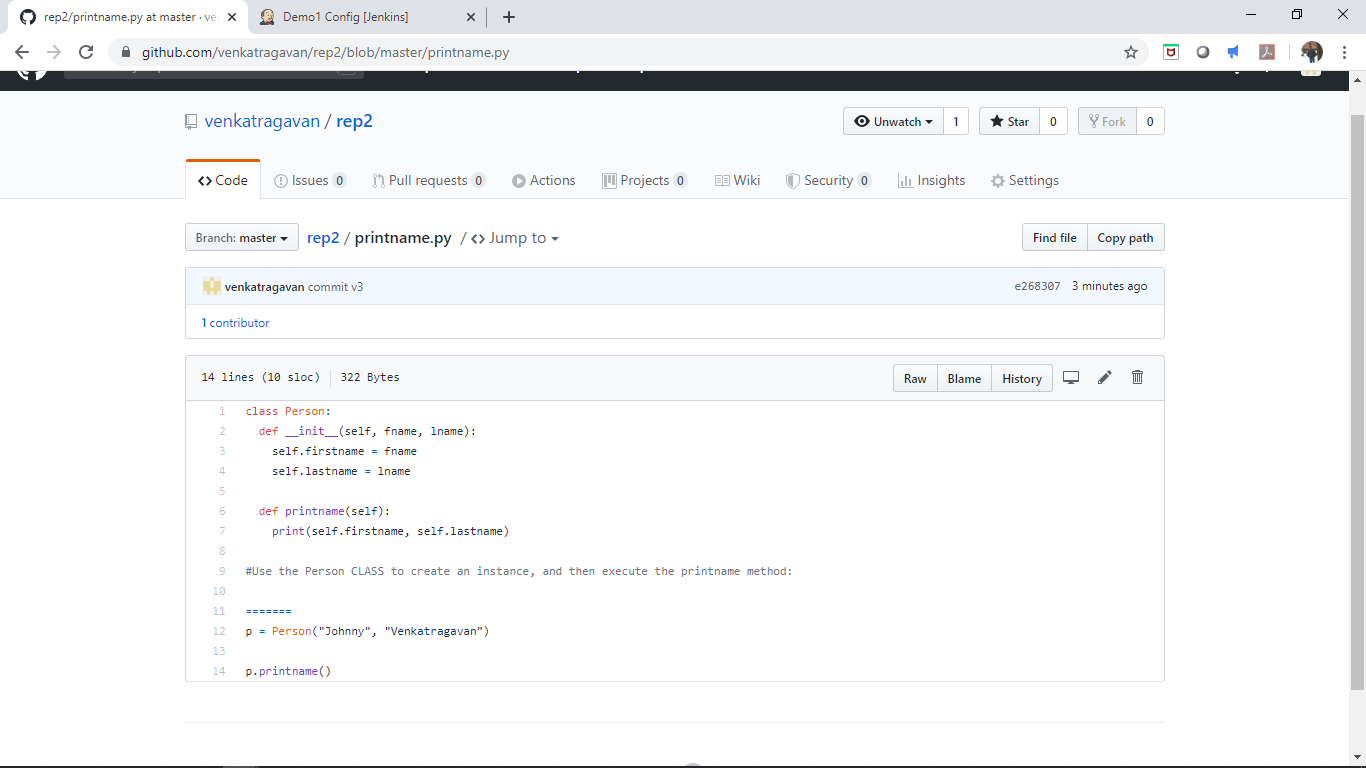
**3. Job Scheduling**

Created a small python application and moved the same from local git to remote github. Built a jenkin job with build trigger “GitHub hook trigger for GITScm polling” this enables the build to be scheduled whenever there is a commit at the remote github repository using webhooks. Please refer the below screenshot:

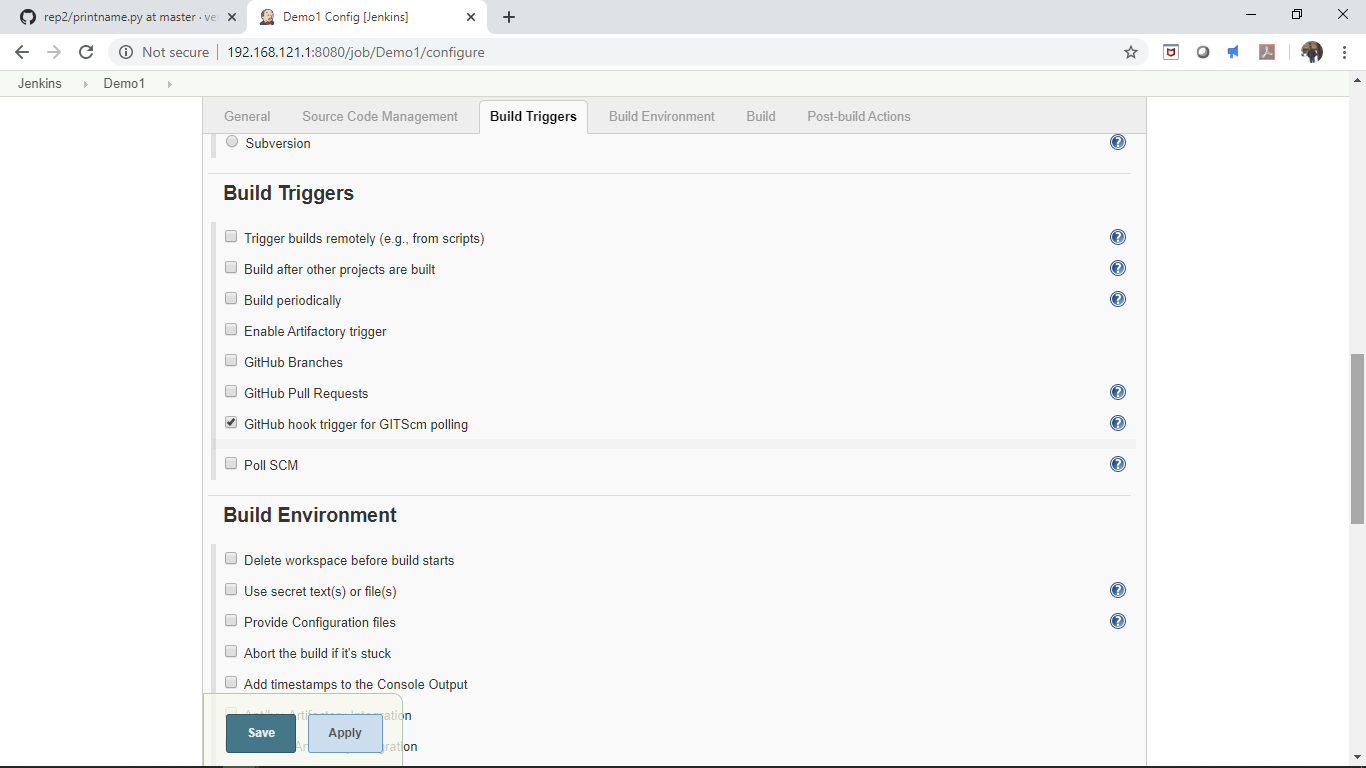
Jenkin job Demo1:



Remote Github repository:

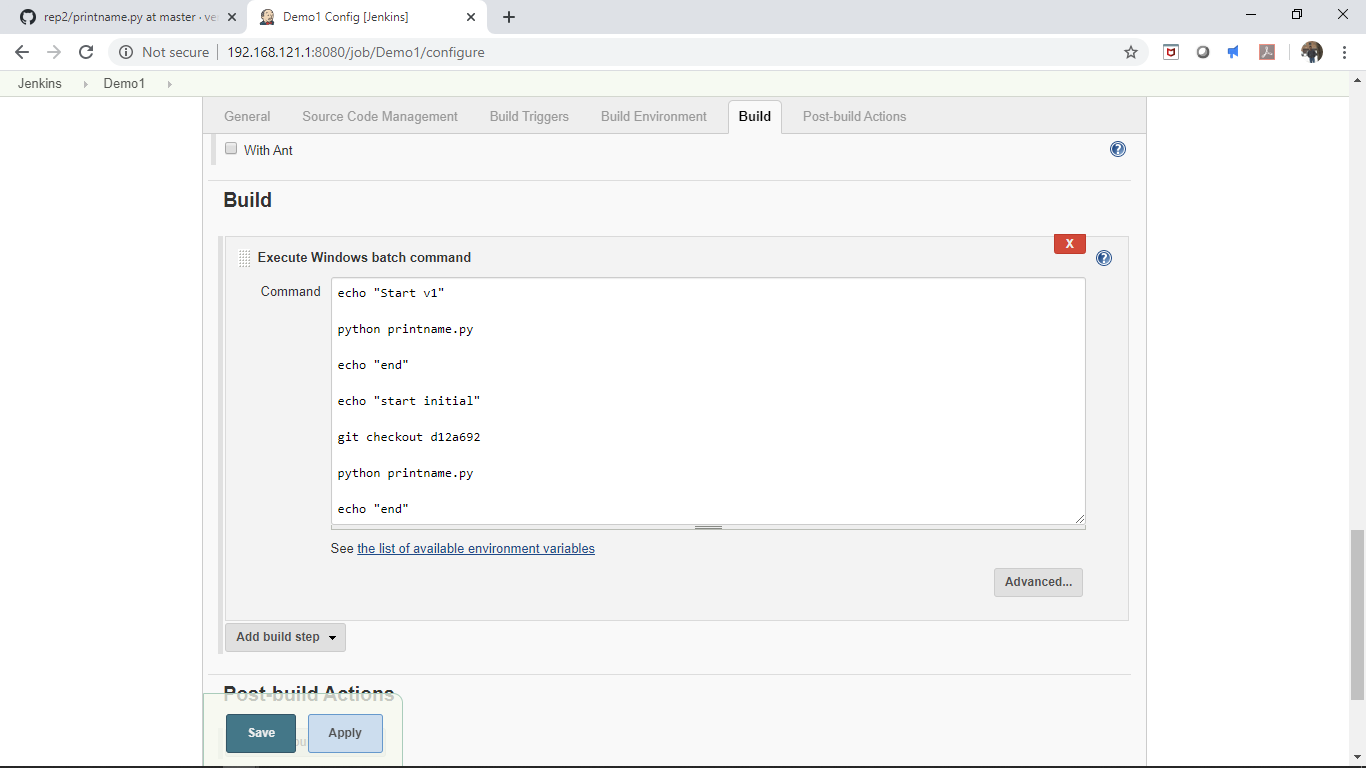


Jenkin job has a build trigger for Github hook trigger:

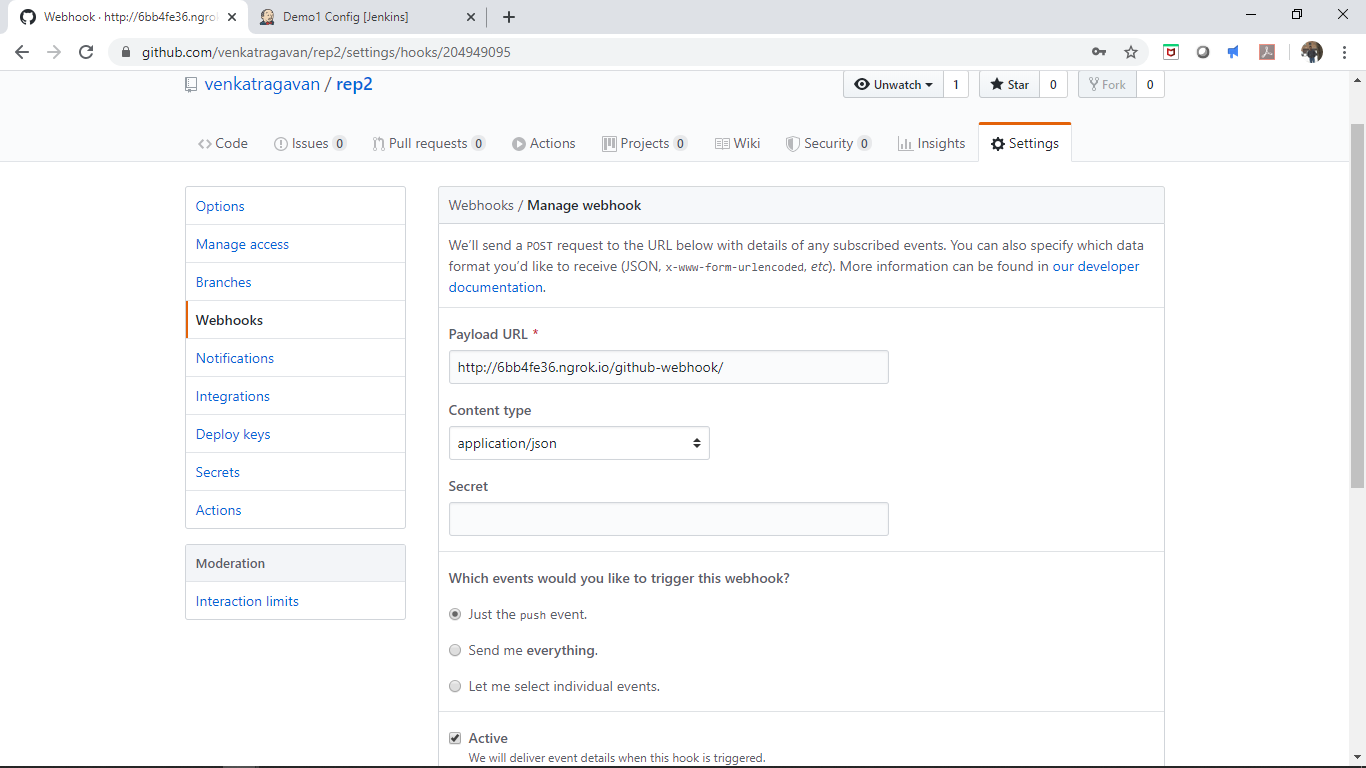


Build step:

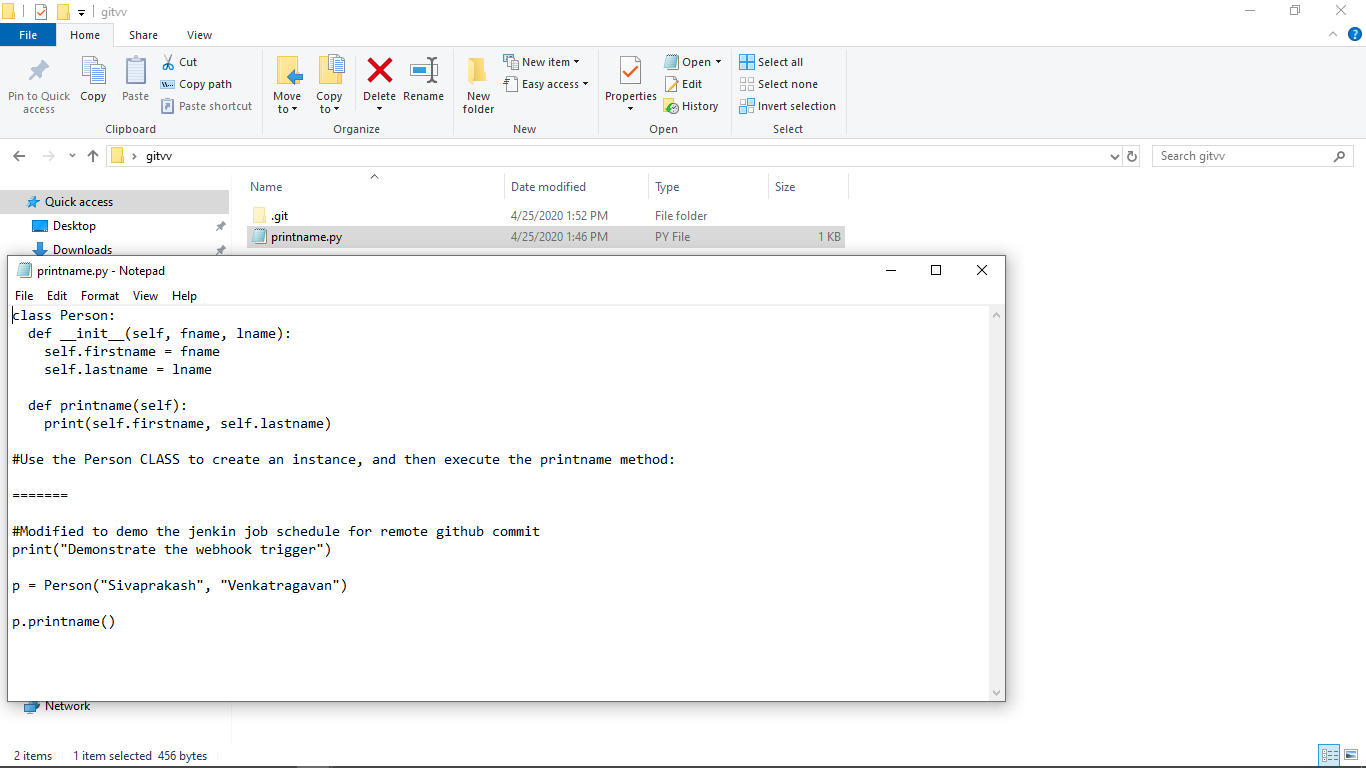
Executed windows batch command to run the python app for current master version and a old commit version

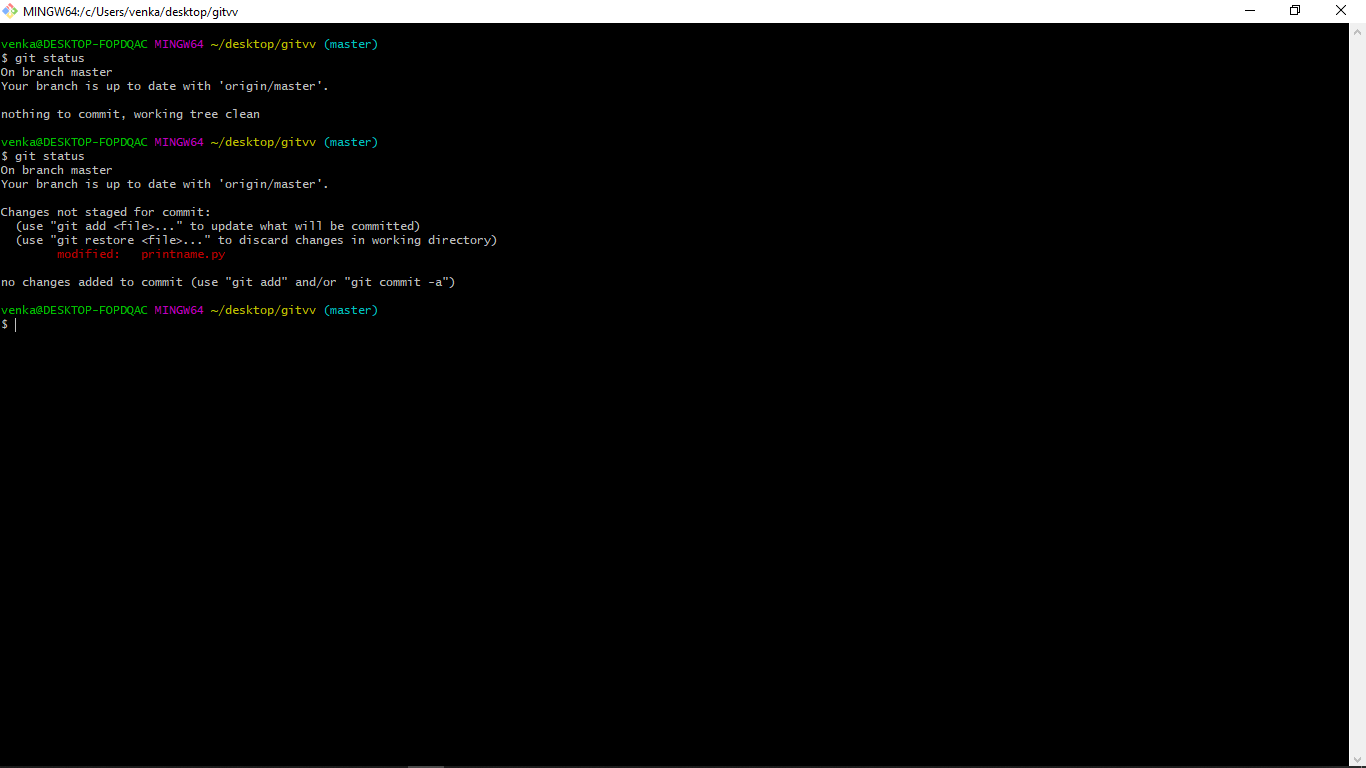


Created a webhook to trigger a push event to jenkins url as when a commit is pushed to the remote github

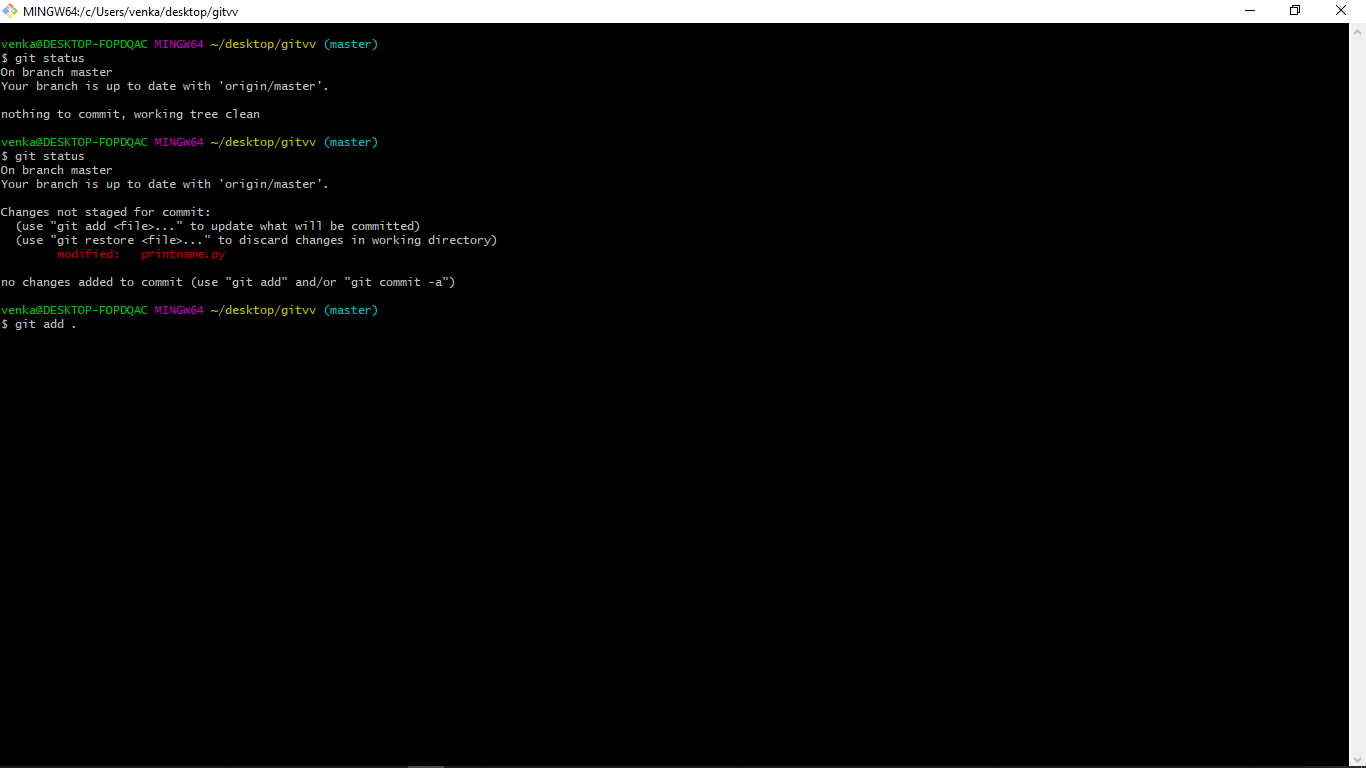


Did a change to the local repo:



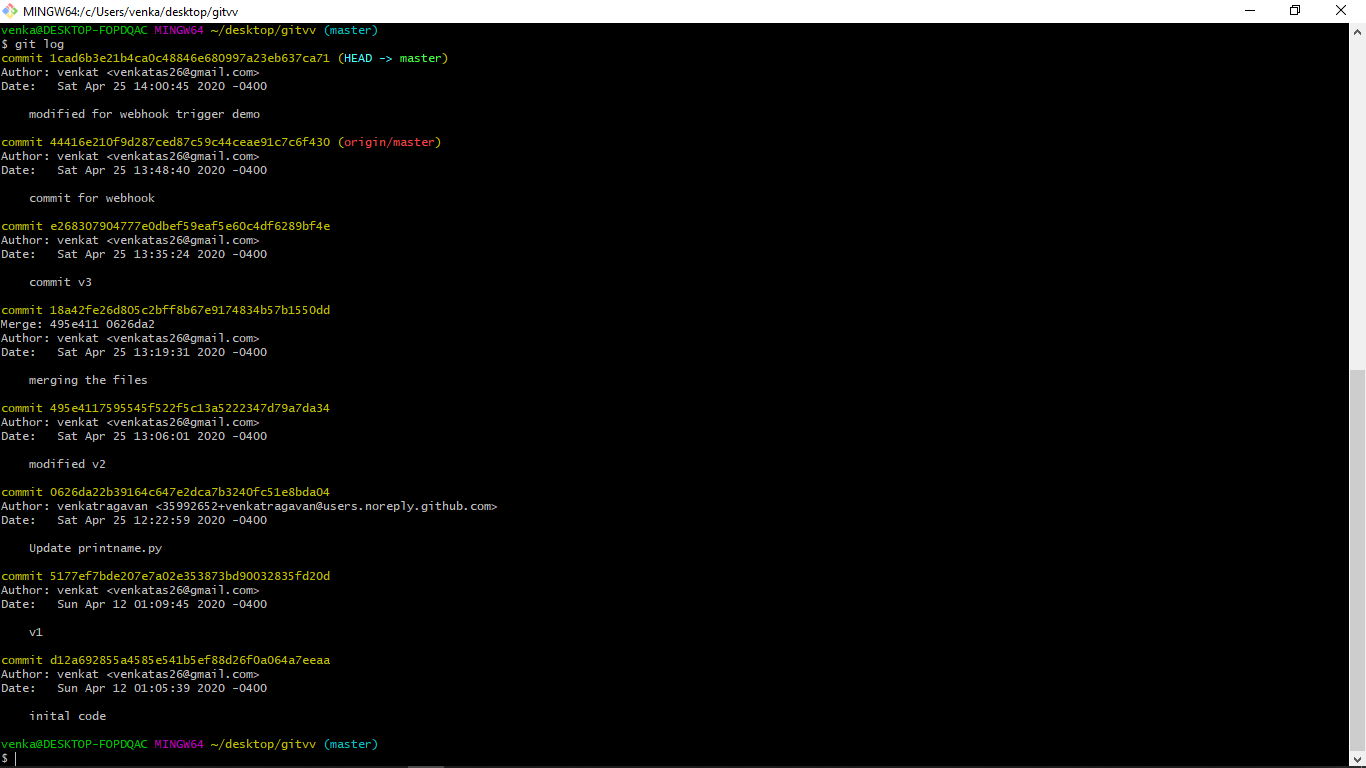


Staged the changes:

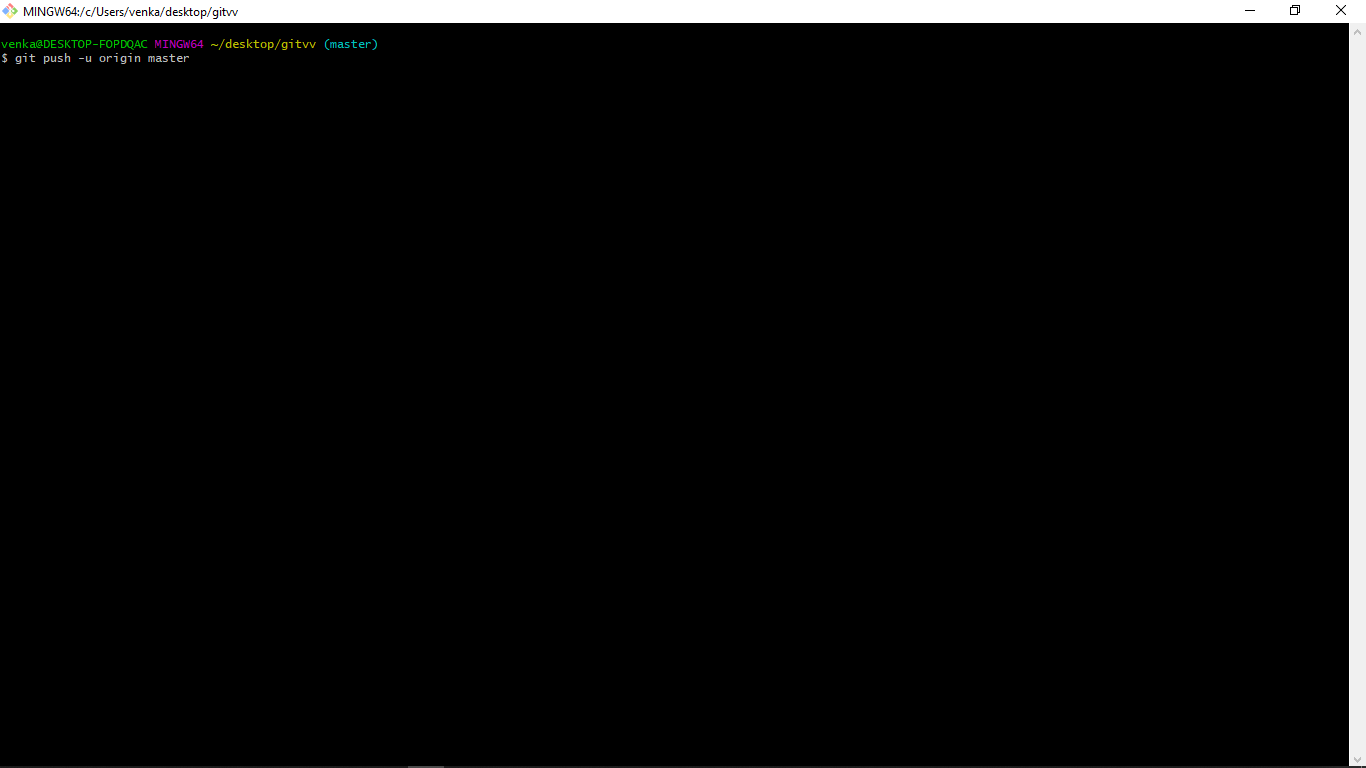


Did a local commit:

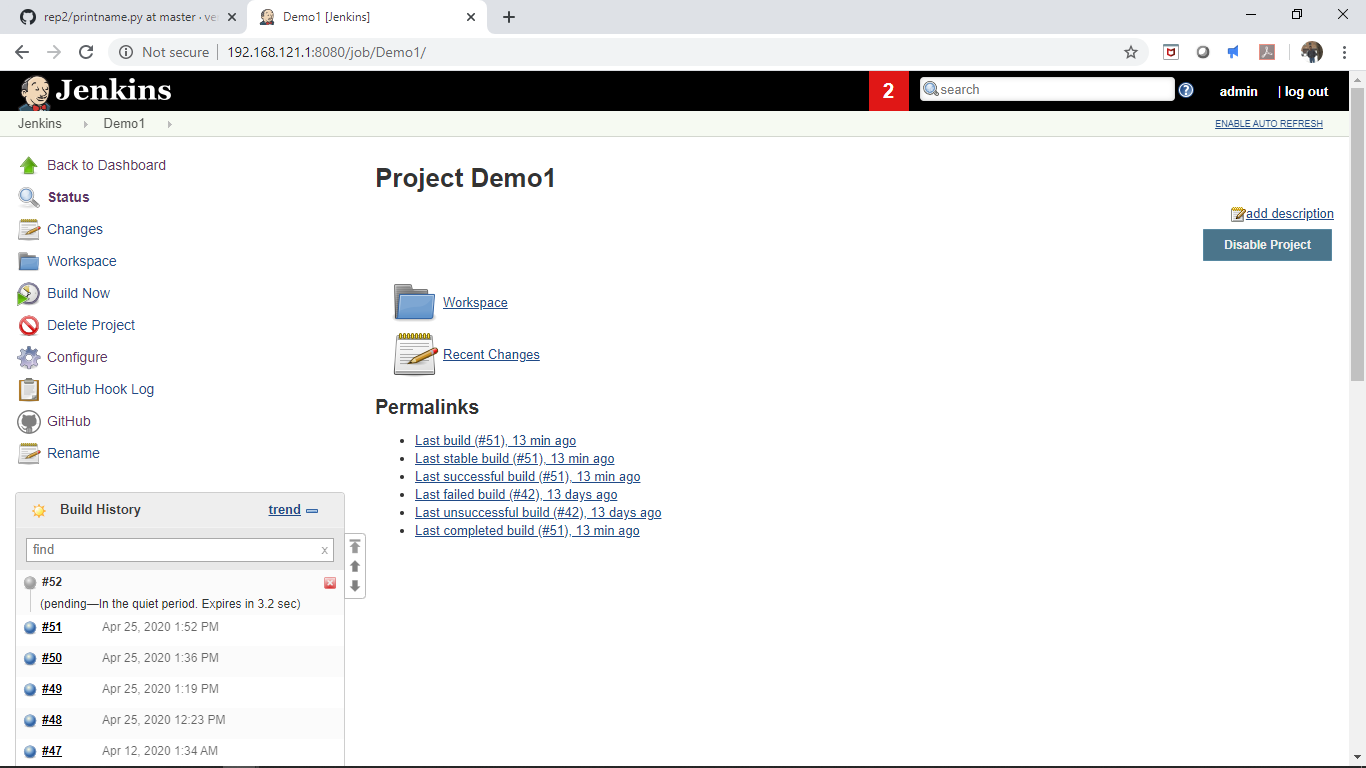




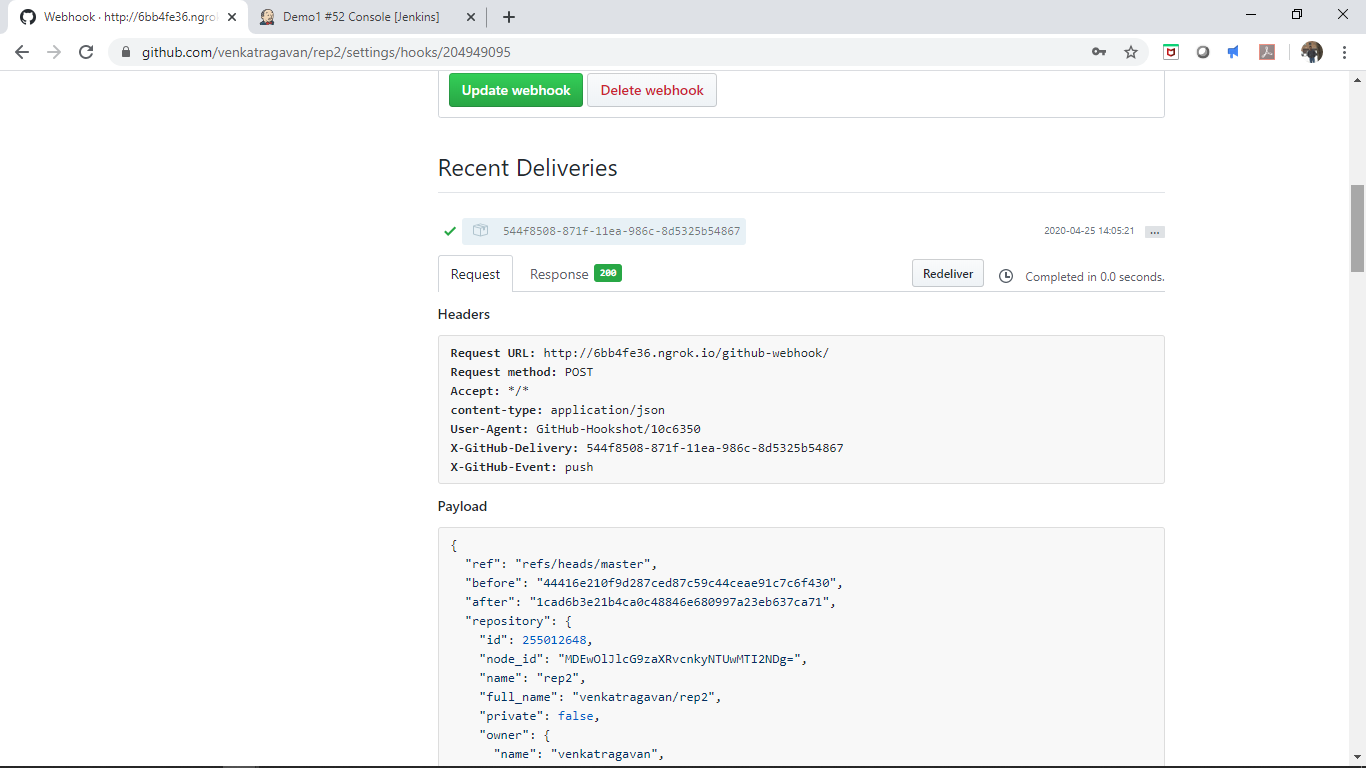
Pushed the local changes to remote repostiory



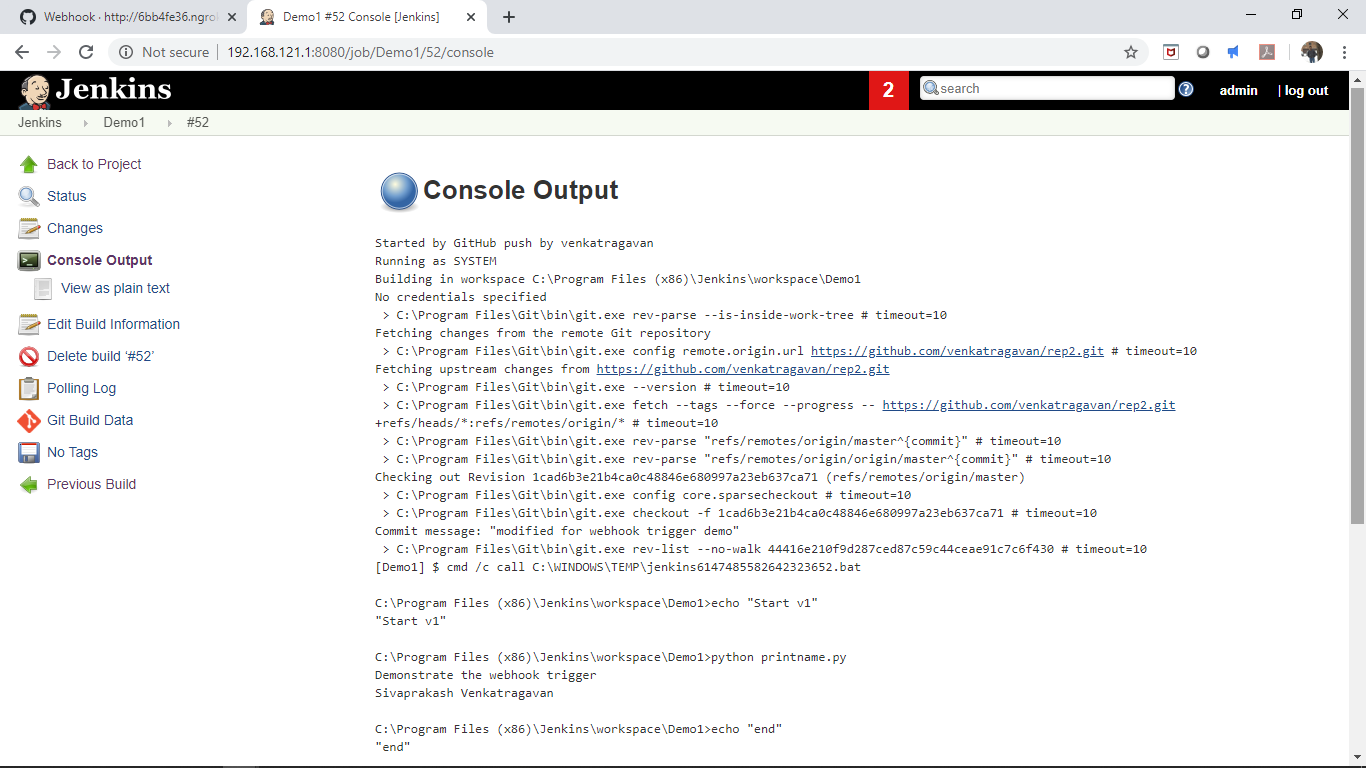
As mentioned below, the webhook triggered the jenkin scheduler:



Below screenshot from github shows the webhook has sent a push event message to jenkins when the remote commit happened via git.

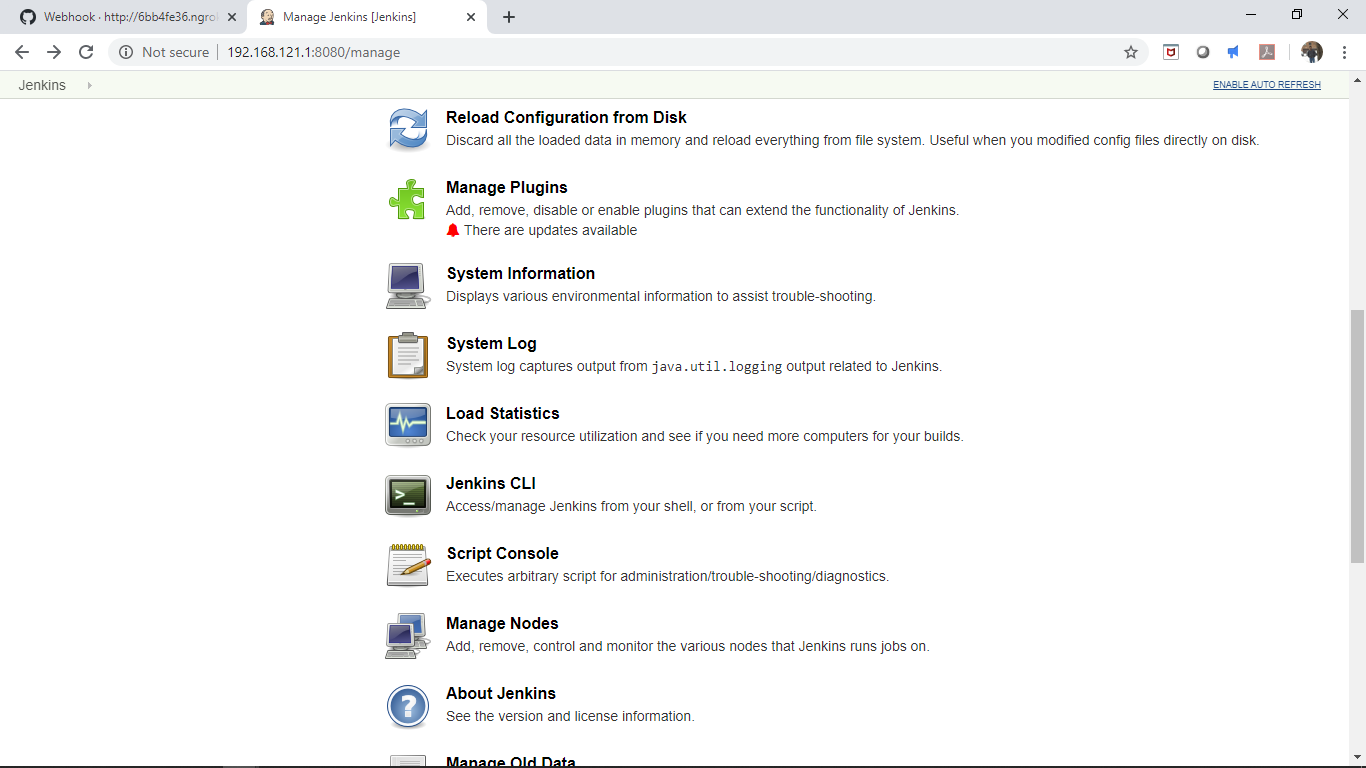


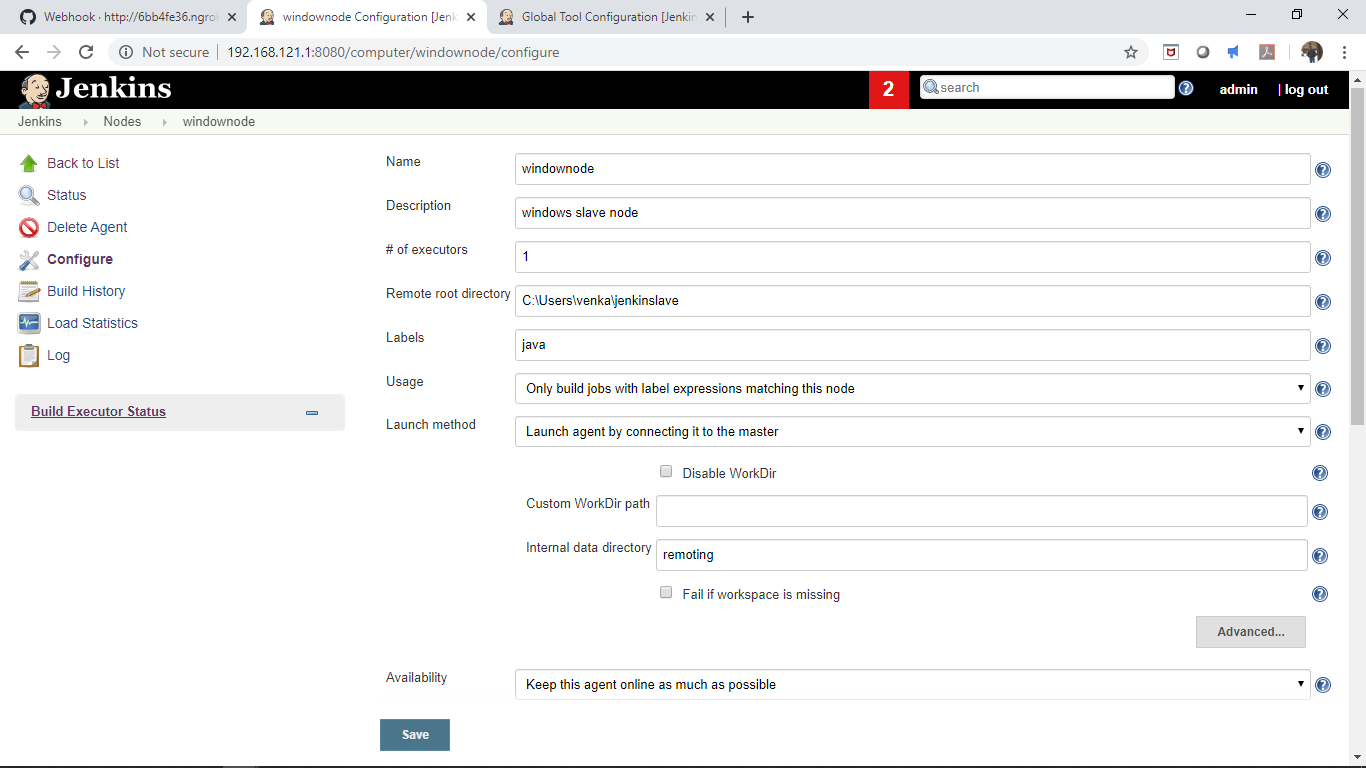
Jenkin job scheduler was successful, below is the console output:

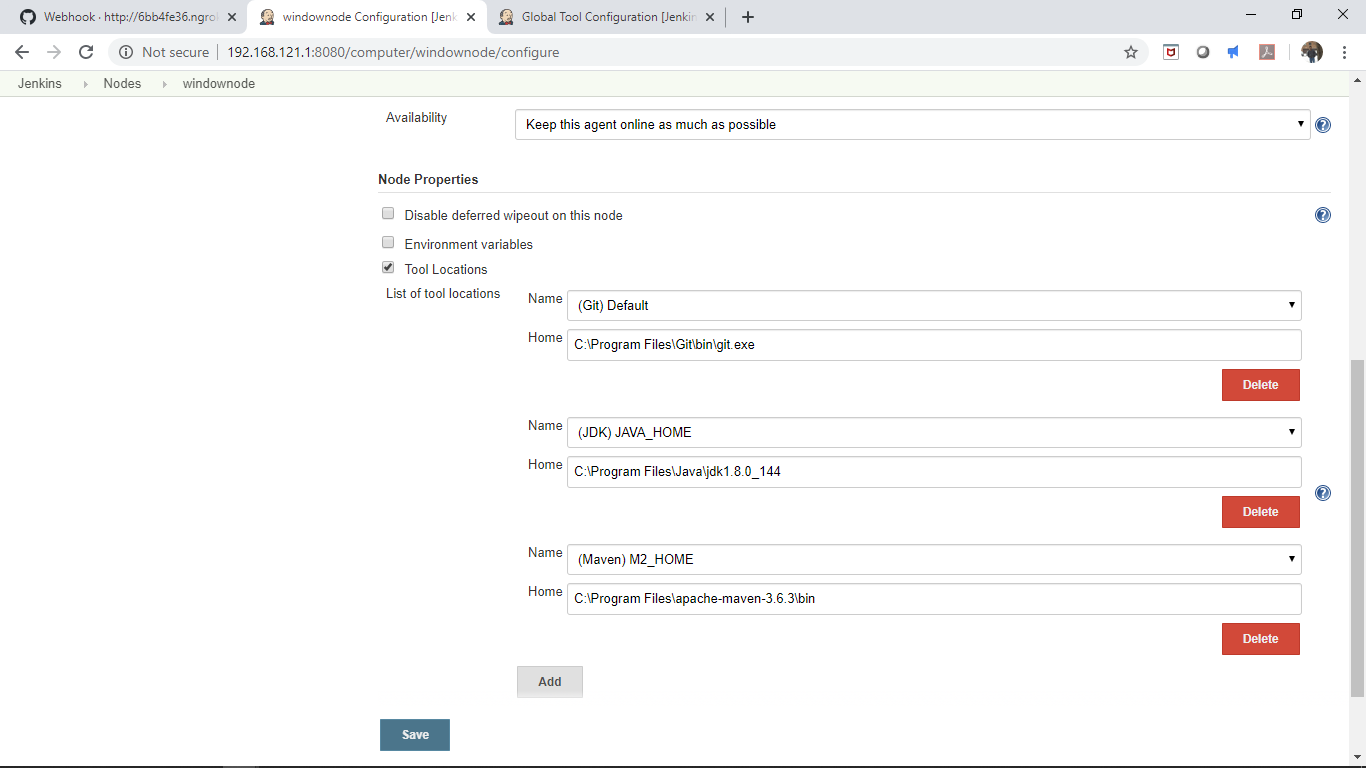


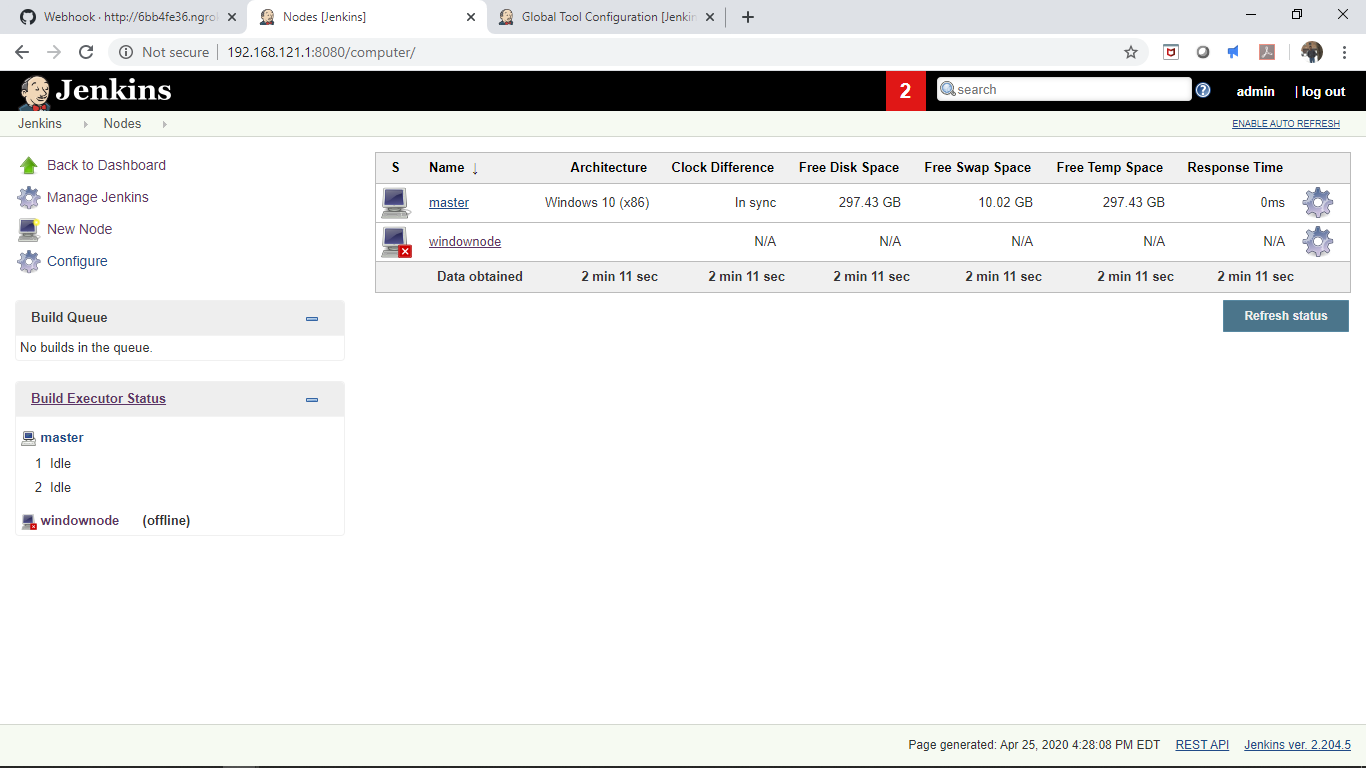
**4. Creating slaves:**

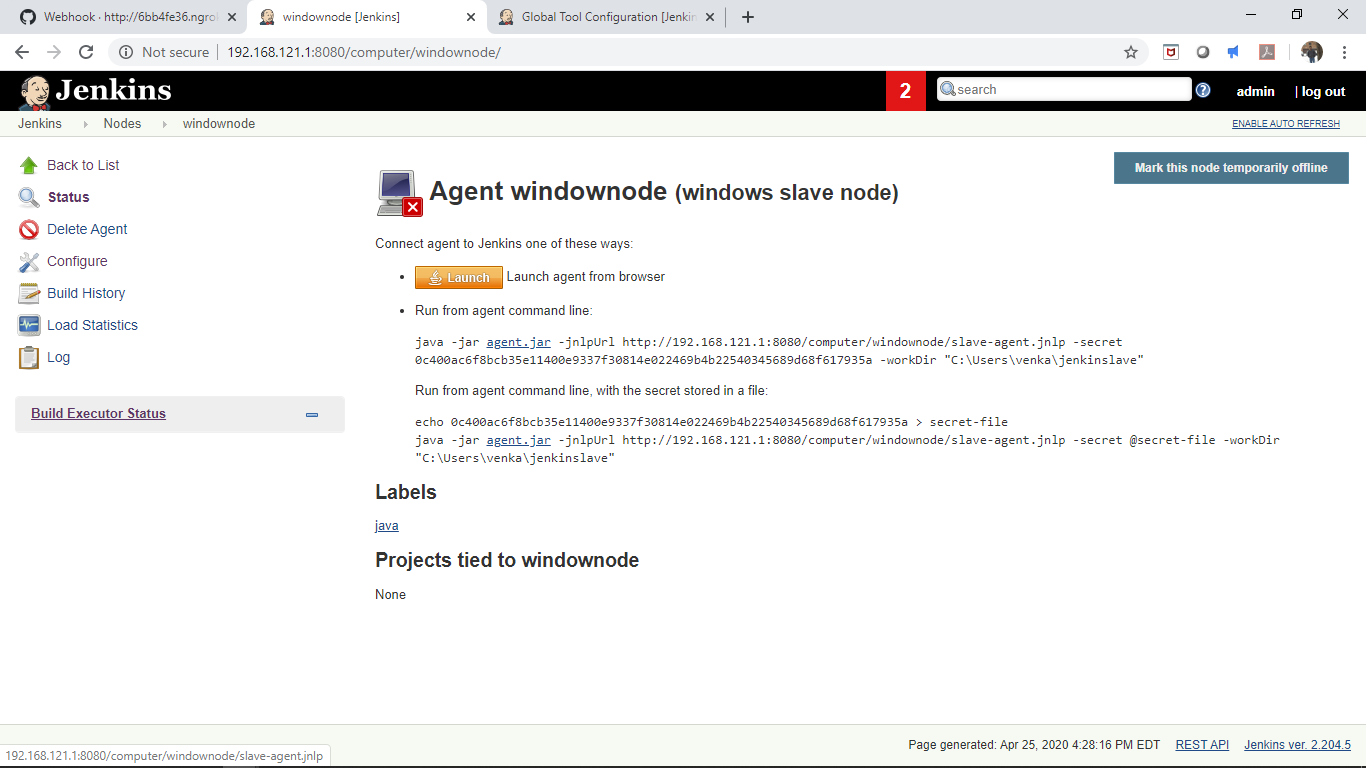
Under manage jenkins, clicked manage nodes:

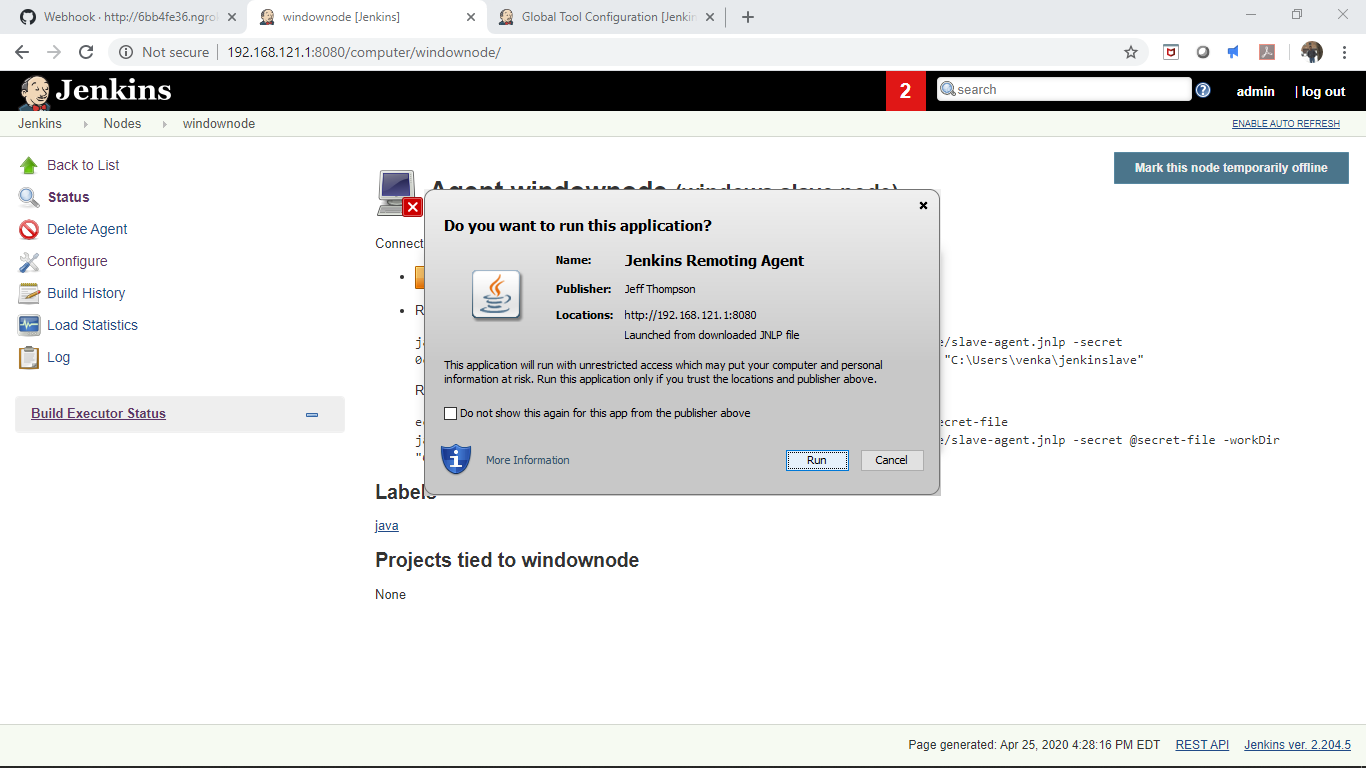


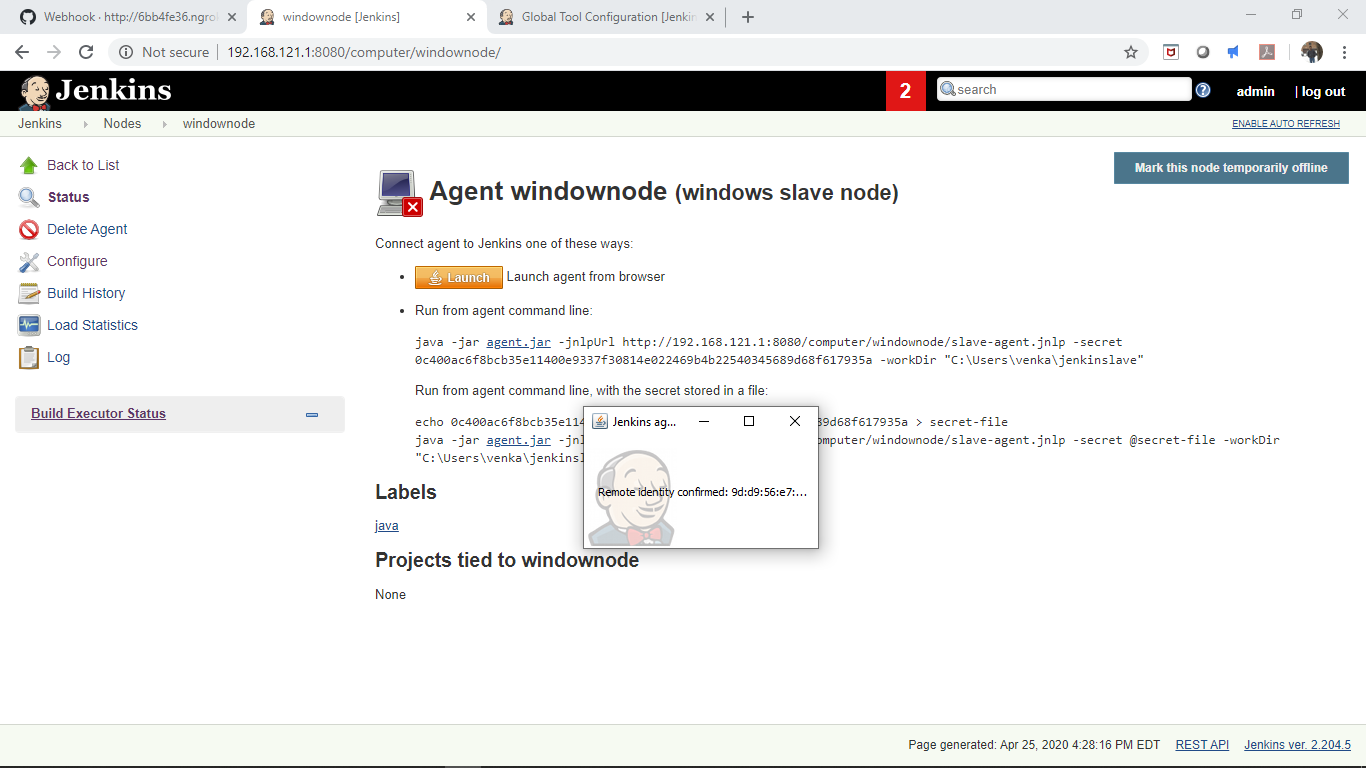


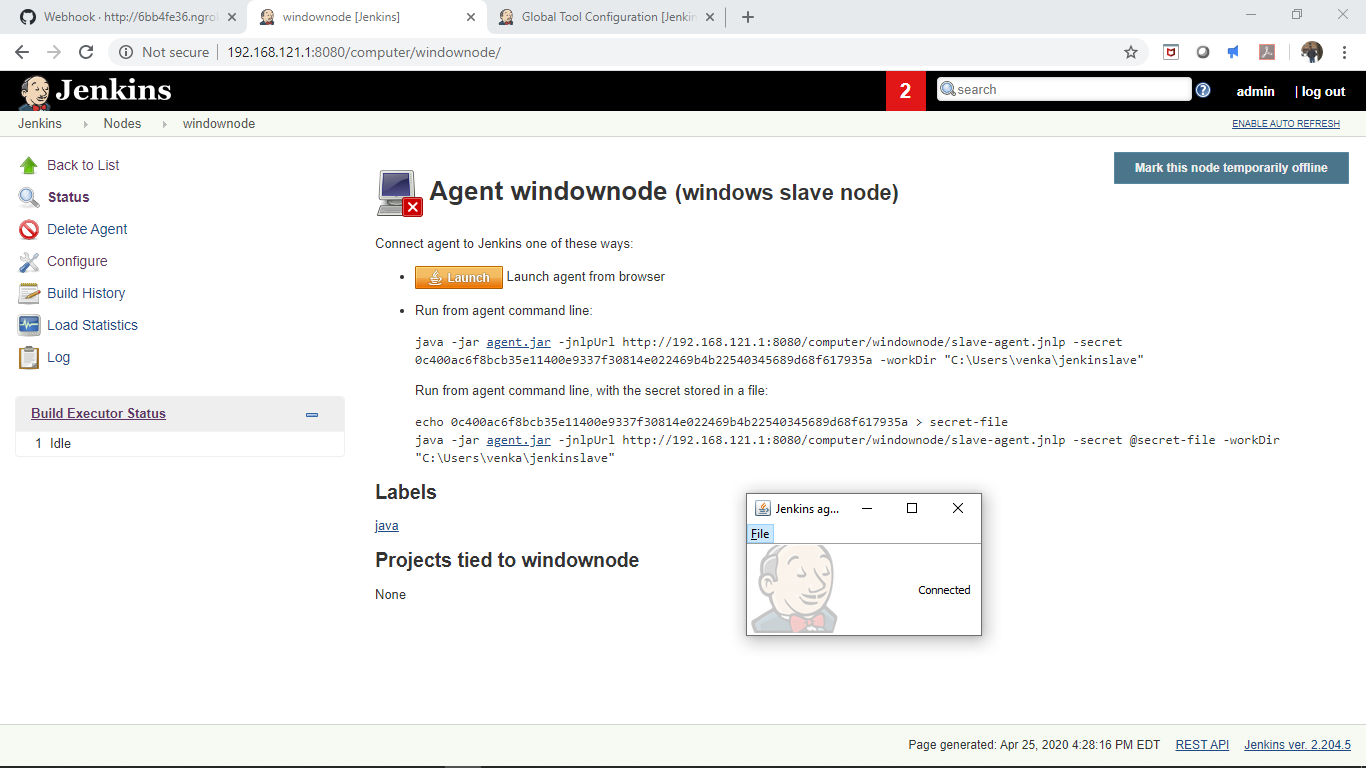




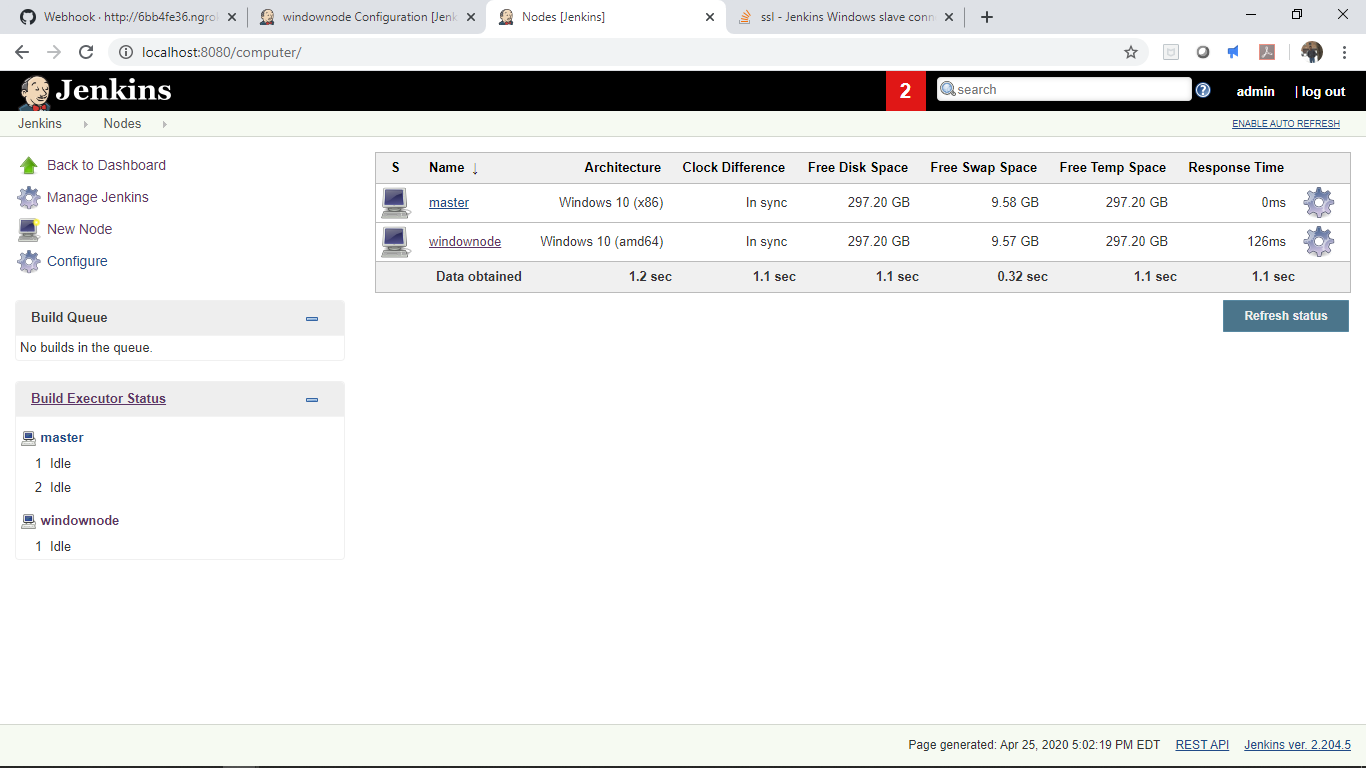






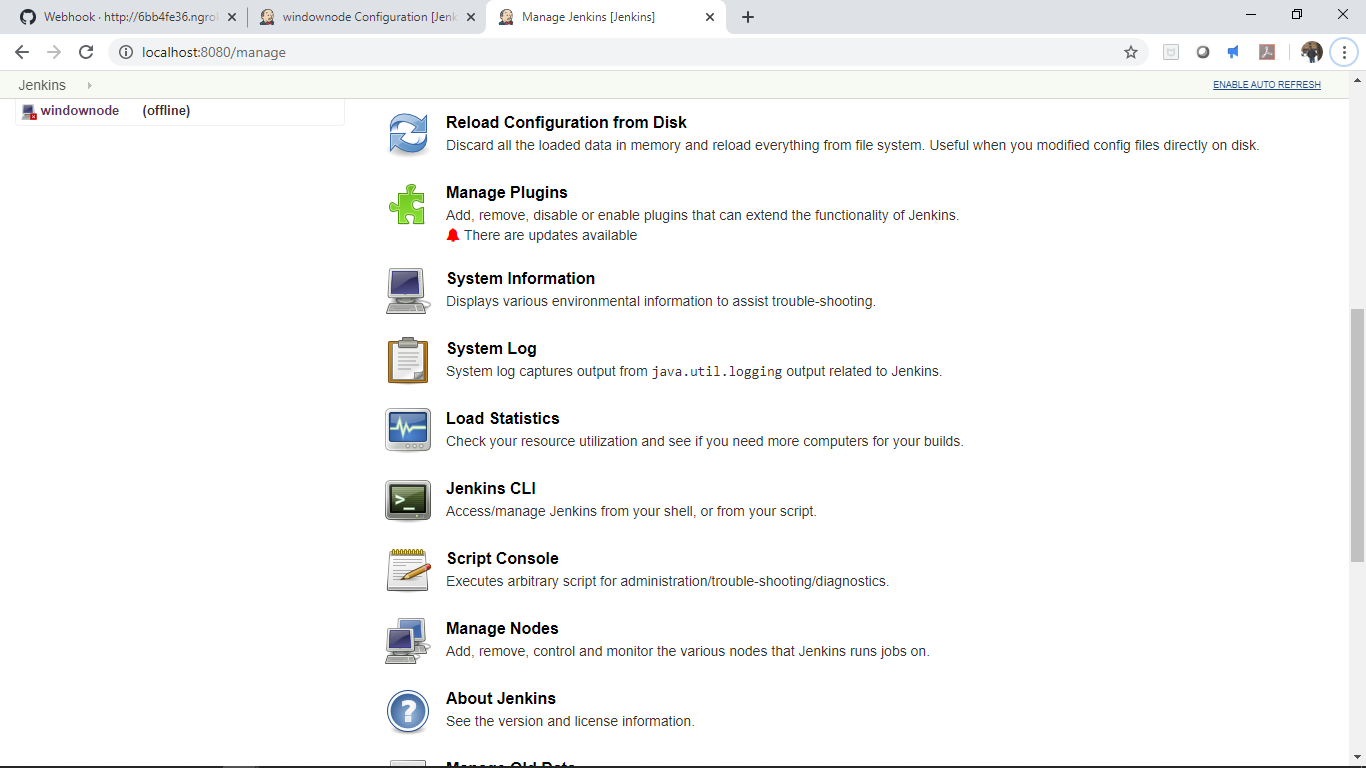




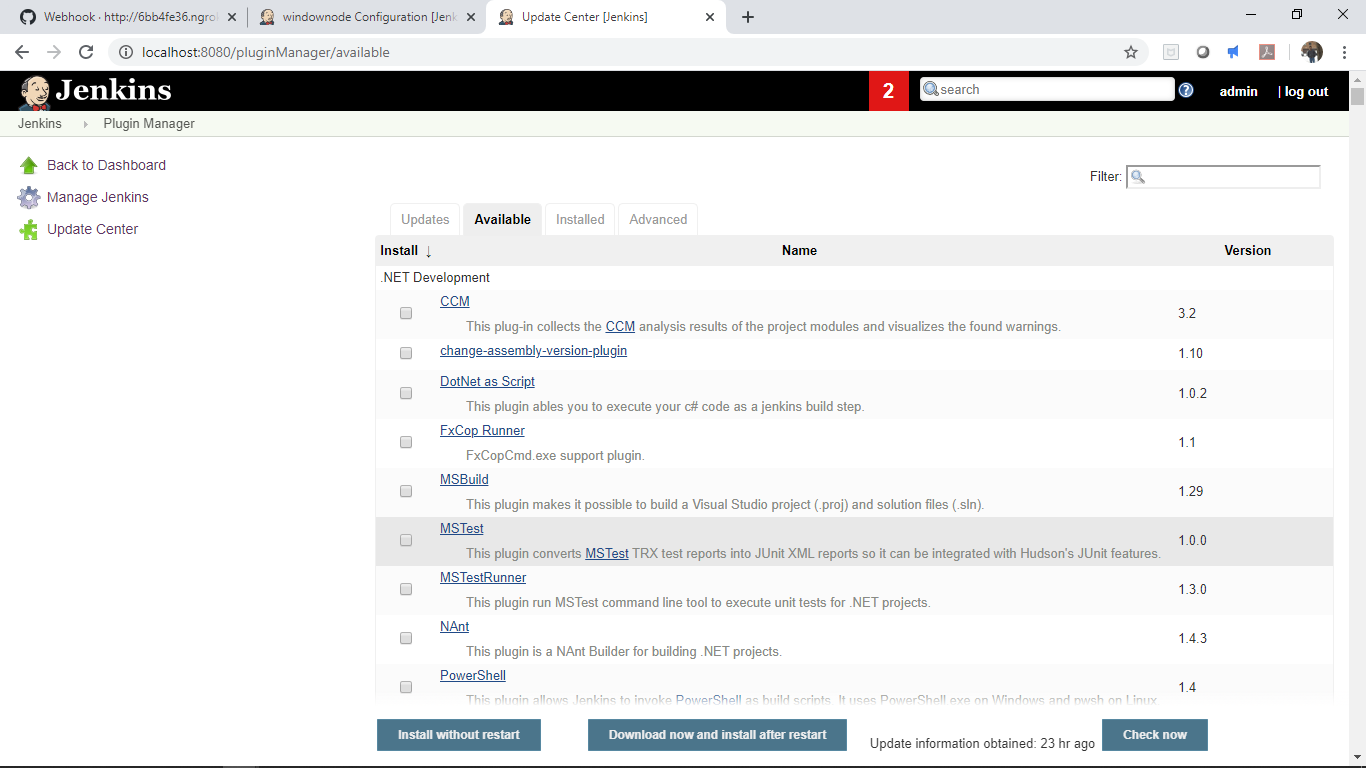


**5. Installing Plugins:**

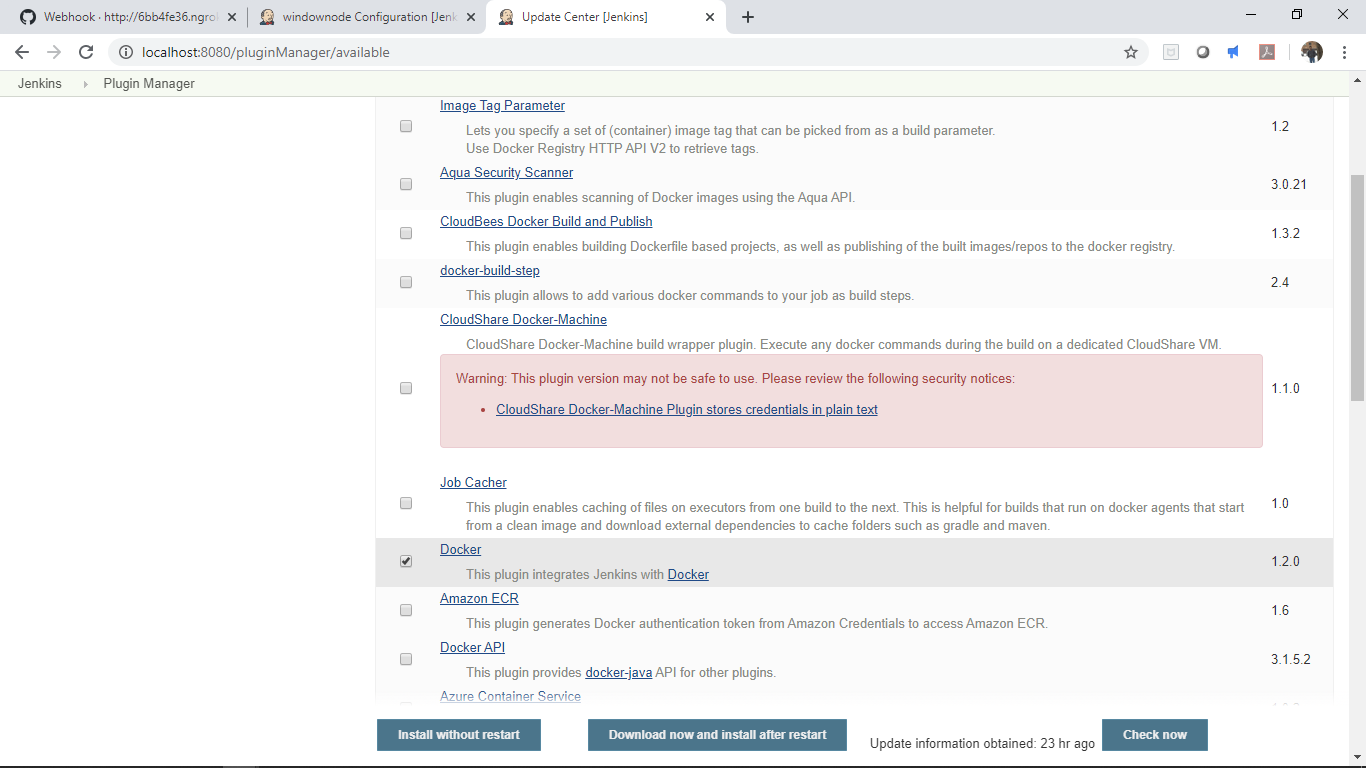
Goto Manage-Jenkins and click Manage plugins:

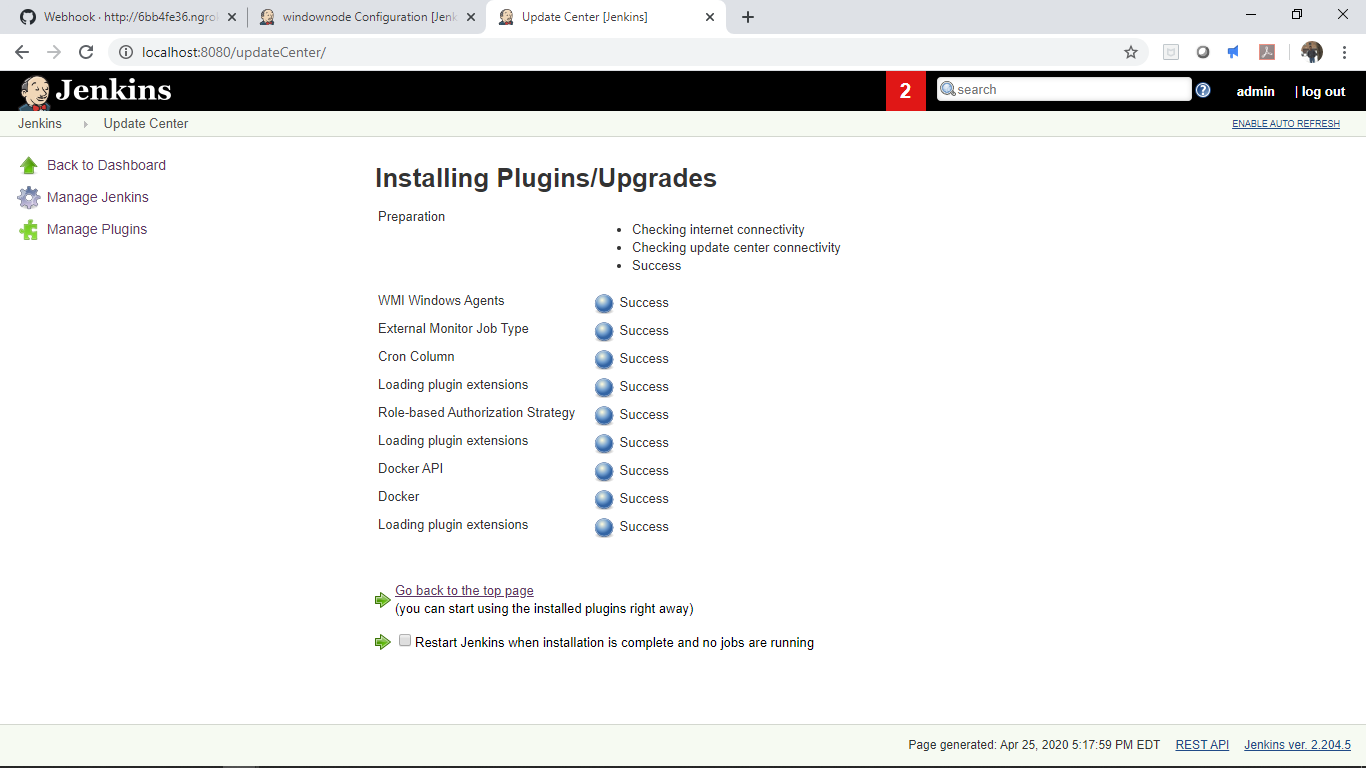


Goto Available section

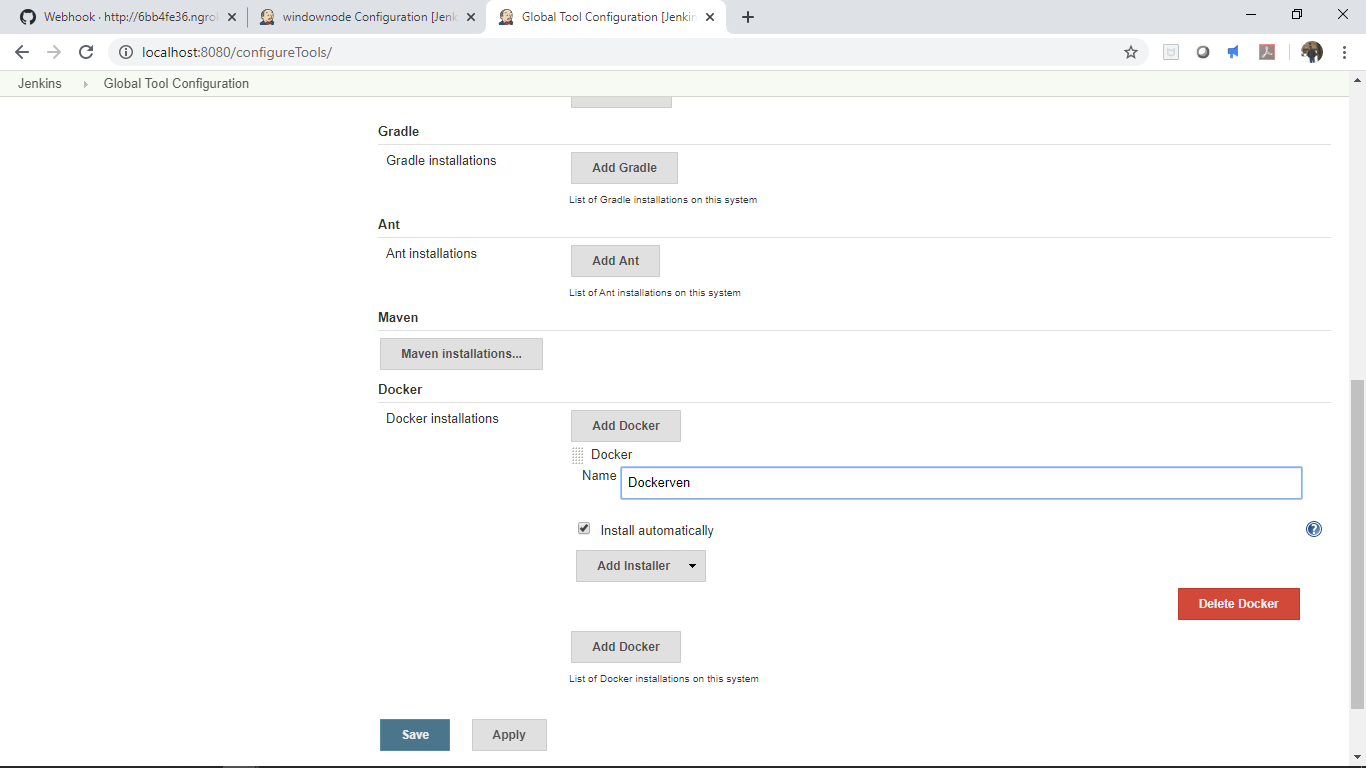


Search for Docker plugin and click install:





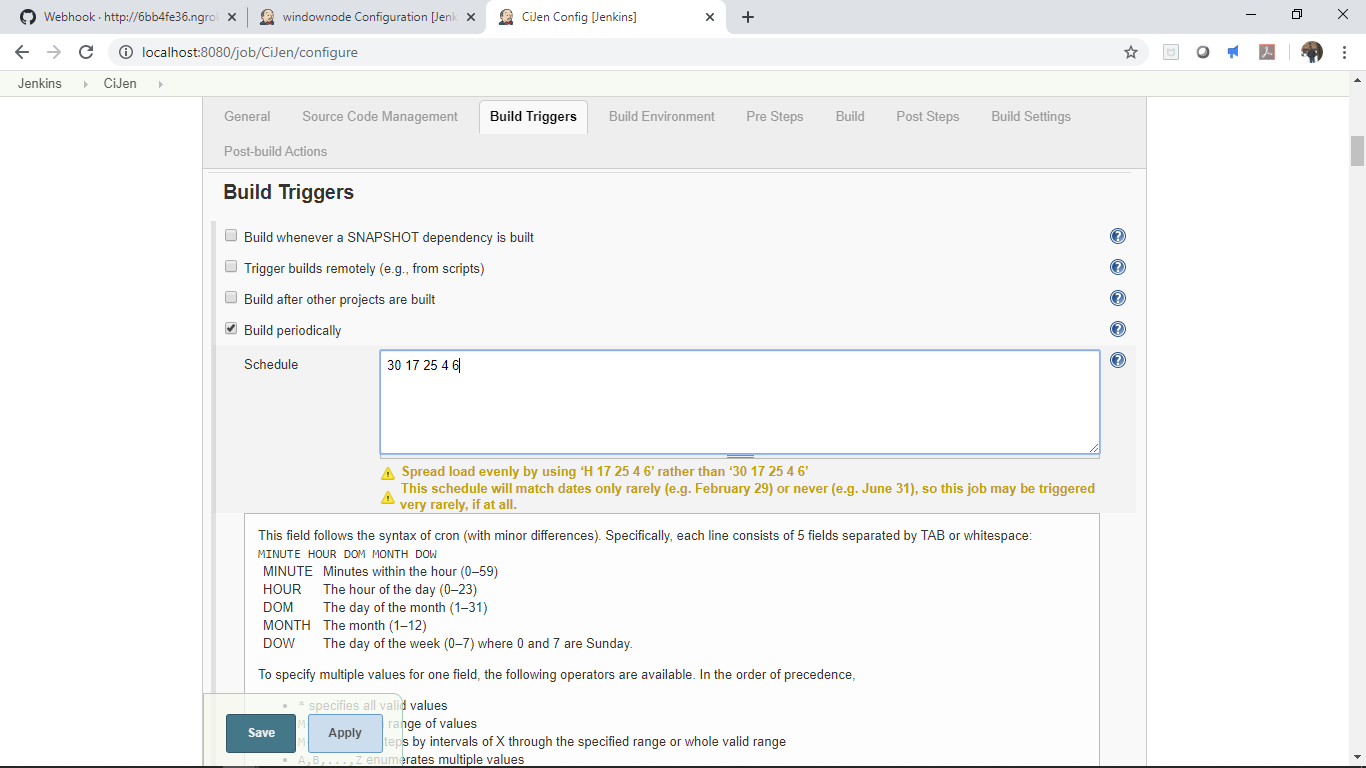
Goto Global tool configuration which has placeholder for Docker configuration as the docker plugin is installed:

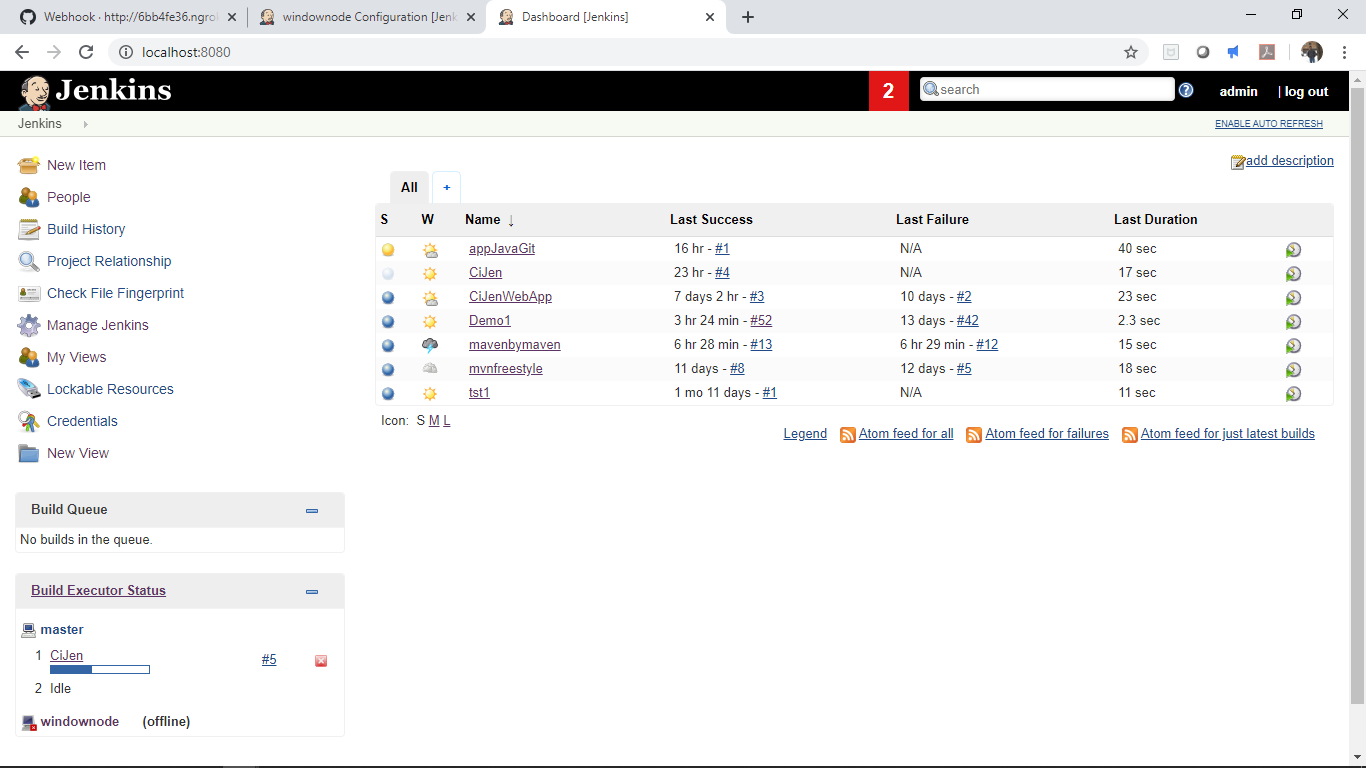


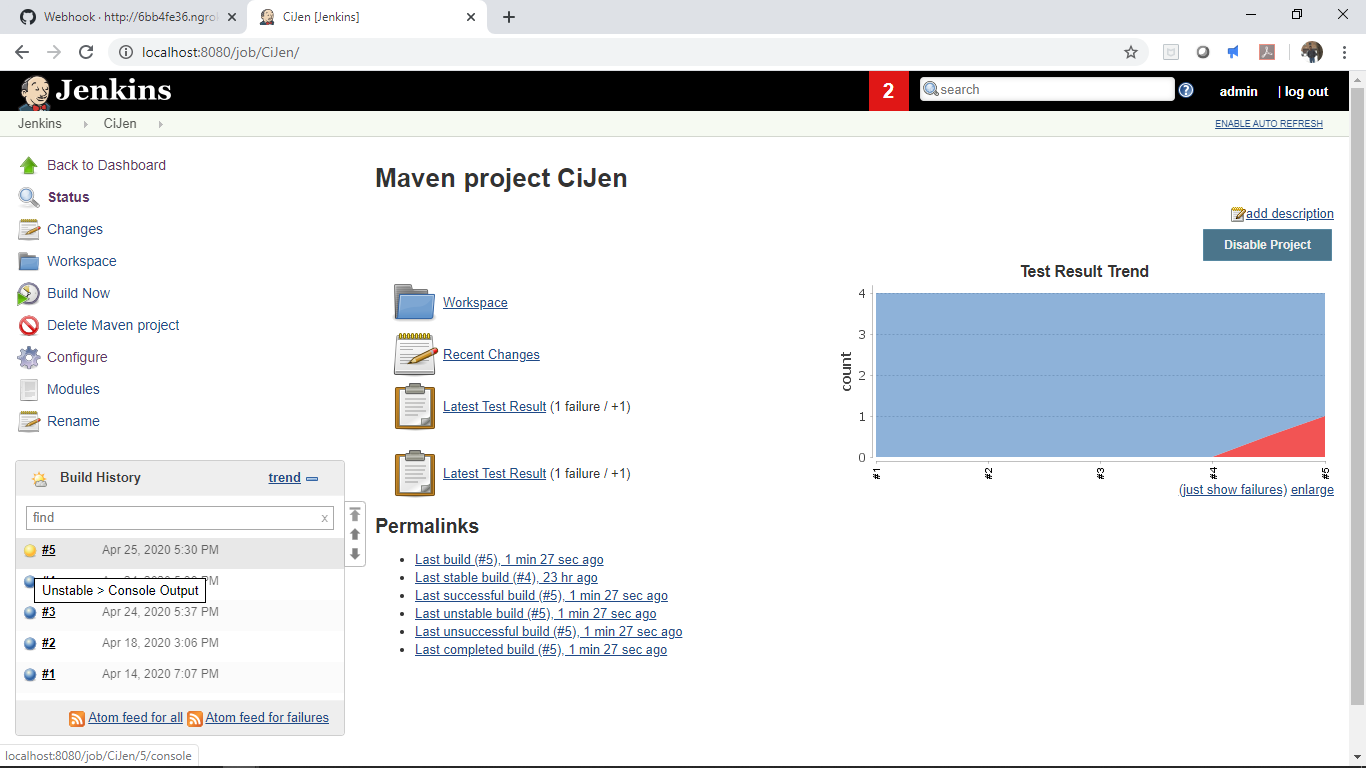
**6. Scheduling Cron jobs in Jenkins**

Used an existing job CiJen, in build trigger chose option “build periodically” as shown below and use the cron format

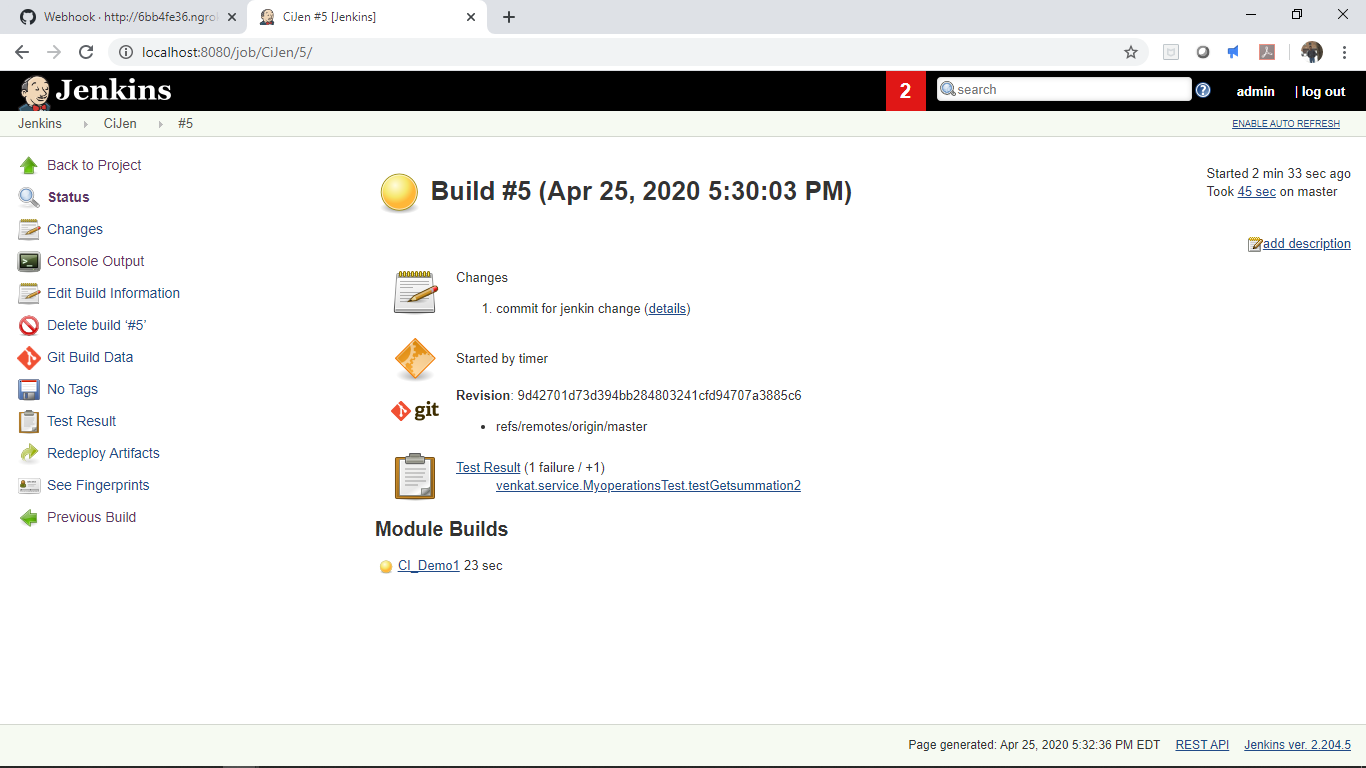
To run the job at 5.30 PM on April 25th.



The job triggered automatically using CRON command at 5.30 PM on April 25th:

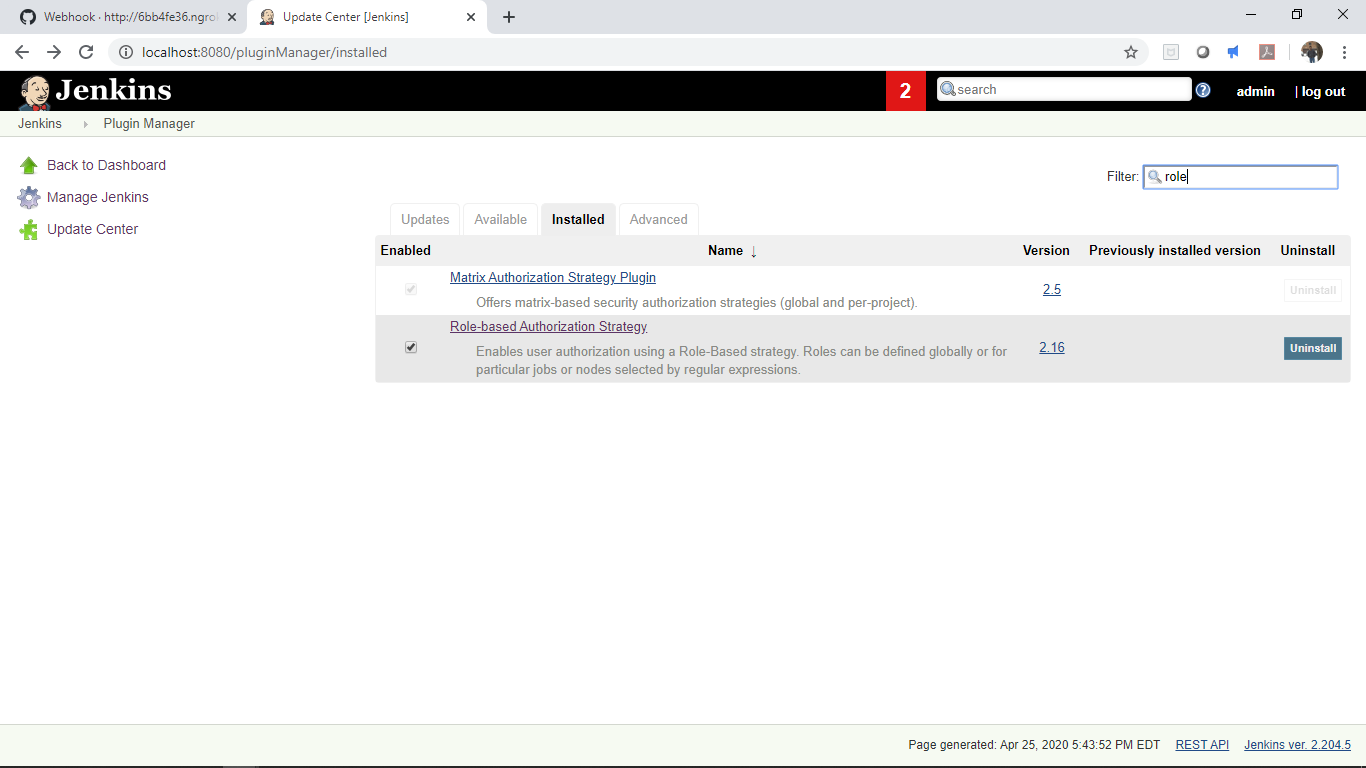


The status below shows the job was scheduled and started by Timer::

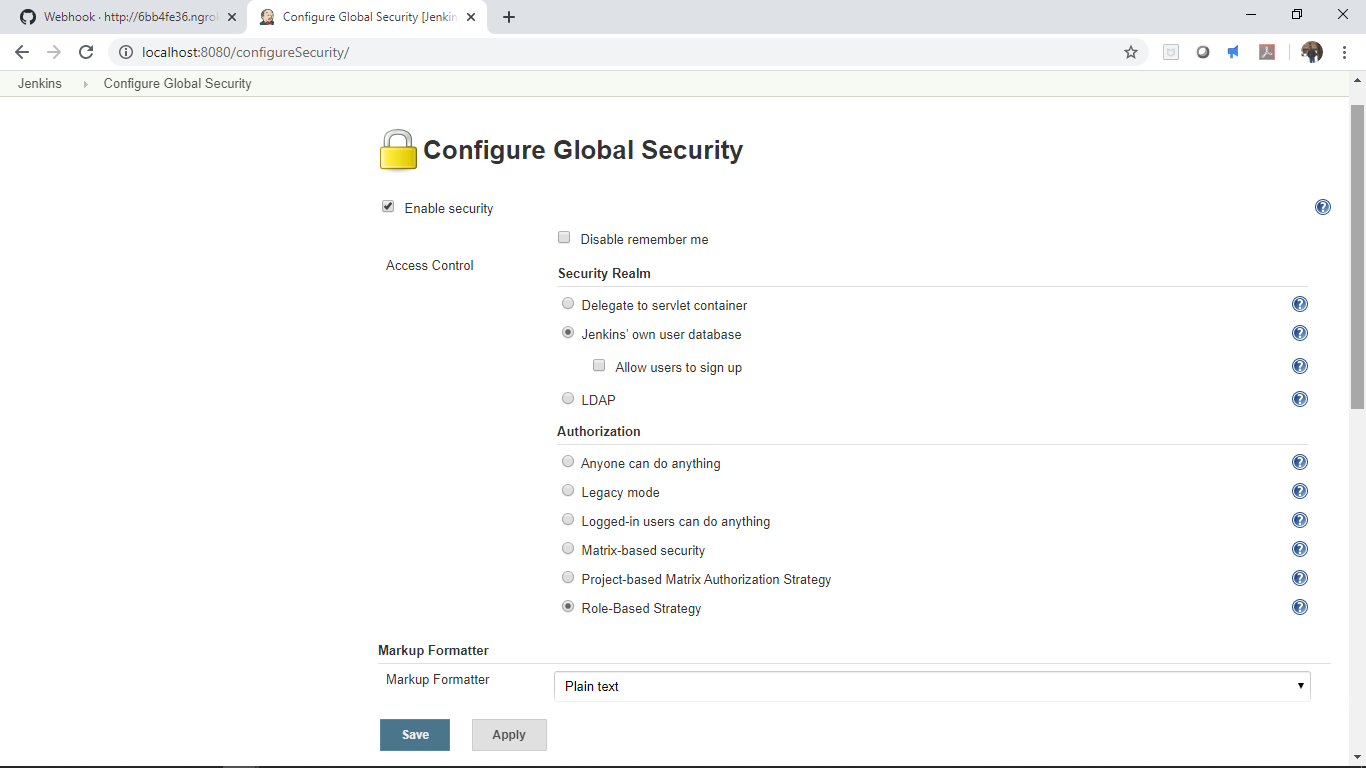


**7. Authorization technique in Jenkins:**

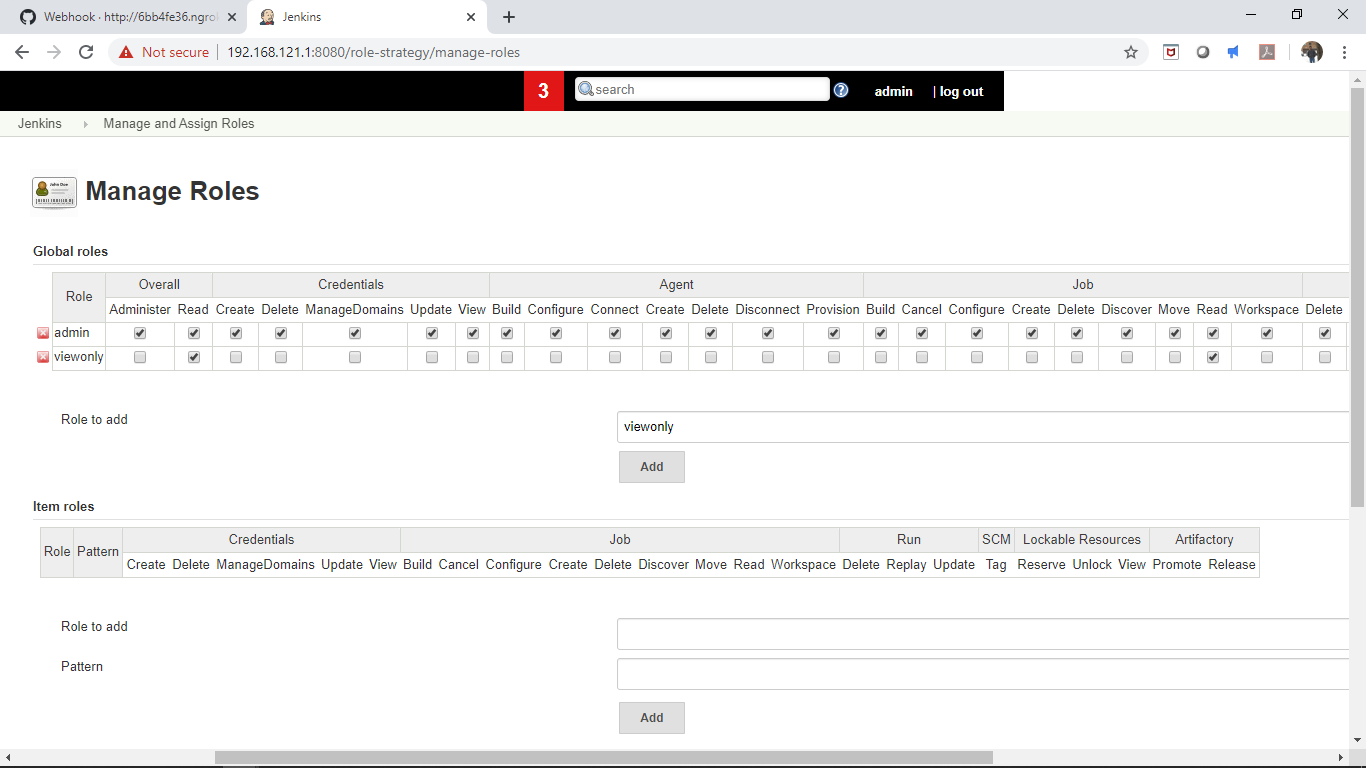
Installed the “Role Based Authorization Strategy” pipeline as shown below:



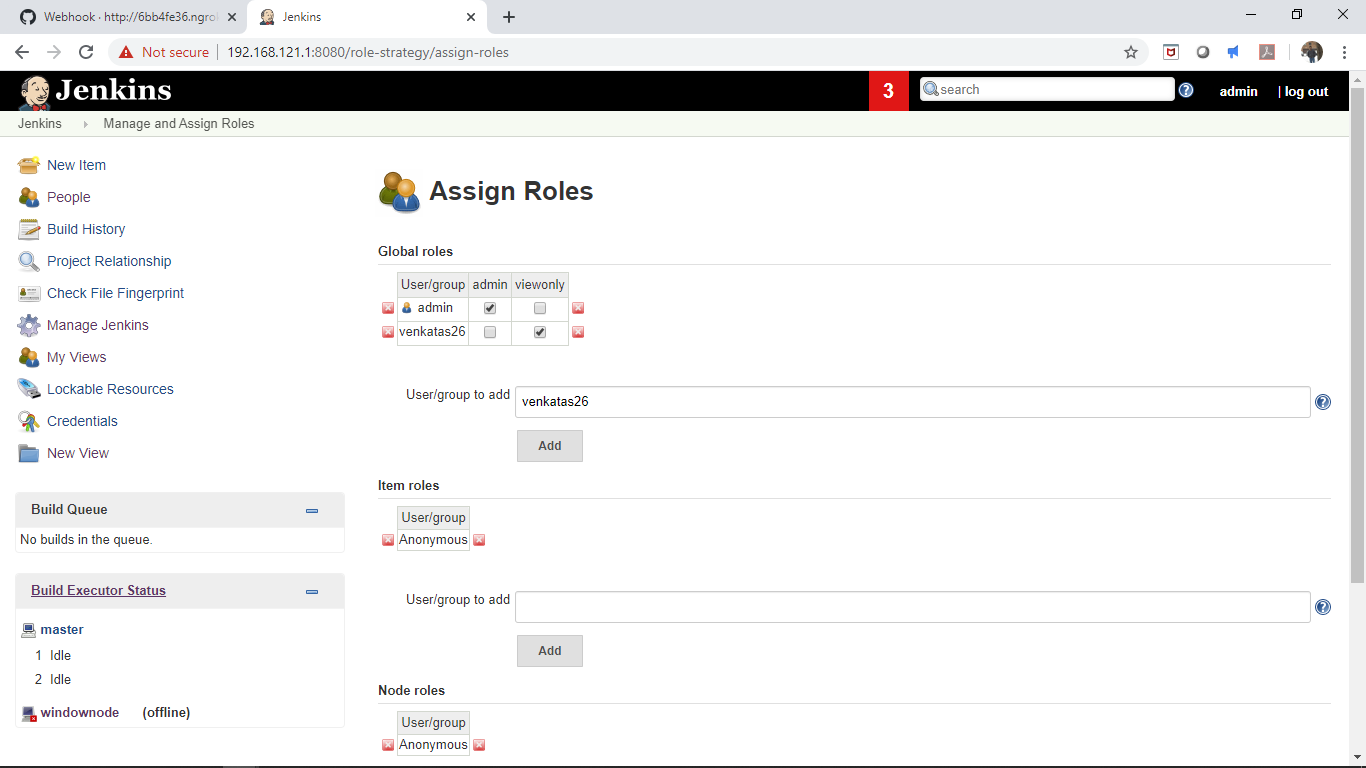
Enabled Role based strategy under “Configure global Security”



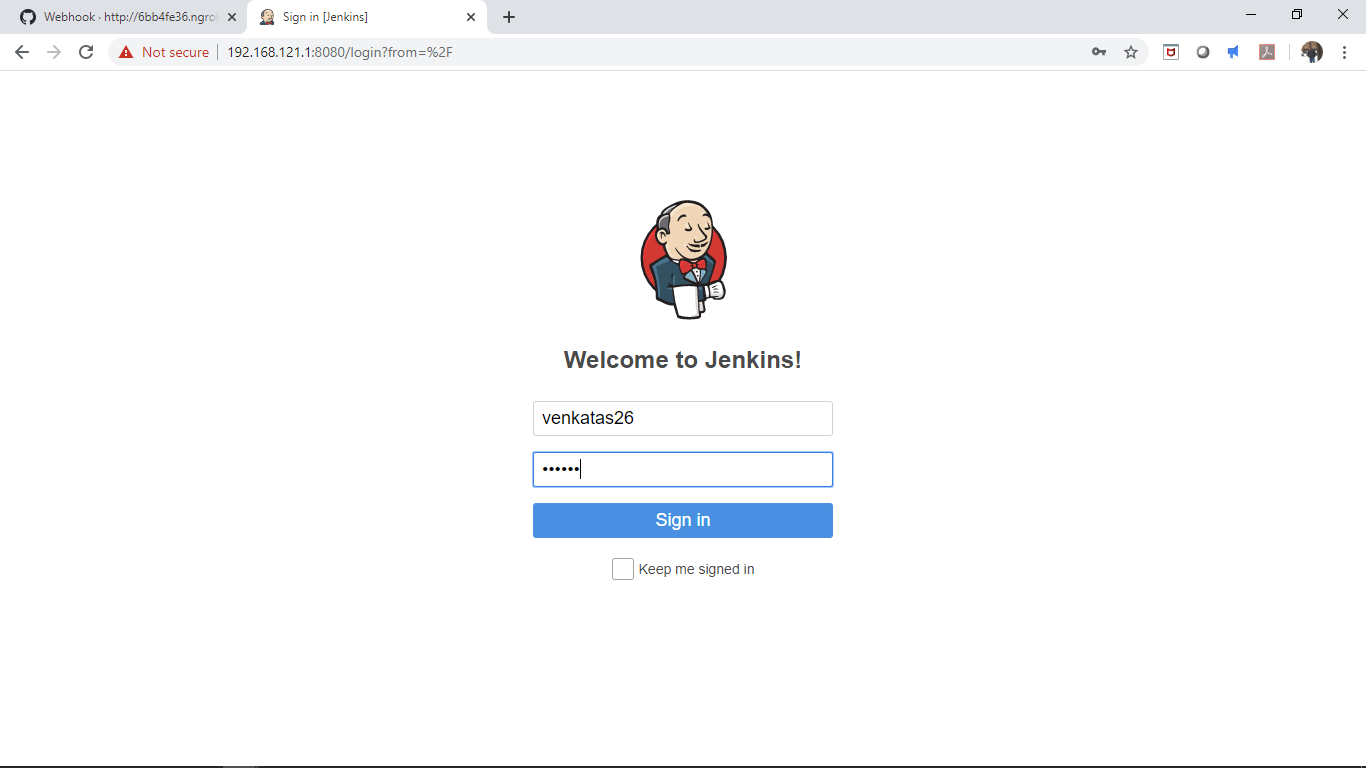
Created a new role “viewonly” with restricted access to only read the Jobs and Views as shown below:



In Assign roles, assigned the userID venkatas26 to “viewonly” role, hence that user can access only to read jobs and view:



Now logged under the user venkatas26



As shown below the user “venkatas26” has only read only access to the views and jobs:

